THE UNIVERSITY
of EDINBURGH

Performance Analysis of Large-Scale Resource-Bound Computer Systems

Alireza Pourranjbar

Doctor of Philosophy

Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh

2015

Abstract

We present an analysis framework for performance evaluation of *large-scale resource-bound* (LSRB) computer systems. LSRB systems are those whose *resources* are continually in demand to serve *resource users*, who appear in large populations and cause high contention. In these systems, the delivery of quality service is crucial, even in the event of resource failure. Therefore, various techniques have been developed for evaluating their performance. In this thesis, we focus on the technique of quantitative modelling, where in order to study a system, first its model is constructed and then the system's behaviour is analysed via the model.

A number of high level formalisms have been developed to aid the task of model construction. We focus on PEPA, a stochastic process algebra that supports compositionality and enables us to easily build complex LSRB models. In spite of this advantage, however, the task of analysing LSRB models still poses unresolved challenges.

LSRB models give rise to very large state spaces. This issue, known as the state space explosion problem, renders the techniques based on discrete state representation, such as numerical Markovian analysis, computationally expensive. Moreover, simulation techniques, such as Gillespie's stochastic simulation algorithm, are also computationally demanding, as numerous trajectories need to be collected.

Furthermore, as we show in our first contribution, the techniques based on the mean-field theory or fluid flow approximation are not readily applicable to this case. In LSRB models, resources are not assumed to be present in large populations and models exhibit highly noisy and stochastic behaviour. Thus, the mean-field deterministic behaviour might not be *faithful* in capturing the system's randomness and is potentially too crude to show important aspects of their behaviours. In this case, the modeller is unable to obtain important performance indicators, such as the reliability measures of the system. Considering these limitations, we contribute the following analytical methods particularly tailored to LSRB models.

First, we present an aggregation method. The aggregated model captures the evolution of only the system's resources and allows us to efficiently derive a probability distribution over the configurations they experience. This distribution provides full faithfulness for studying the stochastic behaviour of resources. The aggregation can be applied to all LSRB models that satisfy a syntactic aggregation condition, which can be quickly checked syntactically. We present an algorithm to generate the aggregated

model from the original model when this condition is satisfied.

Second, we present a procedure to efficiently detect time-scale near-complete decomposability (TSND). The method of TSND allows us to analyse LSRB models at a reduced cost, by dividing their state spaces into loosely coupled blocks. However, one important input is a partition of the transitions defined in the model, categorising them into *slow* or *fast*. Forming the necessary partition by the analysis of the model's complete state space is costly. Our process derives this partition efficiently, by relying on a theorem stating that our aggregation preserves the original model's partition and therefore, it can be derived by an efficient reachability analysis on the aggregated state space. We also propose a clustering algorithm to implement this reachability analysis.

Third, we present *the method of conditional moments* (MCM) to be used on LSRB models. Using our aggregation, a probability distribution is formed over the configurations of a model's resources. The MCM outputs the time evolution of the conditional moments of the marginal distribution over resource users given the configurations of resources. Essentially, for each such configuration, we derive measures such as conditional expectation, conditional variance, etc. related to the dynamics of users. This method has a high degree of faithfulness and allows us to capture the impact of the randomness of the behaviour of resources on the users.

Finally, we present the advantage of the methods we proposed in the context of a case study, which concerns the performance evaluation of a two-tier wireless network constructed based on the femto-cell macro-cell architecture.

Lay Summary

These days most of the applications that we run on our computers or smart phones rely heavily on communication with external services running on clouds or server farms, and many service providers serve millions of users. Users demand responsiveness and low latency unconditionally and at all times. Thus, it is important for service providers to continually assess the performance of their infrastructure, allocate resources if needed and adjust the system parameters to meet users' demands.

Various mechanisms exist that claim to offer intelligent system tuning and accurate performance evaluation. A key requirement however, is that a valid mechanism *needs to be reliable and robust*. The operation of a computer system is usually affected by many factors such as outsourced services, user load, resource failure. When tuning a system or predicting its behaviour, the performance evaluation mechanism must take all such factors into account and give us the system's true behaviour. It is naive to assume that a system under study always follows an *average* or *most commonly observed* path. In contrast, real systems usually experience large variations in their operation (arrival of peak demand, failure of critical and bottleneck resources), giving rise to optimal and potentially many sub-optimal or critical situations. A reliable performance evaluation detects all such situations and investigates user satisfaction in each of them separately.

In this thesis, we propose a performance evaluation framework that promises a high degree of reliability and robustness. Our approach is based on mathematical and quantitative modelling. A model is an abstract, formal representation of a real system and its mathematical analysis provides us with valuable insight on how the system and its users interact. Our models are specified in a high-level modelling language called PEPA that can be easily used to build models of complex computer systems. Given a model, we analyse it to extract the system's optimal and sub-optimal modes of operation using ideas from Markov chain theory. Since the model is formal, this step can be done automatically, and thus we can deal with very complex systems which feature complicated dependencies between parameters. In the second step, the quality of the service provided to the users is measured for each of the operational modes experienced by the system using the theory of differntial equations. Our framework allows us to understand how the system's optimal and sub-optimal configurations affect users' experience. Our framework helps system designers to efficiently tune their systems such that they present satisfactory service in both the ideal and critical circumstances.

Acknowledgements

First, and foremost, I thank my supervisor, Jane Hillston, for her support, guidance and unlimited patience. I started the PhD program in Edinburgh without a strong background in formal methods. She allowed me to learn at my own pace, tolerated all my weaknesses in terms of theory, writing and other skill sets, and continually gave me hope that I can produce this thesis.

I have much gratitude for Vashti Galpin, my second supervisor, for her valuable feedback.

I would like to pay my respect to Ian Stark and Kousha Etessami for their guidance and feedback in my annual review meetings.

I thank the School of Informatics for supporting me financially throughout my PhD program.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

This thesis includes content from the following published papers:

1. A. Pourranjbar, J. Hillston, and L. Bortolussi. Don't just go with the flow: cautionary tales of fluid flow approximation. In Computer Performance Engineering, LNCS 7587, pages 156 – 171. 2013.

2. A. Pourranjbar and J. Hillston. An aggregation technique for large-scale PEPA models with non-niform populations. In Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools, ValueTools '13, pages 20 – 29, Brussels, Belgium, 2013.

Furthermore, we are planning for the following publications which use the content presented in this thesis:

1. The content in Chapter 5 – 9 will be integrated and submitted in a paper titled "Robust Performance Evaluation of Large-scale Resource-Bound Computer Systems Using the Method of Conditional Moments" to a Journal.

2. The content in Chapter 10 will be submitted in a paper titled "Robust Analysis of Two-Tier Wireless Networks Using the Method of Conditional Moments" to a suitable conference.

(*Alireza Pourranjbar*)

Table of Contents

List of Figures

List of Tables

Table of Notations

$\mathbb{M}$	a large-scale resource-bound model.
$\mathbb{M}^{agg}$	the aggregated model which captures the behaviour of $\mathbb{M}$'s small groups only.
D	the state space of the model $\mathbb{M}$.
D^{agg}	the state space of the aggregated model $\mathbb{M}^{agg}$.
$C_{\mathbb{M}}$	the set of sequential processes defined in model $\mathbb{M}$.
N_{agg}	number of states in the aggregated state space.
$\mathbb{Y}_{\boldsymbol{\gamma}}$	a sub-chain in the state space where $\boldsymbol{\gamma}$ is the configuration of the small groups across the states in $\mathbb{Y}_{\boldsymbol{\gamma}}$.
$\mathbb{Y}_{\mathbb{M}}$	the set of sub-chains in the model $\mathbb{M}$'s underlying state space.
$\mathcal{G}(\mathbb{M})$	the set of group labels in $\mathbb{M}$.
$\mathcal{G}_s(\mathbb{M})$	the set of labels of the groups that are small.
$\mathcal{G}_l(\mathbb{M})$	the set of labels of the groups that are large.
$\vec{\mathcal{A}}^*(\mathbb{M})$	the set of action types defined in the model.
$\vec{\mathcal{A}}^*_s(\mathbb{M})$	the set of action types that affect the state of only the small groups.
$\vec{\mathcal{A}}^*_{sl}(\mathbb{M})$	the set of action types that affect the state of both small and large groups.
$\vec{\mathcal{A}}^*_l(\mathbb{M})$	the set of action types that affect the state of only the large groups.
$\Delta_{\mathcal{G}}$	a partition on the set $\mathcal{G}(\mathbb{M})$.
$\Delta_{\mathcal{A}}$	a partition on the set $\vec{\mathcal{A}}^*(\mathbb{M})$.
H	a group label which is a member of the set $\mathcal{G}(\mathbb{M})$.
$ds^*(H)$	the set of local states associated with the sequential processes which form group H.
$\xi(H, C_x)$	the state variable capturing the number of instances in group $H \in \mathcal{G}(\mathbb{M})$ that are in state $C_x \in ds^*(H)$.
$\xi_i(H, C_x)$	the state variable capturing the number of instances in group $H \in \mathcal{G}(\mathbb{M})$ that are in state $C_x \in ds^*(H)$ when the system is in state S_i.

$\boldsymbol{\xi}(H)$	the numerical vector capturing the state of a group $H \in \mathcal{G}(\mathbb{M})$.
$\boldsymbol{\xi}_i$	the numerical vector capturing the state of the whole model when it is in state $S_i = \langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l \rangle$.
$\boldsymbol{\xi}^s$	the vector of state variables related to the small groups in a model $\mathbb{M}$.
$\boldsymbol{\xi}^l$	the vector of state variables related to the large groups in a model $\mathbb{M}$.
$\mathbb{P}_t(S)$	the probability that at time t, the system is in the state S.
$\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$	the conditional expectation of the vector $\boldsymbol{\xi}^l$ at time t given that $\boldsymbol{\xi}^s = \boldsymbol{\gamma}$.
$\mu(\boldsymbol{\gamma}, t)$	the conditional expectation of the vector $\boldsymbol{\xi}^l$ at time t given that $\boldsymbol{\xi}^s = \boldsymbol{\gamma}$.
$\mathbb{E}_{(H,C),t}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$	the element in the vector $\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$ associated with the state variable $\xi(H,C)$.
$\mu_{(H,C)}(\boldsymbol{\gamma}, t)$	a shorthand for $\mathbb{E}_{(H,C),t}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$.
$\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t)$	a higher-order moment of $\boldsymbol{\xi}^l$ associated with the moment vector $\mathbf{I}$ and the aggregated state $\boldsymbol{\gamma}$.
$\mathbf{I}$	the vector representing a conditional moment associated with $\boldsymbol{\xi}^l$, $\mathbf{I} = \langle I_{(H,C_x)} \rangle$, $\xi(H,C_x) \in \boldsymbol{\xi}$.
$\mathbf{e}_i$	a unit vector of size $n(\mathbb{M}, l)$ where all elements are zero except the i-th, which is one.
$\mathcal{V}_\alpha$	the impact vector associated with an action type α.
$\mathcal{V}_\alpha^-$	the vector representing the state variables which decrease as a result of transition of type α.
$\mathcal{V}_\alpha^+$	the vector representing the state variables which increase as a result of transition of type α.
$\mathcal{V}_\alpha^{s,-}$	the vector representing the state variables related to the small groups which decrease as a result of a transition of type α.
$\mathcal{V}_\alpha^{s,+}$	the vector representing the state variables related to the small groups which increase as a result of a transition of type α.
$\mathcal{V}_\alpha^{l,-}$	the vector representing the state variables related to the large groups which decrease as a result of a transition of type α.

$\mathcal{V}_{\alpha}^{l,+}$	the vector representing the state variables related to the large groups which increase as a result of a transition of type α.
$\mathcal{V}_{\alpha,(H,C)}$	the element in the impact vector $\mathcal{V}_{\alpha}$ related to the state variable $\xi(H,C)$.
$\mathcal{V}_{\alpha}^{s}$	the vector capturing the impact of an action type α on the vector $\boldsymbol{\xi}^{s}$.
$\mathcal{V}_{\alpha}^{l}$	the vector capturing the impact of an action type α on the vector $\boldsymbol{\xi}^{l}$.
$\mathcal{V}_{\alpha,(H,C)}^{l}$	the element in $\mathcal{V}_{\alpha}^{l}$ corresponding to the state variable $\xi(H,C) \in \boldsymbol{\xi}^{l}$.
$n(\mathbb{M},l)$	the number of state variables capturing the configuration of the large groups in $\mathbb{M}$.
$n(\mathbb{M},s)$	the number of state variables capturing the configuration of the small groups in $\mathbb{M}$.
$n(H)$	the number of instances in the group H.
$en_{\alpha}(P)$	the local derivative of sequential process P which enables action α.
$en_{\alpha}(G)$	the set consisting of pairs of a group label and a local state. Each pair denotes a local derivative within instances of G that enable action α.
$T(\boldsymbol{\xi}^{l},t)$	a polynomial function representing a reward associated with the state variables in $\boldsymbol{\xi}^{l}$ at time t.

Chapter 1

Introduction

1.1 Motivation

Recent advances in communication technology have enabled us to design and build computer systems with unprecedented *scales*. In contrast with traditional information processing systems which performed on stand-alone and isolated machines, nowadays we envisage systems which inherently rely on the assumptions of inter-connectivity and information flow. The sheer scale of these systems gives rise to a number of key features. Their development path follows a bottom-up approach, as opposed to traditional top-down methods. Thus, a key part of their development concerns the composition of existing systems and platforms. Consequently, the parts that comprise the system are usually heterogeneous. In contemporary systems, control is highly decentralised; different parts of the system are managed by different teams and with different configurations and goals. The traditional boundary between the infrastructure and users is blurred. Users, who usually appear in different groups with different populations, have to be considered when designing the system and their behaviour is integrated with that of the infrastructure. Due to the presence of different entities (hardware, software, services and users) the successful operation of these systems depends on complex and multi-layered dependencies. In contrast with small-scale systems where the design is based on a predefined requirement specification and the behaviour must *deterministically* follow the specification, the behaviour of today's complex systems is *emergent*, continually changing and *stochastic*. Because of the distinct features associated with the scale, the software engineering community introduced the notion of *large-scale* systems [39].

This thesis identifies one important and broad class of large-scale systems referred to as *large-scale resource-bound* (LSRB) systems. This class consists of systems whose interacting entities are conceptually categorised as either being *resources* or *resource users*. The populations of users are usually significantly larger than the populations of resources. In these systems, the users introduce heavy contention on resources and inevitably, some components become the bottlenecks. The dynamics of these components significantly affect the system's overall behaviour and its quality of service. This conceptually defined class encompasses a wide range of systems. Examples include a cloud platform where clusters of servers serve millions of users under peak demand, data centres where the virtualised servers deal with the continuous stream of transactions or a router in the Internet, which is dealing with a continuous flow of incoming packets.

The usability of LSRB systems strongly depends on their performance. Having considered users as key participants, the performance is affected by the system's processes and architecture, resource provisioning, and user demand (load). The systems are expected to be robust, that is, the users' demand needs to be met both when the system is running in the optimal operational mode when there is normal user load and all resources are healthy and active, and also in sub-optimal circumstances, when the peak or bursty load arrives and some of the resources are in failure mode. To achieve this, continuous monitoring and adaptive resource allocation are intertwined in the systems' operation [4].

Resource procurement and maintenance is expensive. Therefore, for a LSRB system to achieve both its economic values and user satisfaction we need rigorous performance evaluation mechanisms that can scale with the system size; analyse the complex dependencies within its sub-components and also the relevant cost-performance trade-offs; and provide us with the system-wide optimal configurations and associated performance predictions. In this thesis, we are concerned with the development of such mechanisms.

The performance evaluation of LSRB systems profoundly benefits from the process of *formal and quantitative modelling* [88]. Using modelling, the complexity caused by the scale is tackled by the power of *abstraction*. The model is an abstract representation of a real system which captures its key dynamics of interest. The more details added to the model, the more complex and closer to reality it becomes. The performance analysis based on modelling then has two steps. First, the model is formally specified.

Second, the key aspects of the real system are studied via the relevant performance metrics derived from the analysis of the model. As formal models are unambiguous, they are amenable to automatic, complex and rigorous mathematical analyses, such as studying their temporal properties or solving cost-performance trade-offs, which are impossible to do manually.

1.2 Modelling Requirements

Quantitative modelling of computer systems gained substantial interest in the 60's with the developments in the field of queueing theory [64, 71]. The goal was to analyse, what in retrospect would be considered *simple*, information processing systems. Ever since, advances in our computational power have expanded the boundaries of modelling capabilities. However, at the same time the scale of systems has also continually increased, which constantly outstrips our capabilities. The distinct features of LSRB systems impose the following requirements for our quantitative modelling processes:

1. Efficient model construction. The model needs to be built in a timely manner and is required to be maintainable. Compositionality is key to achieving these goals, as it simplifies the construction of complex models and allows us to alter sub-components without the need to change the whole model.

2. Faithfulness in specification. The behaviour of sub-components, which are divided into different layers (hardware, software, user), and their emergent evolutions are all stochastic. The model should allow us to express and capture stochastic evolutions at all layers and easily specify the probabilistic decisions made within the sub-components [39].

3. Efficiency in analysis. The modelling framework needs to be equipped with analysis methods which can, using the existing computational capacities, efficiently evaluate our models. Often, the model is even required to be analysed quickly and within few seconds to assist real-time decision making [1].

4. Faithfulness in analysis. The analysis output should provide information that captures key aspects of the system's evolution in sufficient detail [39]. It is imperative to reveal the performance for both the optimal and sub-optimal configurations. The analysis is required to show how the failures propagate through the

system, and in particular, how the user experience is affected.

The above requirements have triggered extensive research in modelling. In response to the first two requirements, high level modelling languages or formalisms such as queueing networks [17], Petri nets [80, 9, 97] and process algebras [75, 62, 48, 13] have been proposed. Furthermore, there have been many evaluation techniques such as enhanced algorithms for analysis of discrete and continuous time Markov Chains [90, 29]; discrete event stochastic simulations [44, 27, 28]; and deterministic approximation (based on mean-field theory) [11, 16] for the last two requirements. One important outcome of this research is the evaluation framework based on *performance evaluation process algebra* or shortly known as PEPA [58].

PEPA is a compositional stochastic process algebra and it has promising features for the construction of models of large-scale systems. The support for compositionality makes PEPA more appealing when compared to queueing networks and Petri nets. In PEPA, models are constructed by defining sequential processes which undertake stochastically timed actions with the possibility of probabilistic choice. This allows us to express and capture timed stochastic evolution at the layer of the system's constituent entities and also its emergent behaviour. Furthermore, PEPA is formal and has been equipped with semantics in terms of *continuous time Markov chains* (CTMC), stochastic simulation and ordinary differential equations [94]. These features identify PEPA as a suitable candidate for performance evaluation of LSRB systems.

1.3 Statement of Problem

One area in which the PEPA framework can be fruitfully extended is in its analysis capabilities to enable efficient and faithful evaluation for LSRB models. The major analysis methods that can currently be applied, at least in theory, are divided into three categories: exact Markovian analysis based on the construction of the complete underlying discrete state space; sampling methods based on Gillespie's stochastic simulation; and deterministic approximation based on mean-field theory and fluid flow approximation [59]. The shortcomings of these methods when applied in practice are the following:

1. Exact Markovian Analysis. PEPA has semantics in terms of labelled transition systems. Using the exact method, the model's underlying discrete state space

is constructed. This is treated as the state space of a CTMC and is analysed by numerical methods to find the evolution of the model's underlying probability distribution. Not only is the complete state space generated, but also the probability of being in every single state is derived. Thus, the method offers the highest degree of analytical faithfulness. However, the problem of state space explosion hampers its practicality; since for large-scale models the state spaces are usually massive, the analysis becomes very computationally expensive or even infeasible. This means that the method is very poor in terms of efficiency.

2. Methods Based on Stochastic Simulation. The model is executed programatically to produce numerous realisations or trajectories across its state space. The trajectories are then used to derive the evolution of an *approximate* probability distribution with respect to the exact one. The more simulation runs are performed, the more accurate and faithful the approximation becomes. Applying this method has two shortcomings. First, it is computationally expensive; in order to obtain accurate results a very large number of simulations need to be run. This is exacerbated when the system has dynamics at multiple time scales and is affected by rare events. Second, there is an intrinsic unavoidable error associated with simulation. The observations made on generated trajectories do not provide any guarantee of *what has not been observed*. The evidence that an event has not been observed in past trajectories by no means implies its impossibility [91]. For instance, the transitive implications (knock-on effects) of a failure may not be witnessed at all due to its low probability and hence, not having covered the state space path that triggers the failure. In this method, one cannot determine if enough simulations have been run unless there is firm a priori knowledge about the system's behaviour. These features mean that although stochastic simulation can be partially useful, it does not guarantee a faithful analysis in the general case.

3. Deterministic Approximation of Stochastic Evolution. The first method in this class is based on the mean-field theory and in particular the well-known Kurtz *convergence theorem* [67]. The theorem is used to prove that in a system with interacting entities, when their populations tend to infinity, the stochastic behaviour tends to a deterministic limit behaviour. The limit can be obtained by solving a set of ordinary differential equations (ODEs) and therefore, the analysis is efficient. However, in realistic large-scale systems the assumption of having

infinitely large populations is not fully adhered to. In these systems we have different types of entities with different population scales, and often the emergent behaviour is highly stochastic. Therefore, analyses based on the deteministic limit behaviour is not faithful in capturing the randomness.

The second method in this class is the fluid flow approximation and is closely related to moment closure techniques [59, 56]. The method is based on a simplifying assumption that the probability distribution underlying a large-scale system are globally and densely clustered around the average[1] and it can be faithfully and sufficiently captured by its first few moments such as mean, variance, skewness, etc. The model is analysed by solving a set of ODEs derived directly from the model and the solution (moments) provides the basis for characterising the underlying distribution. The method is efficient, but the simplifying assumption imposed on the probability distributions is again potentially too crude or abstract to reflect many realistic scenarios. The scale and randomness in large-scale systems makes them inevitably exhibit distinct performance profiles related to optimal and sub-optimal modes of operation, across which users experience different levels of quality of service. As a result, the underlying distribution potentially has arbitrary shapes, including multiple modes (multi-modal distributions) and heavy tail. In such cases, the fluid flow moments fail to faithfully capture the highly stochastic dynamics and may lead to misjudged performance decisions about the reliability of the system. The analysis based on the dynamics being clustered around a single average is not faithful.

In this thesis, we consider these problems and extend the PEPA framework with new analysis methods which are particularly suitable for performance evaluation of LSRB systems. Our methods collectively offer a high degree of efficiency and faithfulness. The summary of our contributions follows.

1.4 Summary of Contributions

Our first contribution is to provide evidence of the shortcomings of existing analysis methods for performance evaluation of LSRB systems. This highlights the benefits of the new analysis methods that we present as our next contributions.

[1] Similar to bell-curved normal distributions where the probabilities exponentially decrease as we move away from the distribution's mean.

1.4.1 The Issue of Faithfulness and Efficiency

In this contribution (published in [79]), we consider a simple scenario for a client-server system, where a small population of servers (resources) are serving a relatively larger population of clients (resource users) under contention. The servers are susceptible to failures and the system has multiple operational modes related to the different number of broken servers. The underlying probability distribution is multi-modal, which cannot be qualitatively or quantitatively captured by its moments. We show that both the exact Markovian analysis and stochastic simulations are computationally expensive, and that fluid flow moments are too crude to capture the highly stochastic emergent behaviour. Using the result of this experiment, we show that existing criteria in the literature that determine the appropriate analysis based on the model are not robust. The experiment provides an intuitive basis for our next contributions.

Next, we propose four new analysis methods for LSRB models.

1.4.2 An Aggregation Method for LSRB Models

State space aggregation is an effective technique for tackling the problem of state space explosion. One class of aggregation methods develops notions of *state equivalence* and then, when applied to a large state space, they generate more compact state spaces where the states which were equal are captured by a single canonical state [25]. This results in a reduced computational cost in analysis. Another interesting class consists of methods that develop notions of *behavioural equivalence* at the level of modelling languages and based on their syntax and structural attributes [32, 37, 81, 60]. For a conforming model, its aggregated model is constructed which can then be used to directly generate the aggregated state space. Here, the generation of the complete state space is fully bypassed. In our first contribution, we propose an aggregation technique which falls in the second class and aggregates LSRB PEPA models (published in [78]).

In LSRB systems, the small populations of resources deal continually with the relatively larger populations of resource users, and the probability of observing situations where resources are waiting idle for users is close to zero. This assumption forms the basis of our aggregation technique. From the original model, this method generates an aggregated model which abstracts from the behaviour of resource users and captures the dynamics of resources only. The aggregated model is analysed to obtain a marginal

probability distribution over all configurations that the resources may experience. This distribution provides information sufficiently detailed on the dynamics of resources and can be used to perform a faithful resource analysis, beyond deriving only their average evolution. For instance, it can be used to derive important reliability measures; to detect the resources' significant optimal, sub-optimal and critical operational modes; and to analyse the service throughput offered to the users. The technique is applicable to both the transient and steady state analyses.

1.4.3 Efficient Detection of Time-Scale Decomposability

LSRB systems can exhibit dynamics and transitions that take place across different time scales. When analysing such a model, it is important to detect if its transitions can be categorised onto a slow / fast spectrum, the condition formally known as *time-scale near-complete decomposition* (TSND) [87]. The importance is two-fold. First, it alerts us to analyse the evolution comprehensively for both time-scales and incorporate both the short and long term phenomena. Second, it opens the possibility of using enhanced analysis methods, which by exploiting this feature can efficiently and accurately, approximate the model's exact underlying distribution. Examples of such techniques can be found in the queueing network literature [33] and in the stochastic process algebra literature [61, 73]. Automatic detection of TSND in a LSRB model requires an upfront reachability analysis of its state space. Since this state space is massive, the reachability analysis becomes computationally expensive. This prohibits our evaluation from taking advantage of the enhanced solution methods.

As our second contribution, we propose a scalable and efficient algorithm for efficient detection of TSND condition in the context of LSRB models. This algorithm is based on hierarchical bottom-up clustering [54] and efficiently gives us a partition on the model's actions dividing them into slow and fast. We present a theorem that supports our approach. We show how our clustering tool can be embedded within a decomposability-based solution method presented in [61] that significantly improves the efficiency of the analysis of LSRB models. This contribution aids the model analysis when the equilibrium behaviour is being investigated.

In the next contributions, we present the method of conditional moments for analysis of LSRB models. This analysis has two parts: analysis of conditional expectations (conditional means or averages) and analysis of higher-order conditional moments.

Our analysis method is closely related to a class of methods developed for biological processes in [53, 72, 38], and is particularly useful for deriving the models' transient behaviours[2].

1.4.4 Analysis Using Conditional Expectations

For LSRB models, it is computationally too costly to study the behaviour of users by deriving the model's complete underlying probability distribution. Instead, using the method of conditional moments we study their stochastic behaviour through the *moments* of the *conditional probability distributions* associated with users *given the different configurations of resources*. Expectations or averages are regarded as the first-order moments of a probability distribution. In the analysis of first-order moments, we derive the conditional expectations related to the stochastic evolution of users given the stochastic evolution of resources, which is captured by the aggregated state space. In contrast with the fluid flow approximation, where the evolution of users is studied by a single expectation, here we calculate many expectations corresponding to the different configurations that the resources experience.

The input is the model and the aggregated state space. These are combined to construct a set of differential algebraic equations (DAEs) whose solution is the transient evolution of conditional expectations. A *mode of operation* is defined as a subset of states within the aggregated state which satisfy a property of interest (for instance, number of resources in failure mode). We show that the DAE solution can be used to derive coarse-grained conditional expectations for any mode of operation. We illustrate that the analysis of conditional expectations can capture the potentially many locations within the complete state space where significant probability masses are clustered. This information is highly useful for qualitative and quantitative representation of the system's underlying probability distribution.

1.4.5 Analysis Using Higher-Order Conditional Moments

The stochastic behaviour of users potentially makes deviations from the conditional expectations. As our fourth contribution, we expand the analysis of conditional moments by including *higher-order conditional moments*. These include the conditional

[2]Our future work will investigate the applicability of the method to the case of steady state analysis.

variances, conditional skewness, etc. given the different configurations of resources. The higher-order moments enable us to obtain a richer representation of the conditional distributions stated above. The derivation of these moments is critical in tasks such as capacity planning, where in addition to the most expected path (average) we also need to know how the system deviates from that path.

Similarly to the previous case, we illustrate how the DAEs related to higher-order moments can be automatically derived from the model. This set of DAEs is larger than that constructed for conditional expectations; as the order increases, the set of equations is augmented with more equations and finding the solution becomes computationally more expensive. We provide an analytical expression for the size of this set of equations and show that, given the capabilities of current DAE solvers, the analysis for up to the third-order (conditional skewness) is practically possible for most models. We illustrate the usefulness of the technique of conditional moments in the context of a simple scenario of a client-server system, and also for a more complex model where we study the dynamics of a two-tier wireless network, whose construction is based on the femto-cell macro-cell architecture [26].

1.5 Thesis outline

The thesis is structured as follows:

- In Chapter 2, we introduce PEPA.
- In Chapter 3, we consider the issue of analysis faithfulness and in the context of an experiment show in detail the shortcomings of the existing analysis methods.
- In Chapter 4, we show our aggregation method.
- In Chapter 5, our algorithm for the detection of time-scale decomposability features is presented.
- In Chapter 6, we describe our analysis of conditional expectations.
- Chapter 7 shows an example, where the analysis of conditional expectations is applied to the evaluation of our client-server model.
- In Chapter 8, we present the analysis of higher-order conditional moments.

- In Chapter 9, we show the application of the analysis of higher-order conditional moments to the client-server system.
- In Chapter 10, as a case study we consider a more complex model related to a two-tier wireless network and show the usefulness of the analysis of conditional moments in a wider context.
- Chapter 11 describes our conclusion and directions for future work.

Chapter 2

Preliminaries

The practice of quantitative modelling is supported by a number of modelling languages. One appealing option is Performance Evaluation Process Algebra, abbreviated as PEPA. In our first chapter, we provide an overview of this language. To cope with the increasing complexity of evaluation of modern systems, the PEPA framework has been continually extended. A number of the recent extensions that are relevant to modelling LSRB systems will be described. This description provides preliminary definitions to be used in the next chapter where we explain the key challenges confronted when evaluating LSRB systems.

2.1 PEPA Language

In PEPA, a system is modelled by specifying the behaviour of its constituent components and the way they interact. In order to specify this information, we use process combinators. We review PEPA combinators in the next section.

2.1.1 PEPA Combinators

PEPA offers the following combinators:

1. **Prefix**: This is represented by "." and is used to express the sequence of activities; the term $(\alpha, r).P$ represents a process which undertakes an activity of type α with rate r and then moves into the state P. Each activity is specified as a pair (α, r) where α is its action type and r is its rate, which describes how fast the

activity is performed. The time it takes to perform an activity (α, r) is captured by an exponentially distributed random variable with parameter r; the activity takes on average $\frac{1}{r}$ time units to finish.

2. **Choice**: The choice combinator allows us to describe a behaviour which includes probabilistic choices and branching; the term $P + Q$ represents a process that exhibits the behaviour of either process P or process Q. Thus, $P + Q$ can perform any activity that P or Q enable. However, the first activity that is finished identifies the change that the process experiences.

3. **Cooperation**: Computer systems are usually composed of different types of components that work together and interact. The interactions are expressed using the cooperation combinator. The term $P \underset{L}{\bowtie} Q$ represents a process composed of two processes P and Q that cooperate on action types within set L, also known as the *cooperation* set. An action type $\alpha \in L$ is enabled only if both P and Q simultaneously enable it (through their α activities). When a shared action α is enabled, the participants can simultaneously perform their corresponding α activities and as a result, their internal states change. The actions that a component enables can be divided into being *individual* or *shared*. For an individual action the component can proceed independently. For a shared one, it needs to wait until the other participants also enable the action type.

   When two or more components undertake a shared activity, the rate at which the activity is performed is determined by the principle of *bounded capacity*. This states that no component should be made to perform an activity faster than its capacity (its rate). Therefore, the rate assigned to the shared activity is defined to be the *minimum* of the rates that the participants offer. For instance, in a system where a buffer is connected to a processor which continually reads the buffer, the rate of the data transfer is determined by the component which is slower.

   A PEPA component can participate *passively* in a cooperation, denoted by the rate $\top$, meaning that it is willing to cooperate at the rate decided by other participants. In the final model, cooperations that involve passive components must include at least one component that is not passive. Otherwise, the model is said to be incomplete.

   Two components can run in parallel without cooperating. This can be specified as $P \underset{\emptyset}{\bowtie} Q$ where the cooperation set is empty. As syntactic sugar, this is expressed

as $P \parallel Q$. The combinator is especially useful when we have repeated replicas of the same component which do not interact with each other and only interact with the components in the rest of the system.

4. **Constant Definition**: The complex process expressions that we build can be referred to by a short name. The combinator $\stackrel{def}{=}$ is used for this purpose. The term $E \stackrel{def}{=} F$ assigns the behaviour captured by F to the constant process E. This combinator is also used for defining cyclic behaviours. For instance, $F = (\alpha, r_1).(\beta, r_2).F$ is a process which cycles through activities (α, r_1) and (β, r_2).

5. **Hiding**: The activities that a component undertakes are visible, in terms of their action type and durations, to the rest of the components within the model. The hiding combinator is used to hide activities from being globally visible. The term P/L represents a process whose activities with types in L are only visible to P and hidden externally. Hidden activities still cause time delays but their action types become unknown. There is a special type τ associated with such activities, which would be witnessed by external components. It is assumed that τ cannot appear in any of the cooperation sets.

In almost all performance evaluation studies carried out with PEPA, the first four combinators (prefix, choice, cooperation, constant) were sufficient to build the models, and the last one was rarely used. In this thesis, we are also restricting ourselves to the first four.

2.1.2 PEPA Syntax

A common approach for building a PEPA model is to first specify the behaviour of the constituent sequential processes using prefix, choice and constant, and then use the cooperation combinator to compose them to build the description of the system as a whole. A two level grammar defining the syntax of PEPA supports this approach:

$$\begin{aligned} S &::= (\alpha, r).S \mid S + S \mid C_S \\ P &::= P \underset{L}{\bowtie} P \mid C \mid S \end{aligned} \tag{2.1}$$

where S represents a sequential process, C_S stands for a process constant which represents a sequential process, P stands for the composition of components, and C represents a constant that specifies either a sequential or composed process. A composed process is also referred to as a *model process*.

Example

Let us present an example of a PEPA model. The following model captures the dynamics of a client-server system. In spite of being simple, this model is capable of exhibiting numerous important and interesting features that will be introduced as we proceed in the thesis.

The system consists of two types of components: clients and servers. A client is represented by the process C_{think} and a server is represented by the process S_{idle}. The behaviour of a system with three clients and two servers is expressed by the following PEPA model:

$$\begin{aligned} C_{think} &\stackrel{def}{=} (think, r_t).C_{req} \qquad & C_{req} &\stackrel{def}{=} (req, r_c).C_{think} \\ S_{idle} &\stackrel{def}{=} (req, r_s).S_{log} \qquad & S_{log} &\stackrel{def}{=} (log, r_l).S_{idle} \end{aligned}$$

$$CS \stackrel{def}{=} (\, S_{idle} \,||\, S_{idle} \,) \underset{\{req\}}{\bowtie} (\, C_{think} \,||\, C_{think} || \, C_{think} \,) \tag{2.2}$$

A client starts in the state C_{think} where it is performing some independent thinking. This can reflect doing some offline computational job which does not require a server. The thinking activity is performed at rate r_t. When the activity is done, the client moves into the state C_{req} where it requests service from a server. The client can perform the action *req*, which represents communication with a server, at rate r_c. When the communication is over, the client returns to its initial state and repeats the above cycle.

A server starts in state S_{idle} where it is waiting and ready to serve a client. An idle server can perform the action *req* at rate r_s. When the server has finished serving a client, it enters the state S_{log} where it logs the service just provided. This happens at rate r_l. When logging is done, the server returns to state S_{idle}.

The last line of the model is called the *system equation*. Here, the behaviours defined for the sequential components C_{think} and S_{idle} are composed together and the action types on which instances of those processes cooperate are specified. In our example, the clients and servers synchronise on activities of type *req*. The instances of clients and those of the servers do not cooperate within themselves.

We can replace the behaviour of a client as:

$$C_{think} \stackrel{def}{=} (think, r_t).C_{req} \qquad C_{req} \stackrel{def}{=} (req, \top).C_{think}$$

This suggests that while cooperating with a server, a client undertakes *req* passively and the rate is determined by the server. To illustrate the use of the choice combinator, we change the behaviour of a server as follows:

$$S_{idle} \stackrel{def}{=} (req, r_s).S_{log} + (brk, r_b).S_{broken}$$

$$S_{log} \stackrel{def}{=} (log, r_l).S_{idle} \qquad S_{broken} \stackrel{def}{=} (fix, r_f).S_{idle}$$

In this system servers are susceptible to failure; a server that is idle either serves a client or breaks down (activity (brk, r_b)). The choice is resolved by a *race condition*. Assuming that the server makes a transition, the activity (req, r_s) occurs with probability $\frac{r_s}{r_s+r_b}$ and the activity (brk, r_b) with probability $\frac{r_b}{r_s+r_b}$.

2.2 Structured Operational Semantics

The structured operational semantics (SOS) of PEPA are a set of rules that determine the set of activities that a process enables and how the process changes as a result of performing them. When the rules are repeatedly applied, we derive a *labelled transition system* (LTS) which captures the states that the process visits. In order to present the SOS first we introduce the notion of *apparent rates*.

2.2.1 Apparent Rate

A process, whether sequential or composed, can enable multiple activities that have the same action type ($P = (\alpha, r_1).P' + (\alpha, r_2).P'' + \cdots$). We define the notion of *apparent rate*, as a way to describe the maximum capacity that a process offers for an action type. This capacity is observed from the point of view of an external observer or component.

Definition 1. *(*Apparent rate of an action type with respect to a process*) The apparent rate of an action type* α *with respect to the process P is denoted as* $r_\alpha(P)$ *and is defined*

as the total rate that P offers for activities of type α *:*

$$r_\alpha(P) = \begin{cases} 0 & P \equiv (\beta, r).P' \ , \ \alpha \neq \beta \\ r & P \equiv (\alpha, r).P' \\ r_\alpha(Q) + r_\alpha(R) & P \equiv Q + R \\ r_\alpha(Q) & P \text{ is a constant: } P \stackrel{\text{def}}{=} Q \\ \min(r_\alpha(Q), r_\alpha(R)) & P \equiv Q \underset{L}{\bowtie} R \ \text{ and } \ \alpha \in L \\ r_\alpha(Q) + r_\alpha(Q) & P \equiv Q \underset{L}{\bowtie} R \ \text{ and } \ \alpha \notin L \end{cases} \tag{2.3}$$

In the next definition, we define the capacity of a process P with respect to α activities that take P to a particular state P' :

Definition 2. (Apparent rate of an action type with respect to two states) *We define* $r_\alpha(P,P')$ *to be the apparent rate of an action type* α *in P restricted to* α *activities leading P to* P' *:*

$$r_\alpha(P,P') = \begin{cases} r & \text{if } P \equiv (\beta, r).Q \ \wedge \ (\beta = \alpha \ \wedge \ P' \equiv Q) \\ 0 & \text{if } P \equiv (\beta, r).Q \ \wedge \ (\beta \neq \alpha \ \vee \ P' \not\equiv Q) \\ r_\alpha(Q,P') + r_\alpha(R,P') & \text{if } P \equiv Q + R \end{cases}$$

P might be passive with respect to α and enable more than one passive α activity. In such a case, each activity is given a weight ($\omega \in \mathbb{N}$), clarifying its relative probability with respect to all such activities. These are handled appropriately in the passive rate arithmetic:

1. $(\omega_1 \top + \omega_2 \top) = (\omega_1 + \omega_2)\top \qquad \forall \ \omega_1, \omega_2 \in \mathbb{N}$
2. $\omega_1 \times (\omega_2 \top) = (\omega_1 \times \omega_2)\top$
3. $\top = 1 \times \top$
4. $\min(\omega_1 \times \top, \omega_2 \times \top) = \min(\omega_1, \omega 2) \times \top$

2.2.2 Semantic Rules

The rules of PEPA's SOS are shown in Fig. 2.1. As an example, we describe the cooperation rules: co_1, co_2 and co_3. Rule co_1 states that the processes that are constrained within a cooperation can freely and independently proceed with the action types that are not in the cooperation set. When two processes E and F synchronise on actions

Prefix

$$\frac{}{(\alpha, r).E \xrightarrow{(\alpha,r)} E}$$

Choice

$$ch_1 : \frac{E \xrightarrow{(\alpha,r)} E'}{E + F \xrightarrow{(\alpha,r)} E'}$$

$$ch_2 : \frac{F \xrightarrow{(\alpha,r)} F'}{E + F \xrightarrow{(\alpha,r)} F'}$$

Constant

$$\frac{E \xrightarrow{(\alpha,r)} E'}{A \xrightarrow{(\alpha,r)} E'} \; (A \stackrel{def}{=} E)$$

Cooperation

$$co_1 : \frac{E \xrightarrow{(\alpha,r)} E'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha,r)} E' \underset{L}{\bowtie} F} \; (\alpha \notin L)$$

$$co_2 : \frac{F \xrightarrow{(\alpha,r)} F'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha,r)} E \underset{L}{\bowtie} F'} \; (\alpha \notin L)$$

$$co_3 : \frac{E \xrightarrow{(\alpha,r_1)} E' \quad F \xrightarrow{(\alpha,r_2)} F'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha,R)} E' \underset{L}{\bowtie} F'} \; (\alpha \in L)$$

where $R = \frac{r_1}{r_\alpha(E)} \frac{r_2}{r_\alpha(F)} \min(r_\alpha(E), r_\alpha(F))$

Figure 2.1: PEPA structured operational semantics

within set L, if E can perform (α, r) (this is stated as $E \xrightarrow{(\alpha,r)} E'$) with $\alpha \notin L$, then the composed process can do the same activity where E changes its state and F remains unchanged. Rule co_2 is the symmetric version of co_1 related to the second process F. Rule co_3 states that the cooperating processes change synchronously when undertaking shared activities. If E can perform (α, r_1) and become E', and F does (α, r_2) and becomes F', and $\alpha \in L$, then we infer that the composed process $E \underset{L}{\bowtie} F$ can perform the activity (α, R), $R = \frac{r_1}{r_\alpha(E)} \frac{r_2}{r_\alpha(F)} \min(r_\alpha(E), r_\alpha(F))$, and as a result both E and F change their state.

The calculation of the rate R reflects the fact that in the most general case each of the processes E and F might offer multiple activities of type α; E can be $E = (\alpha, r_1).E' + (\alpha, r_2).E'' + \cdots$ and F can be $F = (\alpha, r'_1).F' + (\alpha, r'_2).F'' + \cdots$. Thus $P \underset{L}{\bowtie} Q$ may evolve through multiple α activities, one of which is $E \to E'$ and $F \to F'$. Each cooperating process independently decides which α activity it performs within the

cooperation. The bounded capacity principle requires that the capacity of $P \underset{L}{\bowtie} Q$ in offering α is $\min(r_\alpha(P), r_\alpha(Q))$ where $r_\alpha(P)$ and $r_\alpha(Q)$ are respectively the capacities (apparent rates) of α within P and Q. This capacity is also the sum of the rates of all one step α activities $P \underset{L}{\bowtie} Q$ enables. The rate R associated with activity $P \underset{L}{\bowtie} Q \xrightarrow{(\alpha,r)} P' \underset{L}{\bowtie} Q'$ is a proportion of $\min(r_\alpha(P), r_\alpha(Q))$ and is affected by the probability that the particular activities $P \xrightarrow{(\alpha,r_1)} P$ and $Q \xrightarrow{(\alpha,r_2)} Q'$ happen among all plausible α activities. The probability of P performing its activity is $\frac{r_1}{r_\alpha(P)}$ and similarly, the probability of F performing its part is $\frac{r_2}{r_\alpha(F)}$. Given the independence of these components, the total rate associated with the shared activity is $R = \frac{r_1}{r_\alpha(E)} \frac{r_2}{r_\alpha(F)} \min(r_\alpha(E), r_\alpha(F))$.

2.2.3 Labelled Transition System

Having reviewed the semantics, now we formally characterise the underlying LTS of a model. For this purpose, we need the following definition:

Definition 3. (Derivative set of a process) *The derivative set of a process P is denoted as $ds(P)$ and is the smallest set of processes such that:*

- *if $P \stackrel{\text{def}}{=} P'$, then P' is in $ds(P)$.*
- *if $P_i \in ds(P)$ and there exists an activity $a = (\alpha, r)$ such that $P_i \xrightarrow{a} P_j$, then P_j is in $ds(P)$.*

Intuitively, the derivative set of a process P consists of the states that the process may visit. When P is one of the model's sequential processes, we call $ds(P)$ the set of *local* states or *local derivatives* of P. Using Def. 3, we characterise the LTS that is constructed for a PEPA model.

Definition 4. (Labelled transition system of a process) *For a process P with the derivative set $ds(P)$, the labelled transition system LTS_P is defined as the tuple $\langle S, \Lambda, \rightarrow \rangle$ where $S = ds(P)$ is the set of states, Λ is the alphabet of the activities enabled by such states, and $\rightarrow \subseteq S \times \Lambda \times S$ is the transition relation. For two states $S_i, S_j \in ds(P)$, we have $(S_i, (\alpha, r), S_j) \in \rightarrow$, if and only if, there exists an activity (α, r) connecting S_i to S_j. For each distinct activity that takes S_i to S_j, one element is added to $\rightarrow$.*

The original definition of a model's LTS in [58] constructs a muti-relation $\rightarrow$. However, throughout this thesis we restrict ourselves to the class of split-free models which

give rise to relations with no duplicate transitions. This class of models will be formally introduced in Sec.2.5.

For a model $\mathbb{M}$, the $LTS_{\mathbb{M}}$ represents its state space. This can be presented as a graph, where the nodes are the members of $ds(C)$ and the arcs are the elements of the relation $\rightarrow$. For every activity that takes a state S_i to S_j a separate arc is added. This graph is referred to as the *derivation graph*. The node corresponding to $\mathbb{M}$'s system equation is regarded as the initial node.

2.3 Underlying Continuous Time Markov Chain

The analysis of a model written in a high-level modelling language involves the derivation of its underlying stochastic process. A PEPA model is mapped to its underlying continuous time Markov chain (CTMC). In this section, we review this mapping.

2.3.1 Continuous Time Markov Chain

A CTMC is a family of random variables $X = \{X_t : t \geq 0\}$ which take values from a countable set of states S and are indexed by time $t \in \mathbb{R}_{\geq 0}$. The behaviour of a CTMC is illustrated by a graph where the nodes are the elements of S and the arcs are the transitions that X_t experiences as time proceeds. Each transition has a rate: if there exists a transition from i to j, $i, j \in S$, with rate r, then assuming that $X_t = i$, the time it takes for the transition to occur is an exponentially distributed random variable T with the distribution $\mathbb{P}(T < x) = 1 - e^{-rx}$. The transition rates between the pairs (i, j) are specified by the *infinitesimal generator matrix* $Q = (q_{ij}), i, j \in S$. Here, each off-diagonal element q_{ij}, $i \neq j$ is the rate of the transition from i to j. Each diagonal element q_{ii} is chosen as the *negative* sum of elements in row i : $q_{ii} = -\sum_{j \neq i} q_{ij}$.

For any time t, the behaviour of X_t is captured by a probability distribution $\Pi_t = \langle \mathbb{P}(X_t = i) \rangle$ defined over its state space. The analysis of a CTMC typically involves deriving the evolution of distributions Π_t or their important measures, such as their mean, variance, etc.

2.3.2 Mapping, From PEPA Model to CTMC

For a PEPA model $\mathbb{M}$ its underlying CTMC is denoted as $X_{\mathbb{M}}$ and can be constructed using $LTS_{\mathbb{M}}$ in the following way. Each process in $ds(\mathbb{M})$ is regarded as a state of $X_{\mathbb{M}}$. Thus, $S = ds(\mathbb{M})$. Moreover, the rate between any two states $i, j \in S$, which are respectively related to processes $P_i, P_j \in ds(\mathbb{M})$, is derived by *summing up* the rates r of all activities that take P_i to P_j:

$$\forall\, i, j\ ,\ i \neq j\ :\ q_{ij} = \sum_{(\alpha,r)\ :\ P_i \xrightarrow{(\alpha,r)} P_j} r \tag{2.4}$$

Theorem 3.5.1 in [58] is key in this mapping. The theorem expresses that the assumption that PEPA sequential processes undertake activities with exponentially distributed delays can be used to prove that the delays associated with the transitions of the underlying Markov chain are exponentially distributed. This means that the summation of Eq.(2.4) is valid in the sense that the model's temporal evolution within the states in $ds(\mathbb{M})$ accurately matches the evolution derived by the analysis of $X_{\mathbb{M}}$.

Example

Consider the client-server model where two clients are served by one server:

$$CS \ \stackrel{def}{=}\ S_{idle} \underset{\{req\}}{\bowtie} (\ C_{think} \ ||\ C_{think}\ ) \tag{2.5}$$

This reflects a system where clients *compete* over receiving the service. The derivation graph is shown in Fig. 2.2. Each state in this graph is regarded as a state of the model's underlying CTMC. For this model, there is a one-to-one relation between the activities of the derivation graph and the transitions of the CTMC. Thus, the state space of the model's CTMC is the same graph with the labels of the action types removed.

2.4 Models of Large-Scale Systems

In this section, we introduce some of the key extensions of the PEPA framework that are useful when building models of large-scale systems.

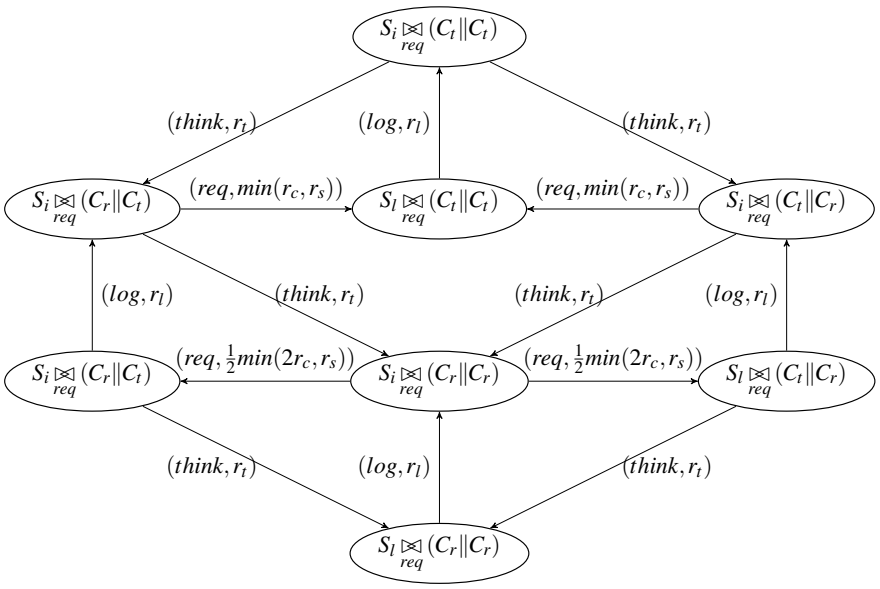


Figure 2.2: The state space of the client-server system with one server and two clients. These abbreviations are used: $S_i : S_{idle}$, $S_l : S_{log}$, $C_t : C_{think}$ and $C_r : C_{req}$.

2.4.1 Grouped PEPA Syntax

Large-scale systems are composed of components which are replicated many times to form populations. When modelling them, we are content to abstract away from the behaviour of individual components, and study the behaviour at the level of populations. For instance, consider the client-server system shown in Eq.(2.2) and assume that the system has 100 clients and 10 servers. When evaluating the system, we capture the *count* of the clients or servers in each of the states S_{idle}, S_{log}, C_{think}, C_{req}, as opposed to focusing separately on the state of each individual entity. Also, when modelling their interaction, we do not care exactly which client is being served by which server; the interactions are regarded as happening between the *group* of clients and the *group* of servers. The shift from the detailed focus on the evolution of individual entities to the higher level of populations is referred to as the *counting abstraction* [59].

The syntax of PEPA has been extended by the *Grouped PEPA* (GPEPA) syntax so that a model's populations can be explicitly expressed [56]. The GPEPA syntax is:

$$\begin{aligned} S &= (\alpha, r).S \mid S + S \mid C_S \qquad P = P \underset{L}{\bowtie} P \mid S \\ D &= D \mid\mid D \mid P \qquad\qquad M = M \underset{L}{\bowtie} M \mid Y\{D\} \end{aligned}$$

Here the terms S, C_S, P are the same as those in Eq.(2.1). D represents the *parallel* composition of instances of a sequential component. These instances constitute a

group and a unique label is assigned to each group, denoted by Y. M represents the composition of groups. We restrict ourselves to GPEPA models in which all groups are strictly *simple*; i.e. each group contains instances of one sequential component only. The models with groups of multiple sequential components can be easily mapped to their corresponding model in which groups are simple. In GPEPA models, we often use the syntactic sugar $P[n]$ to denote the parallel composition of n identical P processes, $P[n] = \underbrace{P \parallel P \parallel \ldots \parallel P}_{n \text{ times}}$.

Example

Here is a version of the client-server system where the groups of clients and servers are explicitly specified in the model, as group *Clients* and group *Servers*:

$$C_{think} \stackrel{def}{=} (think, r_t).C_{req} \qquad C_{req} \stackrel{def}{=} (req, r_c).C_{think}$$

$$S_{idle} \stackrel{def}{=} (req, r_s).S_{log} \qquad S_{log} \stackrel{def}{=} (log, r_l).S_{idle}$$

$$CS \stackrel{def}{=} Servers\ \{\ S_{idle}[10]\ \} \underset{\{req\}}{\bowtie} Clients\ \{\ C_{think}[100]\ \} \tag{2.6}$$

2.4.2 State Space Representation for Large-Scale Models

The state of a large-scale model is captured using numerical vectors with integer elements. In order to introduce these vectors, we need the following definitions.

Definition 5. (Set of groups of a model). *For a grouped PEPA model $\mathbb{M}$, we define $\mathcal{G}(\mathbb{M})$ to denote its set of group labels. This can be derived as:*

$$\mathcal{G}(\mathbb{M}) = \begin{cases} \mathcal{G}(M_1) \cup \mathcal{G}(M_2) & \text{if } \mathbb{M} \equiv M_1 \underset{L}{\bowtie} M_2 \\ Y & \text{if } \mathbb{M} \equiv Y\{D\} \end{cases} \tag{2.7}$$

Definition 6. (Sequential Component of a Group) *Let $\mathcal{C}_{\mathbb{M}}$ represent the set of sequential components defined in the model $\mathbb{M}$. We define function $sc : \mathcal{G}(\mathbb{M}) \to \mathcal{C}_{\mathbb{M}}$ which relates a group's label to the sequential component whose instances form that group. We assume that a sequential component is represented by its initial state.*

Definition 7. (Derivative set of a group) *Let $ds(P)$, $P \in \mathcal{C}_{\mathbb{M}}$ be the set of local derivatives of a sequential component P. We define $ds^*(H)$ as the derivative set of group $H \in \mathcal{G}(\mathbb{M})$, and is derived as $ds^*(H) = ds(sc(H))$.*

Using the above definitions, now we characterise the numerical vector that captures the state of a group H within a model $\mathbb{M}$.

Definition 8. (Numerical vector capturing state of a group) *Let H be a group in $\mathcal{G}(\mathbb{M})$ with the derivative set $ds^*(H)$. At any given time, the state of the instances in group H is captured by the vector:*

$$\boldsymbol{\xi}(H) = \langle\, \xi(H,P) \mid P \in ds^*(H) \,\rangle$$

where $\xi(H,P)$ is a variable capturing the number of instances in H that are currently in state P.

For a group H, $\boldsymbol{\xi}(H)$ expresses how the population of sequential processes within H are distributed across the local states $ds^*(H)$. The state of the model is captured by putting together the vectors constructed for groups $H \in \mathcal{G}(\mathbb{M})$:

Definition 9. (Numerical vector underlying a large-scale model) *The numerical vector used for the state representation of model $\mathbb{M}$ is*

$$\boldsymbol{\xi} = \langle\, \boldsymbol{\xi}(H) \mid H \in \mathcal{G}(\mathbb{M}) \,\rangle$$

where $\boldsymbol{\xi}(H)$ is the numerical vector defined as above.

2.4.3 Population-Based State Space

The GPEPA syntax is regarded as a convenient means of specifying the model's populations, and introduces no additional layer of expressiveness. Therefore, we can use the same semantics of Fig. 2.1 to generate the model's underlying state space. However, this state space would not be readily expressed in terms of numerical vectors. For this purpose, all states that have the same numerical vector are lumped into their corresponding numerical vector. The theoretical basis of the corresponding lumpability condition is laid out in [47, 59]. The result is a state space fully expressed in numerical vector form.

Example

Consider the client-server system shown in Eq.(2.5). The numerical vector for capturing the state of the system is $\langle S_i, S_l, C_t, C_r \rangle$ where S_i is the number of servers being idle, S_l is the number of servers logging, C_t is the number of clients thinking and C_r is the

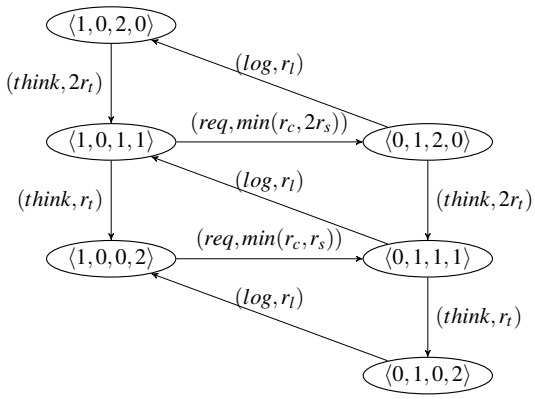


Figure 2.3: The state space of the client-server system with one server and two clients represented using numerical vector $\langle S_i, S_l, C_t, C_r \rangle$.

number of clients requesting. The model's original state space was shown in Fig 2.2. The state space built using the notion of numerical vectors is shown in Fig 2.3.

2.4.4 Count-Oriented Semantics

The *count-oriented semantics* is an extension of SOS and enables us to directly generate the state space in the numerical vector form. The semantics will be formally introduced in Sec. 4.3.2, but to help with our presentation let us informally describe it here and briefly explain what it achieves. First, we extend the notion of apparent rate functions to make them applicable to numerical vectors.

Definition 10. (Apparent rate of an action with respect to a group for two specific local states) *Let H be a group in $\mathcal{G}(\mathbb{M})$ with derivative set $ds^*(H)$, and P, P' be two local states within $ds^*(H)$. The total rate observed in group H for α activities that take instances of P into P' is denoted by $r_\alpha(H, P, P')$ and is derived as:*

$$r_\alpha(H, P, P') = \xi(H, P) \times r_\alpha(P, P') \tag{2.8}$$

where $r_\alpha(P, P')$ is defined using Def. 2. For any α, P and P' the function $r_\alpha(H, P, P')$ is a linear function with respect to the population count $\xi(H, P)$.

Next, we consider the capacity of a whole group in performing an action type α. Recall that in the most general case, there might be more than one local state in $ds^*(H)$ which enables α.

Definition 11. (Apparent rate of an action type offered by a group) *Let H be a group*

in $\mathcal{G}(\mathbb{M})$ with the derivative set $ds^*(H)$. The apparent rate of an action type α with respect to H is denoted by $r_\alpha(H)$ and is defined as:

$$r_\alpha(H) = \sum_{P \in ds^*(H)} \xi(H,P) \times r_\alpha(P)$$

where $r_\alpha(P)$ is derived using Def.1. For any action α and group H the function $r_\alpha(H)$ is a linear function with respect to the population counts $\xi(H,P)$ where each P is a local state that enables α.

The numerical state vectors can evolve due to the transitions made by the individual instances. Consider a group H with $P \in ds^*(H)$, $P = (\alpha, r_1).P' + (\alpha, r_2).P'' + \cdots$. A state change such as $P \xrightarrow{(\alpha, r_\alpha)} P'$ by *any* individual P causes the whole vector $\boldsymbol{\xi}(H)$ to change:

$$\boldsymbol{\xi}(H) = \langle \cdots, \xi(H,P), \cdots, \xi(H,P'), \cdots \rangle \xrightarrow{(\alpha,R)}$$
$$\boldsymbol{\xi}'(H) = \langle \cdots, \xi(H,P) - 1, \cdots, \xi(H,P') + 1, \cdots \rangle \quad (2.9)$$

where R is evaluated by Eq.(2.8). When instances in two distinct groups cooperate (say, P_1 instances in H_1 cooperate with P_2 instances in H_2 on a shared α activity becoming P_1' and P_2' respectively) the update to the numerical state vector will make changes analogous to those outlined above, i.e. the subvectors corresponding to the two groups will simultaneously be updated. Then, the rate of the shared activity will be:

$$R = \frac{r_\alpha(H_1, P_1, P_1')}{r_\alpha(H_1)} \frac{r_\alpha(H_2, P_2, P_2')}{r_\alpha(H_2)} \min(r_\alpha(H_1), r_\alpha(H_2)) \quad (2.10)$$

In general, the rate R is a function which includes rational divisions of linear functions and (possibly recursive) minimum expressions between linear functions. The construction of this expression follows a similar reasoning to the one shown on Page. 19.

Given a model's initial state in the numerical vector form, the count-oriented semantics takes into account the activities of the model's sequential processes and derives the set of changes or transitions (such as Eq.(2.9)) that the initial state may experience. This is used to infer the set of states, also expressed in the numerical form, that the system visits next. The rates associated with the transitions are obtained using the population-based apparent rate functions defined as above. By applying the semantic rules repeatedly, we derive the model's transition system that shows a state space in numerical vector form. The transition system is used to derive the model's underlying *population-based* CTMC.

In a model with state vector $\boldsymbol{\xi} = \langle \xi(H,P) \rangle$, $H \in \mathcal{G}(\mathbb{M})$, $P \in ds^*(H)$, each state variable $\xi(H,P)$ is regarded as a random variable. Furthermore, the state vector $\boldsymbol{\xi}$ is a vector of dependent random variables. To the model's underlying CTMC we assign a joint probability distribution $\mathbb{P}_t(\boldsymbol{\xi})$ for capturing the evolution of the state variables. The goal of the model analysis is to obtain different quantitative measures over this distribution, such as some marginal distributions of interest, or the moments such as expectation, variance, etc.

2.5 Split-Free Models

In the derivation of population-based apparent rate functions (Eq.(2.8) and Eq.(2.10)) we imposed no restriction on the structure of PEPA models. In particular, we accommodated cases where an action type shared within a cooperation could be *split into multiple* activities of that type in the context of the participating components. An example of this is the process $P[n] \underset{\alpha}{\bowtie} Q[m]$ where instances of process $P = (\alpha, r_1).P' + (\alpha, r_2).P'' + \cdots$ cooperate with instances of $Q = (\alpha, r_1').Q' + (\alpha, r_2').Q'' + \cdots$ on α actions. Assuming that P and Q have no further α activities, the cooperation on α is split into four combinations; each α activity by a P is connected to one of the α activities in Q and vice versa. The cooperation is divided into different activities, each with a *different impact* (update) on the numerical vector. A model in which there exist a splitting cooperation is called a *splitting* model [55].

The existence of a splitting cooperation has one important consequence. The apparent rate of a splitting action will include *divisions of linear functions* and hence, the function as a whole becomes non-linear. As illustrated in Eq.(2.10), the non-linearity is due to the probability expressions that assign weights to all possible transitions enabled by that action.

The presence of such non-linear terms introduces unnecessary and otherwise avoidable complications when developing rate function approximations in the context of analysis methods (e.g. different classes of moment closure techniques). For models which do not include splitting cooperation, the probability fractions reduce to one and the rate functions are piece-wise linear. Consequently, in their analysis it is possible to apply many accurate and useful approximations that are well studied and significantly improve the efficiency. However, for splitting models such techniques become inapplicable and if used, the results would be poor in accuracy. Since there are potentially

many models that capture the dynamics of a given system, we prefer our models not to be splitting.

The complications of the splitting cooperations are avoided by the design principle of *split-freeness* [55]. Let G represent a group or a composition of groups, and $\overrightarrow{\mathcal{A}^*}(G)$ be the set of action types enabled by sequential instances within G. We regard a model $\mathbb{M}$ to be split-free if $\forall\, \alpha \in \overrightarrow{\mathcal{A}^*}(\mathbb{M}) \;:\; sf(\mathbb{M}) = true$, where the boolean function $sf(\cdot)$ is defined as:

$$sf(M_1 \underset{L}{\bowtie} M_2) \;=\; \begin{cases} sf(M_1,\alpha) & \text{if } \alpha \in \overrightarrow{\mathcal{A}^*}(M_1) \;\wedge\; \alpha \notin \overrightarrow{\mathcal{A}^*}(M_2) \\ sf(M_2,\alpha) & \text{if } \alpha \notin \overrightarrow{\mathcal{A}^*}(M_1) \;\wedge\; \alpha \in \overrightarrow{\mathcal{A}^*}(M_2) \\ false & \text{if } \alpha \in \overrightarrow{\mathcal{A}^*}(M_1) \wedge \alpha \in \overrightarrow{\mathcal{A}^*}(M_2) \wedge \alpha \notin L \\ sf(M_1,\alpha) \wedge sf(M_2,\alpha) & \text{if } \alpha \in \overrightarrow{\mathcal{A}^*}(M_1) \wedge \alpha \in \overrightarrow{\mathcal{A}^*}(M_2) \wedge \alpha \in L \end{cases}$$

$$sf(H\{D\},\alpha) \;=\; true \qquad \begin{array}{l}\text{if there exists at most one local derivative } P \in ds^*(H) \\ \text{which enables at most one } \alpha \text{ activity } P \xrightarrow{(\alpha,\cdot)} P'\end{array}$$

Intuitively, the condition means that if an action type α is not shared, then it only appears within a single sequential process in the model. On the other hand, if it is shared, then it gives rise only to a single combination of activities.

The majority of the models we build in practice are split-free. Furthermore, non-conformance of a model can be flagged by an initial syntactic analysis and the model can be transformed into a split-free one through simple preprocessing such as action renaming. Outlining the full extent of such transformations is not within the scope of this thesis. However, since the split-freeness imposes no restriction on the expressiveness of PEPA and does not conceptually restrict the scope of systems that can be modelled, in this thesis we assume that our models are split-free. Based on this consideration, we introduce the following definitions related to population-based apparent functions.

Definition 12. (Local state of a sequential process enabling an action) *Let $\overrightarrow{\mathcal{A}^*}(\mathbb{M})$ be the set of actions defined in the sequential processes of model $\mathbb{M}$. For an action $\alpha \in \overrightarrow{\mathcal{A}^*}(\mathbb{M})$ and a sequential process $P \in \mathcal{C}_{\mathbb{M}}$ we define $en_\alpha(P)$ as the local derivative in $ds(P)$ that enables α.*

Definition 13. (Apparent rate of an action with respect to a group) *For a group $H \in \mathcal{G}(\mathbb{M})$ and an action $\alpha \in \overrightarrow{\mathcal{A}^*}(\mathbb{M})$, the apparent rate $r_\alpha(H)$ is expressed by a linear*

function as:

$$r_\alpha(H) = \xi(H,P') \times r_\alpha(P') \tag{2.11}$$

where $P' = en_\alpha(P)$ *and* $P = sc(H)$.

Definition 14. *(Local states of a group or group composition enabling an action) Let* H *be a group with* $P = sc(H)$. *For an action type* α, *we define* $en_\alpha(H)$ *as the local derivative in* $ds^*(H)$ *which enables* α. *Furthermore, let* G *be a group composition with* $\mathcal{G}(G)$ *as its set of group labels. The set of local derivatives of instances within* $\mathcal{G}(G)$ *that are involved in performing* α *is denoted by* $en_\alpha(G)$ *and is derived as:*

$$en_\alpha(G) = \{\ (H,P) \mid H \in \mathcal{G}(G)\ ,\ P = en_\alpha(\ sc(H)\ )\ \} \tag{2.12}$$

Definition 15. (Apparent rate of an action with respect to a group or group composition) *For a group or group composition* G, *the apparent rate of* α *with respect to* G *is expressed by a linear function as:*

$$r_\alpha(G) = \min_{(H,P)\in en_\alpha(G)} r_\alpha(H) = \min_{(H,P)\in en_\alpha(G)} \xi(H,P) \times r_\alpha(P) \tag{2.13}$$

where $\xi(H,P)$ *is the element of the state vector associated with the local derivative* (H,P).

We will use the above definitions in the next chapters.

Chapter 3

Faithfulness in the Analysis of LSRB Models

3.1 Introduction

We focus on LSRB models, one class of large-scale models in which the groups involved are divided into two categories: resources or resource users. We assume that the groups of users are significantly larger than those of the resources. The categorisation of groups is indeed conceptual and adaptive. Modern computer systems have complex architectures with components that represent various roles. Nonetheless, it is usually possible to form, for a defined set of analysis goals, a dependency hierarchy among the components which captures their reliance on each other. Within this hierarchy, the resources, on whose services others rely, reside towards the bottom, and resource users which rely on lower level components, reside towards the top. A system is then *mapped* to a LSRB model when a horizontal cut is specified on the hierarchy; the components above the line are regarded as users and those below are taking the role of resources. When the level of abstraction or the performance questions of interest change, a new cut and a new categorisation is specified.

In this chapter, we describe some of the key behavioural features that LSRB models exhibit. These features give a sharp contrast to these models against the class of large-scale models where all groups are uniformly large (for example, a peer-to-peer network where the nature of cooperation is more distributed and there is no notion of resource / resource users.). Then we illustrate some shortcomings of the existing anal-

ysis methods when applied to LSRB models. The context of our presentation will be an experiment performed on the model of a client-server system, where a small group of servers (resources) serve the large groups of clients (resource users). We show that for LSRB models, the methods that are typically prescribed for large-scale models potentially fail to simultaneously provide high degrees of efficiency and faithfulness.

We use the observations made during the experiment to propose a framework (a number of new analysis methods) particularly tailored for LSRB models. In this chapter, an overview of this framework will be provided. The last part of our chapter concerns the related work. Indeed, the problem of the analysis of large-scale models with highly stochastic behaviours is not new. We present alternative methods such as hybrid simulation, Langevin expansion, conditional linear noise approximation, etc. and compare them with our framework.

Our presentation follows this structure. In Sec. 3.2 we describe some notation used when building LSRB models. In Sec. 3.3, we describe the existing analysis methods of the PEPA framework. In Sec. 3.4, we report the results of an experiment which shows the pitfalls of utilising those methods for LSRB models[1]. We also provide an overview of our new analysis methods. In Sec. 3.6, we describe the related work and differentiate our contribution from the previous work.

3.2 Notation - LSRB Models

A LSRB model is specified by a pair $\langle \mathbb{M}, \Delta_{\mathcal{G}} \rangle$ where $\mathbb{M}$ is a grouped PEPA model and $\Delta_{\mathcal{G}} = \{ \mathcal{G}_s(\mathbb{M}) , \mathcal{G}_l(\mathbb{M}) \}$ is a partition over the set $\mathcal{G}(\mathbb{M})$ of the model's groups. The set $\mathcal{G}_s(\mathbb{M})$ consists of the labels of the small groups and $\mathcal{G}_l(\mathbb{M})$ consists of labels related to the large groups. Based on the conceptual categorisation of the entities, the small groups capture the dynamics of resources and the large ones capture the dynamics of the resource users. We assume that the modeller provides $\Delta_{\mathcal{G}}$ by considering the role of components and comparing the size of a system's groups against a defined threshold.

Example

Consider the following client-server model and assume that $n_s = 2$ and $n_c = 150$; the

[1] The content presented in Sec. 3.4 has been published in [79].

servers form a small group and the clients a large one.

$$C_{think} \stackrel{def}{=} (think, r_t).C_{req} \qquad C_{req} \stackrel{def}{=} (req, r_c).C_{think}$$

$$S_{idle} \stackrel{def}{=} (req, r_s).S_{log} \qquad S_{log} \stackrel{def}{=} (log, r_l).S_{idle}$$

$$CS \stackrel{def}{=} Servers\{\ S_{idle}[n_s]\ \} \underset{\{req\}}{\bowtie} Clients\{\ C_{think}[n_c]\ \} \tag{3.1}$$

The associated LSRB model is specified as $\langle CS, \Delta_{\mathcal{G}} \rangle$ where $\mathcal{G}_s(CS) = Servers$ and $\mathcal{G}_l(CS) = Clients$.

3.2.1 Partition on State Variables

Recall that $\boldsymbol{\xi}$ denotes the model's underlying numerical state vector. The partition $\Delta_{\mathcal{G}}$ is used to form a partition over variables of $\boldsymbol{\xi}$. Without loss of generality, $\boldsymbol{\xi}$ is reformulated as $\boldsymbol{\xi} = \langle \boldsymbol{\xi}^s, \boldsymbol{\xi}^l \rangle$, where $\boldsymbol{\xi}^s = \langle \boldsymbol{\xi}(H) \mid H \in \mathcal{G}_s(\mathbb{M}) \rangle$ consists of the state variables related to the small groups and $\boldsymbol{\xi}^l = \langle \boldsymbol{\xi}(H) \mid H \in \mathcal{G}_l(\mathbb{M}) \rangle$ consists of the state variables associated with large groups.

Example

The state vector capturing the state of the above client-server model is $\boldsymbol{\xi} = \langle S_i, S_l, C_t, C_r \rangle$, where S_i captures the number of servers being idle, S_l captures the number of servers logging, C_t is the number of clients thinking and C_r is the number of clients requesting. Given the partition $\Delta_{\mathcal{G}}$ as above, we have $\boldsymbol{\xi} = \langle \boldsymbol{\xi}^s, \boldsymbol{\xi}^l \rangle$, where $\boldsymbol{\xi}^s = \langle S_i, S_l \rangle$ and $\boldsymbol{\xi}^l = \langle C_t, C_r \rangle$.

3.2.2 Impact Vectors

On Page 27, we showed how the values of the state variables in $\boldsymbol{\xi}$ change as a result of individual or shared actions. The variables experience transitions such as Eq.(2.9), where those related to the local states enabling the action decrease by one and the variables related to the next local states are increased by one. We define the notion of *impact vectors* to capture the *net change* or influence caused by an action type on the elements of vector $\boldsymbol{\xi}$. Here, our presentation will be brief. For the formal construction of impact vectors, see [35, Chapter 3].

	Impact vector : $\mathcal{V}_\alpha$			
	$\mathcal{V}^s_\alpha$		$\mathcal{V}^l_\alpha$	
Action α	S_i	S_l	C_t	C_r
req	-1	$+1$	$+1$	-1
log	$+1$	-1	0	0
think	0	0	-1	$+1$

Table 3.1: Impact vectors of the actions defined in the client-server model.

Definition 16. (Impact vectors) *Let* $\vec{\mathcal{A}}^*(\mathbb{M})$ *denote the set of action types of model* $\mathbb{M}$. *For any* $\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})$, *the impact vector* $\mathcal{V}_\alpha = \langle \mathcal{V}_{\alpha,(H,P)} \mid H \in \mathcal{G}(\mathbb{M}), P \in ds^*(H) \rangle$ *represents the net change induced on* ξ *when an* α *activity is performed. The impact vector* $\mathcal{V}_\alpha$ *is derived as:*

$$\forall\ H \in \mathcal{G}(\mathbb{M})\ ,\ P \in ds^*(H)\ :$$

$$\mathcal{V}_{\alpha,(H,P)} = \begin{cases} -1 & \exists\, P, P' \in ds^*(H)\ :\ P \xrightarrow{(\alpha,\cdot)} P' \\ & \textit{if performing } \alpha \textit{ requires a component in local state } (H,P) \\ +1 & \exists\, P, P' \in ds^*(H)\ :\ P' \xrightarrow{(\alpha,\cdot)} P \\ & \textit{if performing } \alpha \textit{ produces a component in local state } (H,P) \\ 0 & \textit{otherwise} \end{cases}$$

Example

The set of impact vectors of the client-server model are shown in Table 3.1. For each of the actions *req*, *log*, *think* its impact vector captures how a transition of that type changes the state vector $\langle S_i, S_l, C_t, C_r \rangle$.

In order for an action α to be enabled, there must be at least one instance in each of the local states that enable α. Obviously, if the activity is individual, the requirement is simpler and we would be concerned with only one local state. We define the notion of *negative impact vector* to capture the requirement of an action being enabled in terms of the numerical values of the state variables.

Definition 17. (Negative impact vectors) *For any action type* α, *its negative impact*

	Impact vector : $\mathcal{V}_\alpha$			
	$\mathcal{V}^s_\alpha$		$\mathcal{V}^l_\alpha$	
Action α	S_i	S_l	C_t	C_r
req	1	0	0	1
log	0	1	0	0
think	0	0	1	0

Table 3.2: Negative impact vectors of the actions defined in the client-server model.

	Impact vector : $\mathcal{V}_\alpha$			
	$\mathcal{V}^s_\alpha$		$\mathcal{V}^l_\alpha$	
Action α	S_i	S_l	C_t	C_r
req	0	1	1	0
log	1	0	0	0
think	0	0	0	1

Table 3.3: Positive impact vectors of the actions defined in the client-server model.

vector is $\mathcal{V}^-_\alpha = \langle\ \mathcal{V}^-_{\alpha,(H,P)} \mid H \in \mathcal{G}(\mathbb{M})\ ,\ P \in ds^*(H)\ \rangle$ *where:*

$$\mathcal{V}^-_{\alpha,(H,P)} \quad = \quad \begin{cases} +1 & \textit{if performing } \alpha \textit{ requires a component in local state } (H,P) \\ 0 & \textit{otherwise} \end{cases}$$

For $\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})$, the vector $\mathcal{V}^-_\alpha$ is zero everywhere, except the elements $\xi(H,P)$ corresponding to local states that enable α.

Example

The negative impact vectors of the client-server model are shown in Table 3.2.

It is useful to define also the *positive* impact of an action. This captures the state variables that increase as a result of performing the action.

Definition 18. (Positive impact vectors) *For any activity* α*, its positive impact vector is* $\mathcal{V}^+_\alpha = \langle\ \mathcal{V}^+_{\alpha,(H,P)} \mid H \in \mathcal{G}(\mathbb{M})\ ,\ P \in ds^*(H)\ \rangle$ *where:*

$$\mathcal{V}_{\alpha,(H,P)} \quad = \quad \begin{cases} +1 & \textit{if performing } \alpha \textit{ generates a component in local state } (H,P) \\ 0 & \textit{otherwise} \end{cases}$$

For α, the vector $\mathcal{V}^+_\alpha$ is zero everywhere, except the elements $\xi(H,P)$ corresponding to local states that are one step derivatives of α activities within the sequential components. For any action type $\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})$, we have:

$$\mathcal{V}_\alpha = \mathcal{V}^+_\alpha - \mathcal{V}^-_\alpha \tag{3.2}$$

Example

The positive impact vectors of the client-server model are shown in Table 3.3.

3.2.3 Partition on an Impact Vector

Similarly to $\boldsymbol{\xi}$ being divided into two sub-vectors $\boldsymbol{\xi} = \langle \boldsymbol{\xi}^s, \boldsymbol{\xi}^l \rangle$, the partition $\Delta_{\mathcal{G}}$ is used to divide the impact vectors into two parts. For example, $\mathcal{V}_\alpha$ is formulated as $\mathcal{V}_\alpha = \langle \mathcal{V}_\alpha^s, \mathcal{V}_\alpha^l \rangle$, where $\mathcal{V}_\alpha^s$ is the impact of α on the vector $\boldsymbol{\xi}^s$, and $\mathcal{V}_\alpha^l$ is the impact on $\boldsymbol{\xi}^l$. A similar partitioning is possible for $\mathcal{V}_\alpha^-$ and $\mathcal{V}_\alpha^+$ resulting respectively in $\mathcal{V}_\alpha^{-,s}$, $\mathcal{V}_\alpha^{-,l}$, $\mathcal{V}_\alpha^{+,s}$ and $\mathcal{V}_\alpha^{+,l}$.

Example

The partitions related to the impact vectors of the client-server model are shown in Table 3.1, 3.2 and 3.3 over the columns. For each impact vector, we have specified how it affects $\boldsymbol{\xi}^s = \langle S_i, S_l \rangle$ and $\boldsymbol{\xi}^l = \langle C_t, C_r \rangle$ in two separate parts.

3.3 Analysis Methods for Large-Scale Models

In this section, we review the main analysis techniques available in the PEPA framework. For each technique we describe its assumptions, algorithm, output and computational cost. In the following, we assume that a model $\mathbb{M}$ has the state space denoted by D which is built based on the counting abstraction. Also, each numerical state $S_i \in D$ has the state vector $\boldsymbol{\xi}_i$; this is expressed as $S_i = \boldsymbol{\xi}_i$.

3.3.1 Exact Markovian Analysis

Exact Markovian analysis is the standard technique for deriving the evolution[2] of the model's underlying probability distribution $\mathbb{P}_t(S_i)$, $S_i \in D$. The distribution is the solution of the set of Chapman-Kolmogorov (C-K) equations associated with the CTMC whose state space is captured by D [50, Chapter 6]. The assumption of applying the method is that D is small enough such that it can be fully constructed. Then, the probability of being in each state $S_i \in D$ is governed by:

$$\frac{d\,\mathbb{P}_t(S_i)}{d\,t} = \sum_{\alpha : S_j \xrightarrow{(\alpha, r_\alpha(S_j))} S_i} r_\alpha(S_j) \cdot \mathbb{P}_t(S_j) \quad - \sum_{\alpha : S_i \xrightarrow{(\alpha, r_\alpha(S_i))} S_j} r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) \quad (3.3)$$

[2]Here, we are focusing on the transient evolution.

For any state S_i, its equation captures how the probability of being in the state evolves as a result of the probability fluxes in and out of S_i. Using an initial distribution $\mathbb{P}_{t_0}(S_i)$, the system of equations is solved and the evolution of probability distribution $\mathbb{P}_t(S_i)$ is derived for the time period of interest. A common solution method is uniformisation [102], which avoids using differential integration and offers a higher degree of numerical stability.

The method offers the highest degree of analytical faithfulness; at any time t, the probability of being in any state that the model may be in is derived. The probability distribution that is obtained captures the model's stochastic behaviour in full detail. However, for realistic large-scale systems, their underlying state spaces are usually massive, even after the counting abstraction, and this type of analysis is intractable or if applied, is computationally very expensive.

3.3.2 Stochastic Simulation

The method of stochastic simulation relies on the assumption that instead of the construction of the complete state space, the behaviour of the model can be analysed by observing a large number of traces or trajectories across its state space. Often, the algorithm used for trajectory generation is Gillepie's SSA [44] or its extensions [46, 84]. Given an initial state $S \in D$, the time Δt until the next transition is sampled by considering the rates of the actions enabled in S. Having decided on Δt and the action that occurs first, the clock is moved forward by Δt and the state is updated according to the impact of the fired action. By following the above steps repeatedly, a trajectory is generated. By producing a large number of trajectories, the approximate distribution $\mathbb{P}_t^{apx}(S)$, $S \in D$ is derived.

The computational cost of the analysis by stochastic simulation depends on the scale of the model. As the scale grows, the frequency of events occurring increases, and to accurately generate the trajectories we need to consider smaller time steps. This increases the cost of generating each trajectory. Also, to obtain statistically accurate results, we often need to run a very large number of simulations, even when the goal of analysis is focused on a single set of parameter values. These features mean that although the stochastic simulation makes the analysis tractable, but it is still considered to be a computationally expensive method.

3.3.3 Fluid Flow Approximation

Fluid flow approximation is based on one very important observation made when the scale of PEPA models increases [94]. To describe this, we introduce the notion of Lipschitz continuity [83], density dependent Markov chains (DDMC) [10], and the convergence property of DDMCs [66].

3.3.3.1 Lipschitz Continuity

Let E denote an open set in $\mathbb{R}^d$. A function $f : \mathbb{R}^d \to \mathbb{R}$ is Lipschitz continuous in E if there exists a real constant $K \geq 0$ such that:

$$\forall\ \boldsymbol{\xi}_i\ ,\ \boldsymbol{\xi}_j \in E\ \ ,\ \ \frac{||\ f(\boldsymbol{\xi}_i) - f(\boldsymbol{\xi}_j)\ ||}{||\ \boldsymbol{\xi}_i - \boldsymbol{\xi}_j\ ||} \leq K \tag{3.4}$$

Lipschitz continuity is a strong form of continuity and a conforming function features a number of interesting features, including:

1. It is globally differentiable across E.
2. The change of the function between any two points in E is bounded and thus, the function changes happen in a relatively slow and controlled way (as a counter-example, consider $f(x) = x^2$, which is *not* Lipschitz in $\mathbb{R}$).

3.3.3.2 Density Dependent Markov Chains

Let $\{X^{(N)}(t)\}$ be a family of population-based CTMCs defined over state space $D \subset \mathbb{Z}^d$ where d is the dimension of the state vector. Here, N denotes the *scaling factor*; $X^{(N)}$ has the same set of jumps (experiences the same impact vectors) as $X^{(1)}$, but its populations are N times larger. The family $\{X^{(N)}(t)\}$ is referred to as density dependent if there exists a function $\varphi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ that expresses the elements of the infinitesimal generator matrices $Q^{(N)}$ as:

$$\forall\ \boldsymbol{\xi}_i \in D\ \ :\ \ q^{(N)}_{\boldsymbol{\xi}_i, \boldsymbol{\xi}_i + \mathcal{V}} = \varphi(\boldsymbol{\xi}_i, \mathcal{V}) = N \times \varphi\left(\frac{\boldsymbol{\xi}_i}{N}, \mathcal{V}\right) \tag{3.5}$$

Obviously, the rate of a jump that a state enables depends on the values of its population counts (state variables). In DDMCs, however, such rates can also be formulated in terms of population *densities*. Assuming that N is the scaling factor (representing the

system size) the density vector associated with each state $\boldsymbol{\xi}_i$ is captured by $\frac{\boldsymbol{\xi}_i}{N}$. Studying the sequence of CTMCs $X^{(1)}, X^{(2)}, \cdots, X^{(N)}$ through their density vectors allow us to compare them and observe the effect of increasing scale in the sequence.

3.3.3.3 Convergence of DDMCs

Let $\{X^{(N)}(t)\}$ be a family of DDMCs in $\mathbb{Z}^d$ with the function φ defined as above. Suppose that φ is Lipschitz continuous in $\mathbb{R}^d$. Also, consider the vector field $F : \mathbb{R}^d \to \mathbb{R}^d$:

$$\forall\ x \in \mathbb{R}^d \ \ :\ \ F(x) = \sum_{\mathcal{V}} \mathcal{V} \times \varphi(x, \mathcal{V}) \tag{3.6}$$

with the associated ODE problem:

$$\frac{d\,x(t)}{d\,t} \ = \ F(x) \tag{3.7}$$

Then we observe that when $N \to \infty$, the sequence $X^{(1)}(t), X^{(2)}(t), \cdots, X^{(N)}$ converges to a deterministic limit in the sense that:

$$\left(\lim_{N\to\infty} \frac{X^{(N)}(t_0)}{N} = \delta\right) \implies \forall\, \varepsilon > 0 \ :\ \lim_{N\to\infty} \left(\sup_t || \frac{X^{(N)}(t)}{N} - x(t)|| > \varepsilon\right) = 0 \tag{3.8}$$

Intuitively, this means that when $N \to \infty$ and the initial densities $\frac{X^{(N)}(t_0)}{N}$ converge to δ, the behaviour of the discrete state process $\frac{X^{(N)}(t)}{N}$ is approximated by the evolution of a continuous process $x(t)$, which is the solution of Eq.(3.7) given the initial value $x(t_0) = \delta$.

NB. The original convergence theorem of Kurtz in [67] for DDMC is slightly more precise than the version above, in the sense that it considers the convergence only with respect to open set $E \subset \mathbb{R}^d \times \mathbb{R}^d$ where φ is strictly Lipschitz. In this thesis we are considering PEPA models which have bounded state spaces that can be enclosed in $\mathbb{R}^d$, and have piece-wise linear rate functions with simple behaviours. Thus, we can safely assume that $E = \mathbb{R}^d \times \mathbb{R}^d$, and relax the other necessary conditions imposed by the original theorem. A detailed discussion of the additional conditions is provided in [18].

3.3.3.4 Convergence of PEPA Models

Consider the sequence of models $\mathbb{M}^{(1)}, \mathbb{M}^{(2)}, \cdots, \mathbb{M}^{(N)}$. A model $\mathbb{M}^{(N)}$ has the same sequential processes and groups as $\mathbb{M}$, but its groups contain populations which are N

times larger. Tribastone et al. [94] established that the convergence property of DDMC can be used to form a convergence property for PEPA models of increasing scales, which then forms the basis of fluid flow approximation. Recall that the state vector $\boldsymbol{\xi}$ evolves in $\mathbb{Z}^d$ where d is the number of state variables. We define the vector field $F : \mathbb{R}^d \to \mathbb{R}^d$:

$$\forall\ x \in \mathbb{R}^d \ :\ F(x) = \sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \mathcal{V}_\alpha \cdot r_\alpha(x) \tag{3.9}$$

and the associated ODE problem $\frac{d\,x(t)}{d\,t} = F(x(t))$. Then we observe that the above sequence of models gives rise to a sequence of DDMCs $X^{(1)}, X^{(2)}, \cdots, X^{(N)}$, with the asymptotic probabilistic convergence property of Eq.(3.8). The initial value associated with δ is derived by dividing the initial state of the models $\mathbb{M}^{(N)}$ by N. For increasing N, $\frac{X^{(N)}(t)}{N}$ converges to a deterministic limit.

One important corollary is that the re-scaled solution $N \cdot x(t)$ captures the mean evolution of $X^N(t)$ (expectation of $\mathbb{E}_t[\boldsymbol{\xi}^{(N)}]$), and for large N, the stochastic process $X^{(N)}(t)$ can be written in terms of this mean and a normally distributed random variable $\mathcal{E}^N$ (which captures the dispersion around the mean) multiplied by $\sqrt{N}$ [89]:

$$X^N(t) \approx \overbrace{N \cdot x(t)}^{\mathbb{E}_t[\boldsymbol{\xi}^{(N)}]} + \sqrt{N} \cdot \mathcal{E}^N(t) \tag{3.10}$$

According to Eq.(3.10), the expectation $\mathbb{E}[X^{(N)}]$ grows by the factor N, and the noise decreases by the factor $\frac{\sqrt{N}}{N}$ and for $N \to \infty$, the noise is close to zero.

Eq.(3.10) express that as the scale enlarges, the models exhibit wider variance. However, at the same time the behaviour gets increasingly more robust in terms of being clustered around the mean and the impact of these variations becomes increasingly negligible when compared against the mean and the domain of the vector field. More formally, the *index of dispersion* (i.e. the ratio between the variance and the mean) gradually vanishes. Thus, for models where all populations are large, the stochastic behaviour can be faithfully captured by the deterministic evolution of the mean $\mathbb{E}[\boldsymbol{\xi}]$ whilst abstracting from variance [96, Chapter 6]. The deterministic evolution is found by solving the fluid flow approximation equations, a set of differential equations based on Eq.(3.7).

The fluid flow analysis offers the maximum efficiency. The deterministic behaviour is obtained by solving a system of ODEs whose size does not depend on the populations of the instances within the groups. Therefore, the method is particularly useful

for experimental analyses where one needs to sweep a large parameter space to find the optimal configurations.

3.3.4 Moment Closure Techniques

The moment closure technique acknowledges the limitation that the systems we build in practice never have infinitely large populations and aims at quantifying the variance. Assuming that a model's populations are all large and the probability distribution $\mathbb{P}(\boldsymbol{\xi})$, $\boldsymbol{\xi} \in D$, is *densely clustered* around the expectation $\mathbb{E}[\boldsymbol{\xi}]$, the moment closure technique allows us to derive higher-order moments, such as variance and skewness to capture noise and gain a more detailed representation of the shape of $\mathbb{P}(\boldsymbol{\xi})$. The moments are derived by solving a system of ODEs derived directly from the model. The ODEs constructed for the first-order moment $\mathbb{E}[\boldsymbol{\xi}]$ resemble Eq.(3.7): $\frac{d\,\mathbb{E}_t[\boldsymbol{\xi}]}{d\,t} = F(\mathbb{E}_t[\boldsymbol{\xi}])$. To account for the second-order moments and above, the system of ODEs is augmented with additional equations. The construction of these equations is complex and is not included here. The details can be found in [56]. The solution is the evolution of the *central* moments (with respect to the mean) of the distribution $\mathbb{P}_t(\boldsymbol{\xi})$.

The complexity of applying the method depends on the moment order of interest; the higher the order, the more equations are added and the higher is the analysis cost. In practice, assuming that the underlying distribution is uni-modal the analysis involves the derivation of moments of up to the fourth order as the moments beyond, which are costly to derive, add little meaningful information with respect to the shape of the distribution. In spite of the cost incurred by addition of the higher-order moments, the moment closure technique is still considered as an efficient method, since the generation of the complete state space is completely bypassed and the structure of the ODEs does not depend on the population levels. An example is presented in the next section.

3.4 Investigating Analytical Faithfulness

In this section, we report the result of an experiment where the analysis methods presented above were applied to two versions of a client-server model. Here, we illustrate the general steps and key findings. The full description can be found in our paper "Don't just go with the flow, cautionary tales of the fluid flow approximation" [79].

Parameter	r_s	r_l	r_c	r_t	n_s	n_c
Value	500	120	2	0.06	10	l0000

Table 3.4: Parameters used for the first version of the client-server model.

3.4.1 First Model - Client-Server Model Revisited

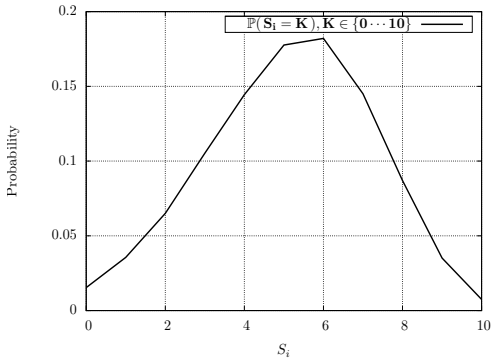
The first model we consider is the one shown on Page 33. We use the synthesized parameters shown in Table 3.4 to initialise the model. The system has 10000 clients that are served by 10 fast servers. The parameters are chosen in a way that the servers perform under heavy contention, and the length of the client waiting queue (clients in state C_r) is strongly sensitive to the number of available servers. The goal is to evaluate the number of clients waiting given the service profile of the servers.

3.4.2 First Model - Analysis

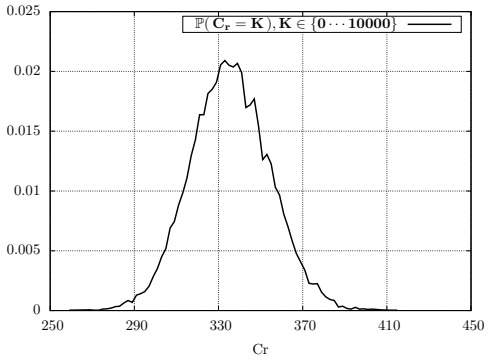
Exact Markovian Analysis. First, we tried to analyse the model using the exact Markovian analysis. The time span of interest is $0 < t < 100$ (hours) which covers the transient evolution until the system settles into the stationary phase. The system gives rise to a state space with 110011 states and 310010 transitions. The machine we used for our analysis was equipped with $4G$ Ram and 2.3Ghz processors. The tool we used was the PRISM's *hybrid* engine [68], which applies complex techniques for compact storage of the state space and uniformisation for transient analysis. The model, with its modest state space size, could unfortunately not be analysed within the maximum allocated time of 24 hours.

Stochastic Simulation. Second, we used stochastic simulation and the tool PEPA Eclipse Plug-in [93]. The result with respect to state variable S_i and C_r is shown in Fig 3.1A and 3.1B. In this analysis we ran 20000 simulation runs, which results in the confidence interval $\mathbb{E}[S_i] \pm 0.6$ and $\mathbb{E}[C_r] \pm 20$ (acceptable for this experiment) at the confidence level 95%. The analysis took nearly 4.5 hours. Here, note the distribution of C_r; it is densely clustered around the expectation $\mathbb{E}[C_r] = 319$.

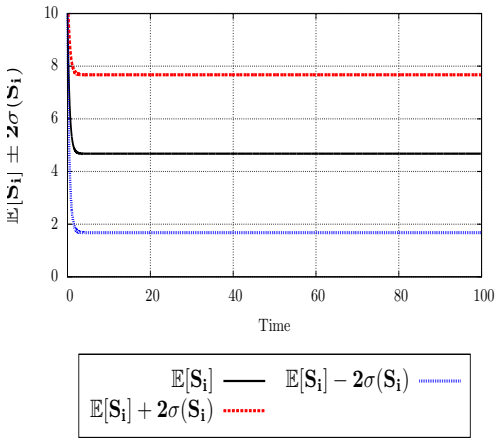
Fluid Flow Approximation and Moment Closure. Finally, we applied the fluid flow approximation and analysis of higher-order moments up to the second order. The set



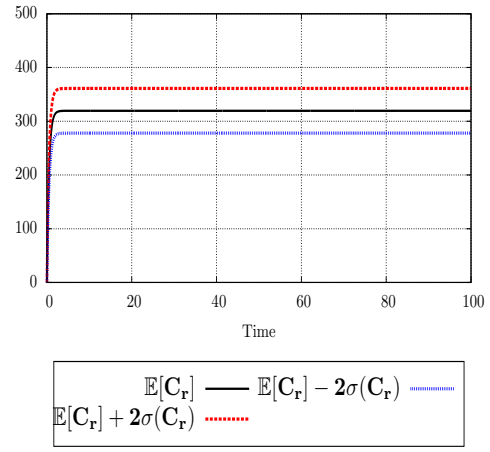
(A) Probability distribution of S_i at $t = 100$ derived using stochastic simulation.



(B) Probability distribution of C_r at $t = 100$ derived using stochastic simulation.



(C) expectation S_i .



(D) expectation C_r.

Figure 3.1: The results of the applying the fluid flow approximation and stochastic simulation on the first version of the model.

of equations constructed for the first-order mean approximation is:

$$\frac{d\,\mathbb{E}_t[C_t]}{dt} = -r_t \times \mathbb{E}_t[C_t] + \min(r_c \times \mathbb{E}_t[C_r], r_s \times \mathbb{E}[S_i])$$

$$\frac{d\,\mathbb{E}_t[C_r]}{dt} = -\min(r_c \times \mathbb{E}_t[C_r], r_s \times \mathbb{E}_t[S_i]) + r_t \times \mathbb{E}[C_t]$$

$$\frac{d\,\mathbb{E}_t[S_i]}{dt} = -\min(r_c \times \mathbb{E}_t[C_r], r_s \times \mathbb{E}[S_i]) + r_l \times \mathbb{E}[S_l]$$

$$\frac{d\,\mathbb{E}_t[S_l]}{dt} = +\min(r_c \times \mathbb{E}_t[C_r](t), r_s \times \mathbb{E}[S_i]) - r_l \times \mathbb{E}_t[S_l]$$

The system of equations was solved with the initial condition $\mathbb{E}[S_i] = 10$, $\mathbb{E}[S_l] = 0$, $\mathbb{E}[C_r] = 10000$ and $\mathbb{E}[C_t] = 0$. The solution with respect to $\mathbb{E}[S_i]$ and $\mathbb{E}[C_r]$ is shown in Fig. 3.1C and 3.1D. Note that $\mathbb{E}[C_r]$ accurately captures the value within the domain of C_r where the distribution is densely clustered. For second-order moments, we derived

the variances and then standard deviations of the state variables using the Grouped PEPA Analyser [23] which provides the automatic derivation of the related equations. The code which runs GPA is available at the our online supplements [77].

For this model, we have the knowledge that the underlying distribution is uni-modal and close to being normally distributed. By exploiting this, we are able to build 95% confidence bounds of the underlying distribution by combining the first and second-order moments. For any normally distributed random variable X with the mean $\mu(X)$ and the standard deviation $\sigma(X)$, we have:

$$\mathbb{P}(\ \mu(X) - 2\sigma(X) \leq X \leq \mu(X) + 2\sigma(X)\ ) \approx 95\% \tag{3.11}$$

This means that 95% of the observations lie within the interval $\mu(X) - 2\sigma(X) \leq X \leq \mu(X) + 2\sigma(X)$. Applying this rule, we combine $\mathbb{E}[S_i]$ and $\sigma_t(S_i)$, and $\mathbb{E}[C_r]$ and $\sigma_t(C_r)$ to build confidence bounds associated with the marginal distributions $\mathbb{P}(S_i)$ and $\mathbb{P}(C_r)$. These are also shown in Fig. 3.1C and 3.1D.

The analysis of moments took approximately 30 seconds. Compare this with 24 hours of waiting for the exact Markovian analysis and the 4.5 hours of the stochastic simulation. Also, compare the confidence bounds at $t = 100$ with the distributions of S_i and C_r. We observe that the combination of the expectations and the bounds allows us to build a faithful representation of the model's underlying distribution at any given time (this has been checked for all $0 \leq t \leq 100$ but we only show the result for $t = 100$). These show the importance of approaches based on the deterministic approximation. When applicable, the randomness can be faithfully studied very efficiently.

3.4.3 Second Model - Unfaithfulness Emerges

Next, we consider the second version of the client-server model:

$$C_{think} \stackrel{def}{=} (think, r_t).C_{req} \qquad C_{req} \stackrel{def}{=} (req, \top).C_{think}$$

$$S_{idle} \stackrel{def}{=} (req, r_s).S_{log} + (brk, r_b).S_{broken}$$

$$S_{log} \stackrel{def}{=} (log, r_l).S_{idle} \qquad S_{broken} \stackrel{def}{=} (fix, r_f).S_{idle}$$

$$CS \stackrel{def}{=} Servers\ \{\ S_{idle}[n_s]\ \} \underset{\{req\}}{\bowtie} Clients\ \{\ C_{think}[n_c]\ \}$$

Here, each server is susceptible to failure. The parameters used are the same as the previous one except those shown in Table 3.5. We have chosen $r_f < r_b$; the average

Parameter	r_s	r_t	r_b	r_f
Value	200	0.05	0.0006	0.0004

Table 3.5: Parameters used in the extension of the client-server model where the servers can break down.

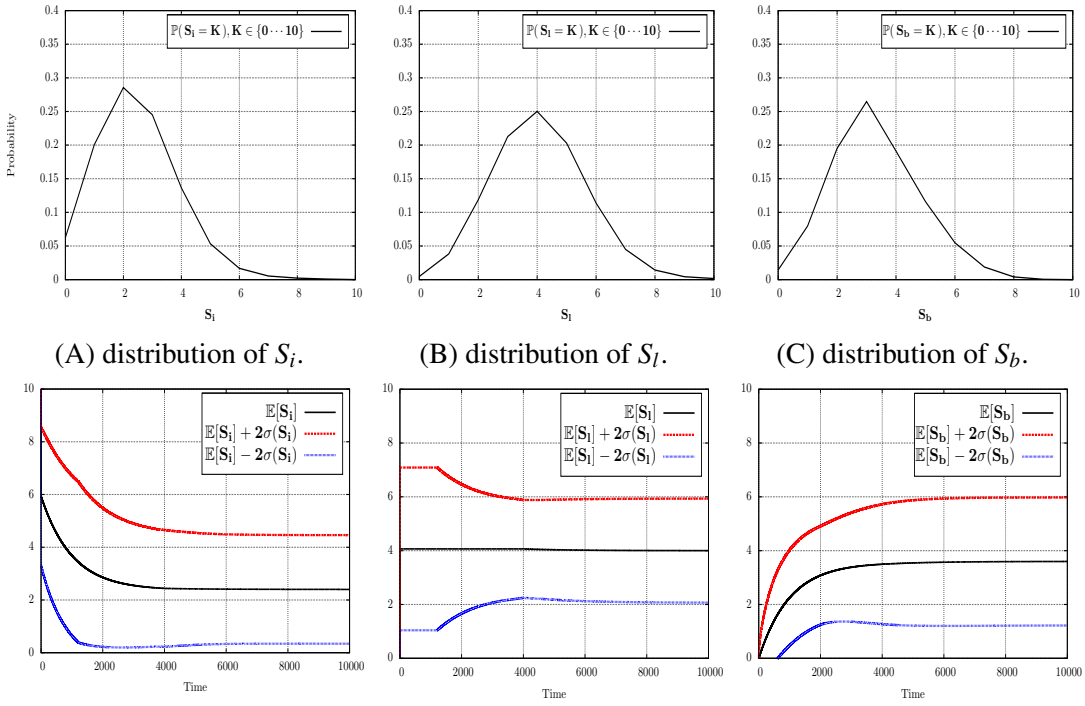
time a fix takes is longer than the time it takes to break down. This helps us study situations where the failure of one server increases the contention on the others and it becomes more likely to see more servers breaking down before the broken ones get fixed.

The state vector used for constructing the state space is $\boldsymbol{\xi} = \langle S_i, S_l, S_b, C_t, C_r \rangle$; the variable S_b is added which captures the number of servers in state S_{broken}. The model is assumed to start from state $\langle 10, 0, 0, 10000, 0 \rangle$. The time of interest is $0 \leq t \leq 16000$; for this version, the mixing time of the underlying CTMC is much longer.

3.4.4 Second Model - Analysis

The aforementioned analysis techniques were also applied to this version of the model. Again, the state space was too large to be analysed by the exact Markovian analysis. The next method was stochastic simulation, which gave us the evolution of the approximate distribution over $\langle S_i, S_l, S_b, C_t, C_r \rangle$. We used this to extract the marginal distributions of S_i, S_l, S_b and C_r. For a time point of interest $t = 16000$, these are respectively shown in Fig. 3.2A, 3.2B, 3.2C and 3.3A. Finally, the fluid flow analysis and moment closure technique were used to derive the expectation and standard deviations of the state variables. These are shown in Fig. 3.2D, 3.2E, 3.2F and 3.3B.

Fig. 3.3A illustrates that the behaviour of this second version is very distinct, both qualitatively and quantitatively, compared to the first version. A server can experience failure; when this happens (by undertaking the *brk* action), the throughput offered by the collection of servers for the *req* action decreases and it causes a rapid spike in C_r. On the other hand, when a broken server becomes active again (by undertaking fix action), the service rate increases and as a result C_r considerably decreases. These fluctuations are illustrated in Fig. 3.4, which shows a trajectory with respect to state variables S_b and C_r. The system has distinct *modes of operation* associated with the different number of servers that are currently broken. In each mode, the clients observe a different service level and variable C_r clusters around different values.



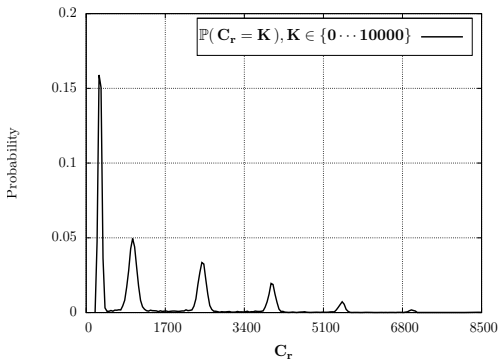
(A) distribution of S_i. (B) distribution of S_l. (C) distribution of S_b.

(D) expected evolution of S_i and the confidence bounds (E) expected evolution of S_l and the confidence bounds (F) expected evolution of S_b and the confidence bounds

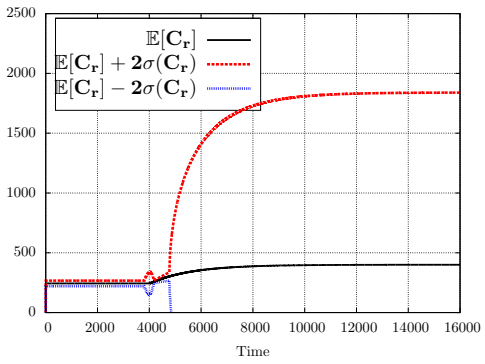
Figure 3.2: The results of applying the fluid flow approximation and stochastic simulation on the model with break-downs.

3.4.5 Experiment Outcome

The aim of the experiment was to investigate the efficiency and faithfulness of the PEPA framework's analysis techniques for LSRB models. Given the computational cost of the exact Markovian analysis and stochastic simulation, the natural choice is to use fluid flow approximation and the moment closure technique. For the first model, the behaviour was closely clustered around a single average (the underlying distribution was uni-modal) and the moments could faithfully capture the model's stochastic behaviour. However, for the second model this is not the case. Compare Fig. 3.3A and 3.3B. Observe that the combination of the expectation and standard deviation does not capture the mutli-modality of C_r's distribution. Whilst the clients appear in a large population and exhibit a robust behaviour in each of the operational modes, once the servers change the current mode, the clients exhibit a distinctively different behaviour. In this case, the central moments are *too crude* to faithfully capture important qualitative and quantitative aspect of the model's dynamics. This is the direct result of

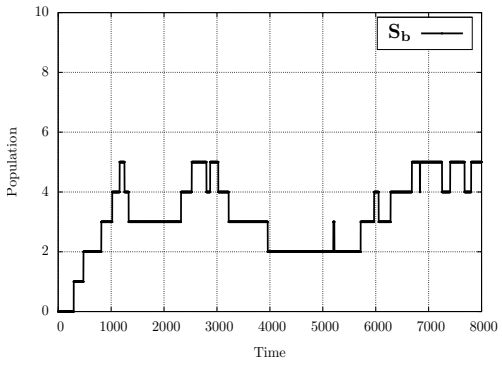


(A) distribution of C_r at $t = 16000$ derived using stochastic simulation.

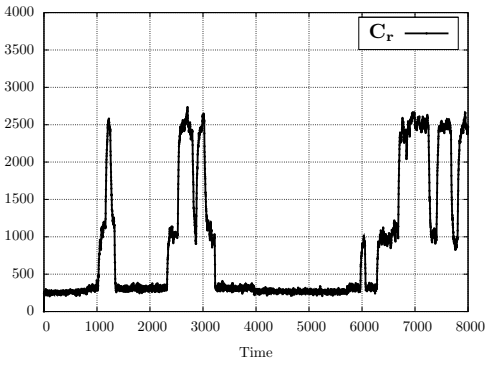


(B) expectation of C_r and the confidence bounds.

Figure 3.3: The result of the analysis with respect to the clients.



(A) A trajectory of the client-server system projected on variable S_b.



(B) A trajectory of the client-server system projected on variable C_r.

Figure 3.4: A trajectory of the system projected on the state variables S_b and C_r.

abstracting away from the highly noisy behaviour of servers, and ignoring the associated impact on the emergent behaviour by only considering their average evolution.

3.5 The Issue of Faithfulness for LSRB Models

The outcomes of the experiment can be carefully generalised to all LSRB models. Recall that the analysis of a LSRB model is essentially boiled down to understanding the behaviour of the sub-vectors $\boldsymbol{\xi}^s$ and $\boldsymbol{\xi}^l$, which respectively capture the state of the model's small and large groups (respectively, resources and resource users). Given that the scale of $\boldsymbol{\xi}^l$ causes the explosion of the state space, and that exact Markovian analysis and stochastic simulation are computationally expensive, the use of the moment based techniques (fluid flow approximation and moment closure technique) is the

natural choice. The pitfall however, is that in doing so, *a key modelling requirement is violated*; these methods are guaranteed to be faithful and informative for models where all groups are large and the emergent behaviour is densely clustered around a single average (see Eq.(3.10)). Indeed, the majority of evaluation studies where the fluid flow approach has been particularly useful involved models with uniformly large populations [22, 8, 2, 52]. For LSRB models though, this assumption is not readily respected and the underlying distribution can exhibit arbitrary properties, such as multi-modality. Applying these methods means that the noisy behaviour of $\boldsymbol{\xi}^s$ and its knock-on effects on the emergent evolution is completely ignored. For these models, there is no justification for moment based techniques and their application can potentially be insufficient and even misleading in understanding the true behaviour of the real system.

Thus, the faithful analysis of LSRB models requires the PEPA framework to be extended by methods which, *whilst being efficient*, *incorporate the highly random behaviour of* $\boldsymbol{\xi}^s$ *into the analysis*.

> The analysis methods should provide the machinery to enable us to capture the potential *noise induced departures* from the fully deterministic behaviour whilst keeping up with the efficiency standards.

With this in mind, we propose the analysis framework of *Conditional Moments For LSRB PEPA Models*, which proposes a number of new methods. The outline of this framework is shown in Fig. 3.5. The input, shown in grey, is a LSRB model $\mathbb{M}$ and the partition $\Delta_{\mathcal{G}}$ on set $\mathcal{G}(\mathbb{M})$. The first step, shown in green, is an aggregation method, which is applied to the model and produces a sub-model which captures the dynamics of small groups only. This sub-model is used to obtained detailed information about the stochastic evolution of resources. One key information is the presence of dynamics at slow / fast time scales. We show that for detecting this feature one can run a quick reachability analysis on the state space of $\boldsymbol{\xi}^s$ and then extend the conclusion to the original model and the whole emergent behaviour. The second step, shown in red, is the computation of conditional expectations. Using the descriptions of the original and aggregated models, a system of differential algebraic equations (DAEs) is constructed which captures the evolution of conditional expectations $\mathbb{E}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\xi}^s\right]$, the average behaviour of the large groups given the different configurations that the small groups may experience. The third step, shown in blue, is the analysis of higher-order

conditional moments. Here, the DAEs are augmented with additional equations which capture further moments of the conditional distributions $\mathbb{P}(\boldsymbol{\xi}^l \mid \boldsymbol{\xi}^s)$. The core idea of the framework is simple: instead of calculating one set of moments which cover all of the system's significant and distinct modes of operation, we calculate one set of moments for each single mode separately. The conditional moments enable us to build rich representation of the model's underlying probability distribution and capture phenomenon such as multi-modality.

Our framework promotes transparency. The analysis methods added, as we will see, are based on heavy algebraic derivations. Nevertheless, we have integrated them with PEPA's syntax. Thus, the modeller works at the highly convenient level of modelling language and the results are calculated and presented seamlessly in terms of the model's entities. The framework also supports adaptivity; when there is more computational power available for analysis, the modeller supplies a new partition $\Delta_{\mathcal{G}}$ with more groups identified as small, and the amount of discreteness in the analysis is increased. The details of each of the framework's steps are explained in the next chapters.

3.6 Related Work

In the performance evaluation literature, there is ample evidence suggesting that the highly random behaviour of many systems cannot be faithfully captured by the central moments of their stochastic processes [49, 36, 98, 20, 86]. This issue has absorbed much attention in the last few years and the development of efficient and faithful analysis methods is currently subject to active research. In this section, we present some of the techniques that are related to our methods. For each method, we explain its input, an overview of mathematical steps, its output and its computational cost.

3.6.1 Hybrid Semantics and Hybrid Stochastic Simulation

Hybrid stochastic simulation (HSS) [84] is an extension of the conventional SSA algorithm which aims to reduce the cost of trajectory generation. HSS relies on the observation that transitions enabled by a system's large populations occur at high frequencies and the state variables related to those populations evolve nearly continuously. On the other hand, the transitions affecting small populations happen at lower frequencies and have somewhat discrete impacts. The transitions are thus categorised as either having a

The pair $\langle \mathbb{M}, \Delta_{\mathcal{G}} \rangle$ is used to specify the model's resources and resource users, and is used to build $\boldsymbol{\xi}^s$ and $\boldsymbol{\xi}^l$.

$$C_{think} \stackrel{def}{=} (think, r_t).C_{req}$$
$$C_{req} \stackrel{def}{=} (req, r_c).C_{think}$$

Resource users / clients

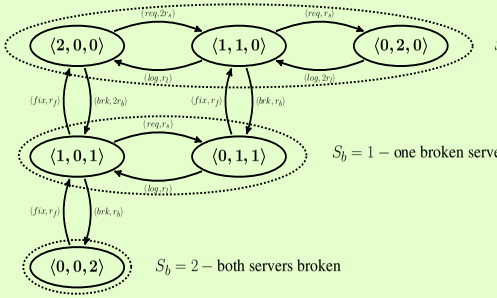
$$S_{idle} \stackrel{def}{=} (req, r_s).S_{log} + (brk, r_b).S_{broken}$$
$$S_{log} \stackrel{def}{=} (log, r_l).S_{idle}$$
$$S_{broken} \stackrel{def}{=} (fix, r_f).S_{idle}$$

Resources / Servers

$$CS \stackrel{def}{=} Servers\{ S_{idle}[n_s] \} \underset{\{req\}}{\bowtie} Clients\{ C_{think}[n_c] \}$$

Cooperation

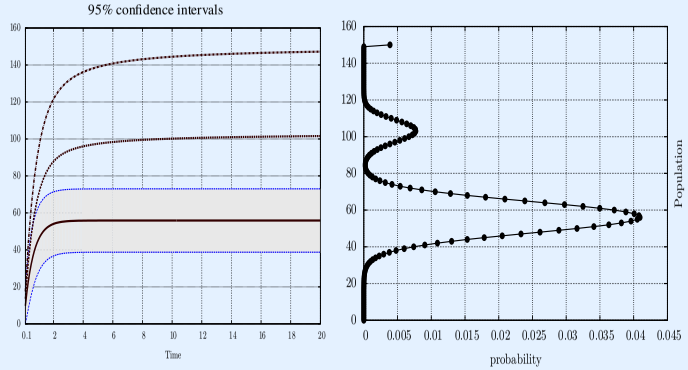
The stochastic behaviour of the system's resources is extracted directly from the model using an aggregation technique.



The aggregated state space is used to detect slow / fast dynamics of the emergent behaviour.

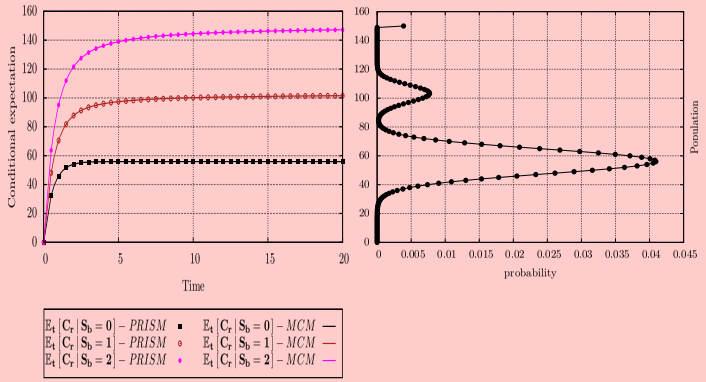
Analysis of Conditional Moments

The analysis of the conditional variances shows the distance between the actual stochastic behaviour of the resource users and their conditionally average behaviour.



Using the conditional variances, for each of the system's mode of operation, not only we have the average behaviour, but also how much the system deviates from the average.

With the analysis of conditional expectations, we obtain the expected behaviour of resource users across the configurations of the resources and also, across their distinct modes of operation.



The conditional expectations are derived by solving a system of differential algebraic equations which can be constructed directly from the model.

Figure 3.5: Overview of the analysis techniques in our framework.

continuous or discrete effect. In contrast with the conventional SSA where the impact of every individual transition is recorded discretely, in HSS the impact of the continuous transitions is captured by numerical integration and only the small populations' transitions are derived by the next jump detection of SSA. The speed up is achieved by capturing the impact of a sequence of fast jumps related to the large populations at once within the individual steps of the numerical integrator.

Previous work has explored the possibility of applying HSS to models specified in high level formalisms [42, 21, 69, 3]. One can systematically apply HSS to PEPA models through the PEPA hybrid semantics [19]. The input is a model and a partition on its actions, which divides them between the continuous / discrete categories. The hybrid semantics then generates a *transition-driven stochastic hybrid automaton* [21] (an extension of hybrid automaton [57]) $\mathcal{T} = (C_o, \mathcal{X}, \mathcal{TC}, \mathcal{TS}, init)$ where C_o is the set of control modes (states affected only by the discrete jumps), $\mathcal{X}$ is the set of state variables changing continuously, $\mathcal{TC}$ is the set of continuous transitions or flows, $\mathcal{TS}$ is the set of stochastic transitions and *init* is the initial state. The hybrid trajectories are generated as follows. Assuming the current mode C_m, the time Δt until the next discrete jump is sampled given C_m and the value of $\mathcal{X}$. The evolution of $\mathcal{X}$ for $[t, t+\Delta t]$ is derived using numerical integration. The clock proceeds Δt units and the above steps are repeated again. Using HSS, the exact behaviour of the model is approximated by obtaining a large number of hybrid trajectories.

Previous experiments show that HSS offers a high degree of faithfulness. However, this comes at the price of deriving a large number of trajectories. Although HSS generates trajectories much more efficiently than the conventional SSA, the fact that many trajectories are needed makes the whole approach computationally expensive.

HSS was one of the early developments which aimed to incorporate noise and randomness into the analysis of large-scale systems. The method showed that including an interplay between discrete jumps and continuous flows leads to a feasible and accurate (quantitatively and qualitatively) approach. This view gave rise to interesting methods which emerged later that employed solely numerical analysis and calculus of stochastic processes, and avoided running repeated simulations. Two such methods will be described in the following.

3.6.2 Using Stochastic Differential Equations

The use of *stochastic differential equations* (SDE) is traced back to the field of statistical physics and the derivation of approximate solutions to the so-called *chemical master equation* (CME) [45], which is the analogous version of the C-K equation in that field. SDEs appear in different forms. Here, we focus on how they are used for the evaluation of systems that give rise to density dependent Markov chains. Kurtz's derivations in [67] show that the behaviour of CTMC $X^N(t)$ with set of impact vectors $l \in \mathcal{L}$ can be approximated by the continuous-valued random variable $Y^N(t)$ that is the solution of:

$$dY^N(t) = \underbrace{F(Y^N(t)) \cdot dt}_{\text{drift}} + \underbrace{\sum_{l \in \mathcal{L}} \overbrace{\sqrt{r_l(Y^N(t))}}^{\text{co-efficient of noise}} \cdot dW_l(t)}_{\text{noise}} \tag{3.12}$$

Here F is a vector field defined similarly to Eq.(3.6), r_l is the rate of jump l happening in Y^N, and $\{W_l(t)\}_l$ are independent Brownian motions. Eq.(3.12) is also referred to as the Langevin equation. The approximation assumes that the stochastic behaviour is formulated in terms of a deterministic term (*drift*) and a noise term that captures the process's fluctuations. The solution of the SDE of Eq.(3.12) is a CTMC.

It is obvious that the expression of a stochastic behaviour by a *single* deterministic drift and the fluctuations around it cannot account for important phenomena such as multi-modality (recall our argument on Page 47 for unfaithfulness of fluid flow approximation). To account for randomness, the SDE approach has been recently extended. In the next two sections, we describe two relevant extensions, namely Langevin approximation with *discrete switches* [7] and *conditional linear noise approximation* [92].

3.6.2.1 Langevin Approximation with Discrete Switches

The method of SDE with switches [7] is similar to hybrid stochastic simulation. The system's transitions are partitioned into discrete and continuous, with some of the variables evolving according to the discrete jumps and the others continuously. The difference here is that for the continuous parts we use SDEs as opposed to ODEs.

The input of the method is a CTMC X and a partition $\mathcal{L} = \{\mathcal{L}_D, \mathcal{L}_C\}$ on set $\mathcal{L}$ of the CTMC's jumps. The behaviour of X is to be approximated by the evolution of a hybrid automaton where Y_D is the set of control modes and Y_C is continuous stochastic

processes. For each mode, the evolution of Y_C is derived by integrating an SDE of the form of Eq.(3.12) related to that mode. The approximate solution is obtained by running multiple hybrid simulations. In [7], Angius et al. provide hybrid semantics for Stochastic Petri nets, which applies the above steps automatically on models written in this modelling formalism, thus improving the method's applicability.

The method offers a high degree of faithfulness and has been shown to be more efficient than stochastic simulation. However, the disadvantage is that it still relies on producing numerous trajectories; for systems where the switching process has a large state space, the method is still computationally demanding, especially if the switching occurs at multiple time scales and there are rare but not absent events.

3.6.2.2 Conditional Linear Noise Approximation

Linear noise approximation (LNA) [76] is a technique used in systems biology to find an approximate solution to the CME by introducing a *very* simplifying assumption that its underlying distribution is normal. Borrowing some biology terminology, assuming that a biological system has volume Ω, the transient evolution of its stochastic process X that counts the population of interacting species is expressed as:

$$X \approx \Omega \cdot \phi + \sqrt{\Omega} \cdot \varepsilon \tag{3.13}$$

where ϕ is the solution of an ODE approximation (drift) and ε is a fluctuation process formed based on particular properties of the system's kinetics [100, Chp. 8]. Note the similarity of Eq.(3.13) to Eq.(3.12). Our focus is on a very recent extension of LNA presented in [92], where the goal is to capture multi-modality in gene regulatory networks by the method of *conditional LNA*. The extension is based on the assumption that multi-modal distributions are mixtures of normal ones and can be characterised by calculating the means and widths associated with each mode.

The analysis requires the state vector to be partitioned as $\langle D, C \rangle$ where D and C respectively capture the counts of species with low and high abundance (respectively, promoters and proteins). The variability, i.e. multiple peaks, is caused by the dynamics of the low population entities. Thus the evolution of D is studied by jumps concerning low copy entities only. On the other hand, the marginal distributions over the abundant entities are studied by the mean and variance of the conditional distributions $\mathbb{P}(C \mid D)$, derived by solving SDEs of the form in Eq.(3.13). The approximate complete probability distribution is derived by the weighted sum of the conditional distributions based

on the marginal distribution over D.

The method of conditional LNA is more efficient than stochastic simulation. Its core idea is very similar to our framework of conditional moments, namely, that the large groups are studied in terms of some conditional distributions. The main shortcoming of the conditional LNA is its excessive reliance on the requirement that the underlying distribution is a mixture of normal modes. As stated, the method was developed very recently and its validity was supported by one example of a gene regulatory network which was previously shown to give rise to mixture of normal distributions. Although the method is acceptable for such cases, it is expected to lead to unacceptable inaccuracy once the modes of the multi-modal distributions have any shape other than normal, such as being slightly skewed to one side. The benefit of our framework is that without imposing any restriction on the shape of the distributions, we are able to calculate their conditional moments up to any order, which then allows us to capture a wider class of distributions.

3.6.3 Conditional Moments of Chemical Master Equation

The last piece of related work we describe is the method of conditional moments (MCM) [53] designed for reaction networks that follow the Michaelis-Menten kinetics and have the presence of some species with low copy numbers. This method was the basis of our development (and inspired the above conditional LNA as well), and is traced back to the work of Engblom in [38].

The assumption of this version of MCM is similar to the previous method; the variability in the emergent behaviour is caused by the species with low copy numbers. The state vector of the stochastic process is reformulated into $\langle D, C \rangle$ where D and C follow the description given above. Similarly to the previous case, the evolution of D is derived by considering transitions which affect D only. However, the evolution of C is studied by the expectation and higher-order moments of the conditional distributions $\mathbb{P}_t(C \mid D)$. The higher the order, the better becomes the representation of the variability of C.

One appealing feature of MCM is that it imposes no particular restriction on the shape of the underlying distribution. The most important drawback is that when constructing the moment equations the evolution of moments of any given order depends on moments which have higher orders (in conditional LNA this is avoided by exploit-

ing the assumption on the shape of the distribution). The dependency gives rise to an unbounded hierarchy of equations, a well-known *curse* which moment closure practices suffer from. To close the system, the hierarchy is cut at some order and the moments above are considered to contribute zero. The quality of approximation depends on the depth of the equations [70].

The question of which layer to cut the hierarchy at is an important one and was considered recently by Schnoerr et al. in [85]. The authors show that the validity of the moment closure techniques highly depends on the parameters of the system: rates defined for the reactions and the population assigned to the species. For a set of parameters, if the equation hierarchy is cut too early, the solution becomes erroneous or might even include imaginary numbers for species populations, which obviously supports no physical interpretation. In other words, for any cutting order the method is valid only with respect to a *critical* subset of the parameter space. The experiments in [85] show that obtaining physically credible and accurate solutions for some realistic systems requires going up to the fourth or fifth order, which drastically increases the cost of analysis.

The negative feature above highlights an important advantage of our framework of conditional moments presented for LSRB PEPA models. In our derivations in Chap. 6 and 8, we show that for PEPA models, the moments of each order $\boldsymbol{I}$ depend on moments with orders less than or equal to $\boldsymbol{I}$. Thus, the system of equations becomes easily closed and no hierarchy cut is needed. This feature is due to the transition rates in PEPA being expressed by piece-wise *linear* functions which then enable us to perform accurate approximations that eventually avoid the unbounded growth in dependency. Our observation is in compliance with the findings of Engblom, suggesting that moment closure techniques give rise to more accurate results for reaction networks with linear rate functions.

A final point which differentiates our work is the fact that we have lifted MCM to the level of PEPA models. The derivation of conditional moments involves tedious algebra, and we believe its use is only viable if it is seamlessly integrated into the high-level modelling languages. To analyse LSRB models, the only additional information needed is the partition $\Delta_{\mathcal{G}}$ on group labels; the next steps are automatically done by software. To our knowledge, we are currently the first group who offers the MCM on an easily usable high-level platform, specifically, that of the PEPA framework.

Chapter 4

An Aggregation Method for Large-scale Resource-bound Models

4.1 Introduction

In this chapter, we propose an aggregation algorithm for LSRB models. For a given model, the method quickly checks if it satisfies a *syntactic* condition. If so, an aggregated CTMC is generated directly from the model, bypassing the construction of the complete state space. The aggregated CTMC captures the evolution of the model's small groups and can be used for efficient derivation of an approximate marginal probability distribution over them. We demonstrate the steps of the algorithm and as an example, describe how it is applied in the context of a client-server system.

Our presentation follows this structure. In Section 4.2 we formally introduce the aggregation condition. In Sections 4.3 and 4.4 we show the aggregation steps and the way the marginal probability distributions is derived. In Section 4.5, we describe the usefulness of the method in the context of an example.

The content of this chapter has been published in [78].

4.2 Aggregation Condition

In order to describe the syntactic aggregation condition, first we present the definitions of *cooperation hierarchy* and *synchronisation inteface*.

4.2.1 Some Definitions

4.2.1.1 Group Hierarchy

In the system equation of a model, the operator $\underset{L}{\bowtie}$ is used to compose the model's groups and form cooperations. The instances of two groups which are composed are restricted to synchronise on actions specified in the associated cooperation set. These groups, however, can be subjected to further compositions and synchronisations with other groups within the model. Thus, a system equation introduces a *hierarchy of cooperation* among groups and assigns to each group, a set of actions that the instances within that group must synchronise on. To clarify, consider the process $(G_1\{\cdot\} \underset{L'}{\bowtie} G_2\{\cdot\}) \underset{L}{\bowtie} (G_3\{\cdot\} \underset{L''}{\bowtie} G_4\{\cdot\})$. Here G_1 and G_2 cooperate on set L'; similarly, G_3 and G_4 cooperate on actions in set L''. Additionally, these component groups are also combined in cooperation on set L. The cooperation on L' restricts the instances in groups G_1 and G_2 and does not affect G_3 or G_4; but all instances are restricted by L.

The notion of cooperation hierarchy in a model $\mathbb{M}$ can be formally captured by a partial order relation $<^*_{\mathbb{M}}$. For group compositions M_i and M_j, we have $M_i <^*_{\mathbb{M}} M_j$, if and only if M_i is composed when constructing M_j. If $M_i <^*_{\mathbb{M}} M_j$, all cooperation sets applied to M_j are also enforced on M_i. The following rules construct $<^*_{\mathbb{M}}$ using $\mathbb{M}$'s system equation.

1. $H\{\cdot\} <^*_{\mathbb{M}} H\{\cdot\}$
2. $X <^*_{\mathbb{M}} A\ , \quad \text{if} \quad A \stackrel{def}{=} X$
3. $H\{\cdot\} <^*_{\mathbb{M}} M_i\{\cdot\} \underset{L}{\bowtie} M_j\{\cdot\}\ , \quad \text{if} \quad H <^*_{\mathbb{M}} M_i \vee H <^*_{\mathbb{M}} M_j$

4.2.1.2 Synchronisation Interface

For a group $H \in \mathcal{G}(\mathbb{M})$, we define $I(\mathbb{M},H)$ as the set of actions on which instances in H are required to synchronise. $I(\mathbb{M},H)$ is defined recursively from the system equation, using the subsidiary function $\mathcal{J}$; $I(\mathbb{M},H) = \mathcal{J}(\mathbb{M},H,\emptyset)$.

$$\mathcal{J}(M,H,K) \;=\; \begin{cases} K & \text{if } M \equiv H\{\cdot\} \\ \mathcal{J}(M_1,H,K\cup L) \cup \mathcal{J}(M_2,H,K\cup L) & \text{if } \; M \equiv M_1 \underset{L}{\bowtie} M_2 \\ \mathcal{J}(A,H,K), & \text{if } \; M \stackrel{def}{=} A \end{cases}$$

We also define $sync(\mathbb{M},H,\alpha)$, the set of groups in $\mathbb{M}$ whose instances synchronise on α activities with instances in H. Using $<^*_{\mathbb{M}}$, $sync(\mathbb{M},H,\alpha)$ can be derived as:

$$sync(M,H,\alpha) = \begin{cases} \emptyset \quad \text{if } M \equiv H\{\cdot\} \\ sync(M_1,H,\alpha) \cup \{H_i \in \mathcal{G}(\mathbb{M}) \,|\, H_i\{\cdot\} <^*_{\mathbb{M}} M_2\} \\ \quad \text{if } M \equiv M_1 \underset{L}{\bowtie} M_2, \alpha \in L, H <^*_{\mathbb{M}} M_1 \\ sync(M_2,H,\alpha) \cup \{H_i \in \mathcal{G}(\mathbb{M}) \,|\, H_i\{\cdot\} <^*_{\mathbb{M}} M_1\} \\ \quad \text{if } M \equiv M_1 \underset{L}{\bowtie} M_2, \alpha \in L, H <^*_{M} M_2 \\ sync(M_1,H,\alpha) \\ \quad \text{if } M \equiv M_1 \underset{L}{\bowtie} M_2, \alpha \notin L, H <^*_{\mathbb{M}} M_1 \\ sync(M_2,H,\alpha) \\ \quad \text{if } M \equiv M_1 \underset{L}{\bowtie} M_2, \alpha \notin L, H <^*_{\mathbb{M}} M_2 \end{cases}$$

4.2.2 Syntactic Condition

As a LSRB model is assumed to be a large-scale model, its complete CTMC is very large and its construction and analysis are computationally expensive. We show that if the model satisfies the following condition, then we can perform an aggregation which results in an aggregated CTMC for $\mathbb{M}$.

Condition 1. (Syntactic aggregation condition) *Let $\vec{\mathcal{A}}(C_u)$ represent the set of actions that a sequential process enables. A model $\mathbb{M}$ can be aggregated with respect to its small groups $\mathcal{G}_s(\mathbb{M})$ if for any shared activity α, synchronised on by one or more large groups and one or more small group, the rate of the shared activity is completely decided by the small groups:*

$$\forall\, H^l \in \mathcal{G}_l(\mathbb{M}) \;\; \forall\, \alpha \in I(\mathbb{M},H)$$
$$[\, (\, sync(M,H^l,\alpha) \,\cap\, \mathcal{G}_s(\mathbb{M})\,) \neq \emptyset \implies$$
$$\forall\, C_u \;\in\; ds^*(H^l) \;\; s.t. \;\; \alpha \in \vec{\mathcal{A}}(C_u)\;,\; \exists\, \omega \in \mathbb{N}\;,\;\; r_\alpha(C_u) = \omega\top\,]$$

The condition expresses that in any synchronisation on a shared action α, if both small and large groups are involved, then all instances in the involved large groups need to undertake α passively. As an example, consider a variant of the client-server model with the client components modified as:

$$C_{think} \stackrel{def}{=} (think, r_t).C_{req} \qquad C_{req} \stackrel{def}{=} (req, \top).C_{think}$$

We assume *Clients* constitutes a large group, and *Servers*, a small one. They synchronise on the *req* activity which the clients undertake passively. Thus the model now satisfies Condition 1. We will see that this implies that in any state of the system, the count-oriented apparent rate of *req* depends only on the configuration of the servers given that clients have a large population.

4.2.3 The CTMC Structure

The CTMC of a LSRB model that satisfies Condition 1 exhibits important lumpability [63] properties which can be exploited in order to build an aggregated CTMC. We introduce these properties and illustrate them in the CTMC of the client-server model initialised with two servers and two clients.

In order to show the structural properties, first we need to describe how the partition $\Delta_{\mathcal{G}}$ on the group labels is used to partition the set of action types into three categories. This will be described in the next sub-section, and then we continue the discussion.

4.2.3.1 Partitioning Actions

In Sec. 3.2, we showed that using $\Delta_{\mathcal{G}}$, we can partition the state vector as $\boldsymbol{\xi} = \langle \boldsymbol{\xi}^s, \boldsymbol{\xi}^l \rangle$, and similarly, the impact vectors as $\mathcal{V}_\alpha = \langle \mathcal{V}_\alpha^s, \mathcal{V}_\alpha^l \rangle$, $\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})$. Using $\Delta_{\mathcal{G}}$, we can also partition the set of actions where each action is categorised based on whether it changes the state of only one or more small groups, the state of only one or more large groups, or simultaneously, the state of both small and large groups.

For $H \in \mathcal{G}(\mathbb{M})$, let $\vec{\mathcal{A}}^*(H)$ represent the set of all the actions enabled by instances within H. Then, $\vec{\mathcal{A}}_i^*(H) = \vec{\mathcal{A}}^*(H) - I(\mathbb{M}, H)$ represents the set of actions not offered in H's interface and related to *individual* activities undertaken by instances in H.

First, assume that $H \in \mathcal{G}_l(\mathbb{M})$. We define $\vec{\mathcal{A}}_l^*(H)$ to be the set of actions that instances in H perform individually or in cooperation with other large groups.

$$\vec{\mathcal{A}}_l^*(H) = \{ \quad \alpha \mid \alpha \in \vec{\mathcal{A}}_i^*(H) \quad \vee \quad (sync(M,H,\alpha) \quad \cap \quad \mathcal{G}_s(\mathbb{M})) = \emptyset \quad \}$$

Based on $\vec{\mathcal{A}}_l^*(H)$, we can derive $\vec{\mathcal{A}}_l^*(\mathbb{M}) = \bigcup_{H \in \mathcal{G}_l(\mathbb{M})} \vec{\mathcal{A}}_l^*(H)$ as the set of actions related to the dynamics of the large groups only.

Second, assume that $H \in \mathcal{G}_s(\mathbb{M})$ and define $\vec{\mathcal{A}}^*_s(H)$ analogously for small groups:

$$\vec{\mathcal{A}}^*_s(H) = \{ \quad \alpha \mid \alpha \in \vec{\mathcal{A}}^*_i(H) \quad \vee \quad (sync(M,H,\alpha) \quad \cap \quad \mathcal{G}_l(\mathbb{M})) = \emptyset \quad \}$$

As above, we define $\vec{\mathcal{A}}^*_s(\mathbb{M}) = \bigcup_{H \in \mathcal{G}_s(\mathbb{M})} \vec{\mathcal{A}}^*_s(H)$ to be the actions in which the large groups do not participate.

Finally, for $H \in \mathcal{G}_s(\mathbb{M})$ we define $\vec{\mathcal{A}}^*_{sl}(H)$ to represent the set of actions shared between instances of H and instances of one or more large groups.

$$\vec{\mathcal{A}}^*_{sl}(H) = \{ \alpha \mid \alpha \in I(\mathbb{M},H) \wedge (sync(M,H,\alpha) \cap \mathcal{G}_l(\mathbb{M})) \neq \emptyset \}$$

Similarly, for model $\mathbb{M}$, $\vec{\mathcal{A}}^*_{sl}(\mathbb{M}) = \bigcup_{H \in \mathcal{G}_s(\mathbb{M})} \vec{\mathcal{A}}^*_{sl}(H)$ is defined to be the set of actions in which both small and large groups participate.

Using the above categorisation, the partition $\vec{\mathcal{A}}^*(\mathbb{M}) = \{\vec{\mathcal{A}}^*_s(\mathbb{M}), \vec{\mathcal{A}}^*_{sl}(\mathbb{M}), \vec{\mathcal{A}}^*_l(\mathbb{M})\}$ is constructed over the set of model's actions $\vec{\mathcal{A}}^*(\mathbb{M})$. It is easy to see that for any $\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})$, if $\alpha \in \vec{\mathcal{A}}^*_s(\mathbb{M})$, then $\mathcal{V}^l_\alpha = 0$; if $\alpha \in \vec{\mathcal{A}}^*_l(\mathbb{M})$, then $\mathcal{V}^s_\alpha = 0$; and if $\alpha \in \vec{\mathcal{A}}^*_{sl}(\mathbb{M})$, then $\mathcal{V}^s_\alpha \neq 0$ and $\mathcal{V}^l_\alpha \neq 0$. For the client-server system, $\vec{\mathcal{A}}^*_l(CS) = \{think\}$, $\vec{\mathcal{A}}^*_{sl}(CS) = \{req\}$ and $\vec{\mathcal{A}}^*_s(CS) = \{log\ ,\ brk\ ,\ fix\}$.

4.2.3.2 Identifiable Sub-chains

Consider a model $\mathbb{M}$ and assume that it respects Condition 1. In Fig. 4.1, we illustrate the impact of a transition of type $\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})$ on a symbolic state $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}^l_i \rangle$. Here, $\boldsymbol{\gamma}$ represents the configuration of the small groups; i.e. $\boldsymbol{\xi}^s_i = \boldsymbol{\gamma}$. Depending on α, one of the following is the case:

1. $\alpha \in \vec{\mathcal{A}}^*_s(\mathbb{M})$. In this case $\mathcal{V}^s_\alpha \neq 0$, $\mathcal{V}^l_\alpha = 0$. The transition only affects the configuration of the small groups and its rate is determined by $\boldsymbol{\xi}^s$: $r_\alpha(S_i) = r_\alpha(\boldsymbol{\gamma})$. For the target state $S_k = \langle \boldsymbol{\xi}^s_k, \boldsymbol{\xi}^l_k \rangle$ we have: $\boldsymbol{\xi}^s_k = \boldsymbol{\gamma} + \mathcal{V}^s_\alpha$ and $\boldsymbol{\xi}^l_k = \boldsymbol{\xi}^l_i$. This transition exists if $\boldsymbol{\gamma} > \mathcal{V}^{s,-}_\alpha$; the small groups satisfy the requirements of performing α.

2. $\alpha \in \vec{\mathcal{A}}^*_{sl}(\mathbb{M})$: In this case $\mathcal{V}^s_\alpha \neq 0$, $\mathcal{V}^l_\alpha \neq 0$. The transition changes the configurations of both small and large groups. In this case, $\boldsymbol{\xi}^s_k = \boldsymbol{\gamma} + \mathcal{V}^s_\alpha$ and $\boldsymbol{\xi}^l_k = \boldsymbol{\xi}^l_i + \mathcal{V}^l_\alpha$. This transition exists if both $\boldsymbol{\xi}^s_i$ and $\boldsymbol{\xi}^l_i$ enable it, that is, when $S_i \in D$, $\boldsymbol{\xi}^s_i \geq \mathcal{V}^{s,-}_\alpha$ and $\boldsymbol{\xi}^l_i \geq \mathcal{V}^{l,-}_\alpha$. Since the large groups are passive with respect to $\vec{\mathcal{A}}^*_{sl}(\mathbb{M})$ actions, $r_\alpha(S_i) = r_\alpha(\boldsymbol{\gamma})$.

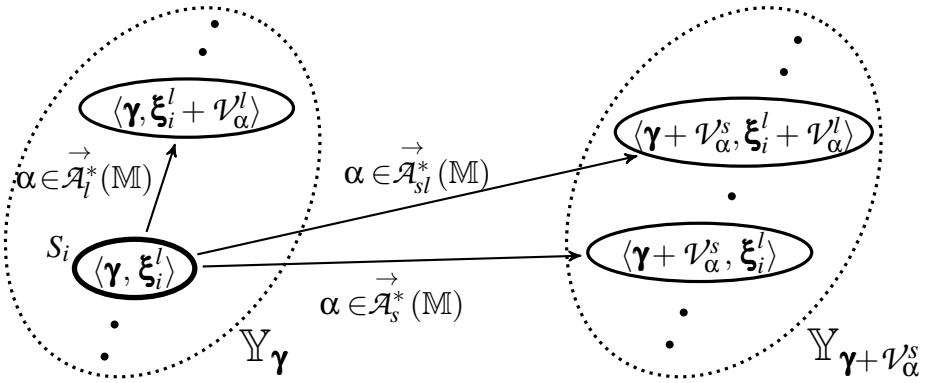


Figure 4.1: The impact of outward transitions with different types on the state vector.

	$S_i \xrightarrow{(\alpha,\cdot)} S_k$, $S_i = \langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l \rangle$, $S_k = \langle \boldsymbol{\xi}_k^s, \boldsymbol{\xi}_k^l \rangle$		
	$\alpha \in \overrightarrow{\mathcal{A}_s^*}(\mathbb{M})$	$\alpha \in \overrightarrow{\mathcal{A}_{sl}^*}(\mathbb{M})$	$\alpha \in \overrightarrow{\mathcal{A}_l^*}(\mathbb{M})$
necessary condition	$\boldsymbol{\xi}_i^s \geq \mathcal{V}_\alpha^{s,-}$	$\boldsymbol{\xi}_i^s \geq \mathcal{V}_\alpha^{s,-}$ $\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}$	$\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}$
impact	$\boldsymbol{\xi}_k^s = \boldsymbol{\xi}_i^s + \mathcal{V}_\alpha^s$ $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l$	$\boldsymbol{\xi}_k^s = \boldsymbol{\xi}_i^s + \mathcal{V}_\alpha^s$ $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l + \mathcal{V}_\alpha^l$	$\boldsymbol{\xi}_k^s = \boldsymbol{\xi}_i^s$ $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l + \mathcal{V}_\alpha^l$
rate	$r_\alpha(\boldsymbol{\xi}_i^s)$	$r_\alpha(\boldsymbol{\xi}_i^s)$	$r_\alpha(\boldsymbol{\xi}_i^l)$

Table 4.1: The properties of the outward transitions from a state S_i

3. $\alpha \in \overrightarrow{\mathcal{A}_l^*}(\mathbb{M})$: In this case $\mathcal{V}_\alpha^s = 0$ and $\mathcal{V}_\alpha^l \neq 0$. $\boldsymbol{\xi}_k^s = \boldsymbol{\xi}_i^s = \boldsymbol{\gamma}$, $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l + \mathcal{V}_\alpha^l$. The rate of the transition is determined by $\boldsymbol{\xi}_i^l$. This transition exists if $S_i \in D$ and $\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}$.

The properties of these transitions are summarised in Table.4.1.

Let us focus on transitions of type $\overrightarrow{\mathcal{A}_l^*}(\mathbb{M})$. These transitions only change the state of $\boldsymbol{\xi}^l$ and leave $\boldsymbol{\xi}^s$ unchanged. Thus, D can be divided into a number of *sub-chains* where the states within each sub-chain are connected by $\overrightarrow{\mathcal{A}_l^*}(\mathbb{M})$ transitions and for which $\boldsymbol{\xi}^s$ remains the same. From a state $S \in D$ a sub-chain $\mathbb{Y}_i$ can be derived using the rules:

1. $S \in \mathbb{Y}_i \wedge S \xrightarrow{(\alpha,\cdot)} S' \wedge \alpha \in \overrightarrow{\mathcal{A}_l^*}(\mathbb{M}) \implies S' \in \mathbb{Y}_i$
2. $S \in \mathbb{Y}_i \wedge S'' \xrightarrow{(\alpha,\cdot)} S \wedge \alpha \in \overrightarrow{\mathcal{A}_l^*}(\mathbb{M}) \implies S'' \in \mathbb{Y}_i$

The sub-chain $\mathbb{Y}_i$ associated with a state S consists of S itself, the states from which S can be reached by $\vec{\mathcal{A}}_l^*(\mathbb{M})$ transitions, and the states reachable from S by such transitions. The states within each sub-chain have the same configuration for small groups. Thus, each sub-chain is identified by the configuration it captures for $\boldsymbol{\xi}^s$; the sub-chain associated with states $S = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}^l \rangle \in D$ is denoted by $\mathbb{Y}_{\boldsymbol{\gamma}}$. For model $\mathbb{M}$ its set of sub-chains is denoted by $\mathbb{Y}_{\mathbb{M}}$.

As an example, in Fig. 4.2A, we show the CTMC of the client-server model with two clients and servers and the sub-chains identified. In each sub-chain, the clients change their state via *think* activities without affecting the servers. For a model with a larger client population, the same partition, but with longer sub-chains would be observable.

The next structural property we present concerns the rate regularity that we observe with respect to the cross-sub-chain transitions.

4.2.3.3 Unlikely Boundary States and Rate Regularities

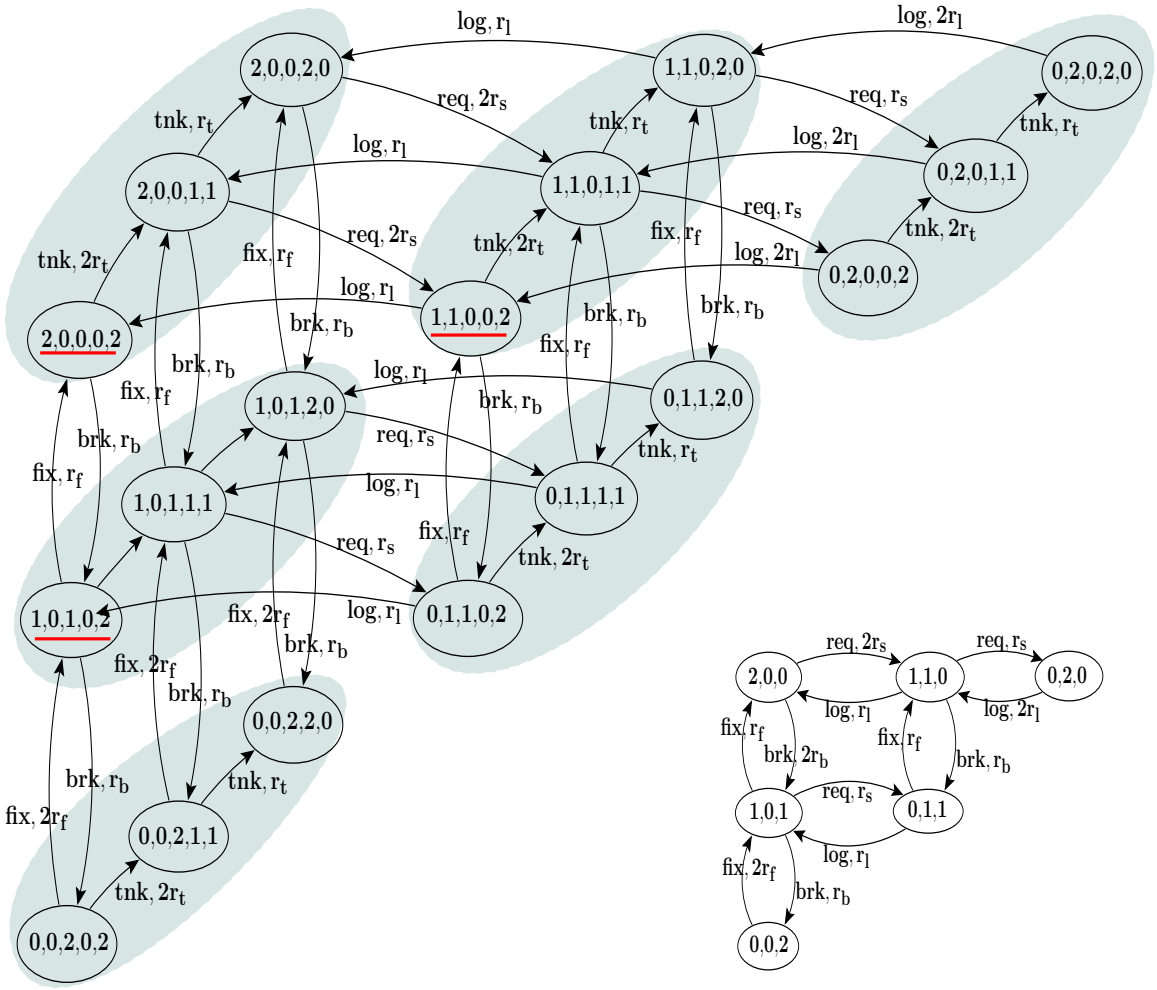
Let us consider a sub-chain $\mathbb{Y}_i$ in $\mathbb{Y}_{\mathbb{M}}$ and an action $\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})$. The action represents cooperation between resources (small groups) and resource users (large groups). In $\mathbb{Y}_i$, we can identify a subset of states where α activities are *not* enabled because there are no instances in one or more relevant large groups to participate. Extending this to all actions in $\vec{\mathcal{A}}_{sl}^*(\mathbb{M})$, in each sub-chain $\mathbb{Y}_i$ we can find a subset of states where one or more actions $\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})$ remain disabled specifically due to the lack of cooperation from the instances in the large groups. We refer to this subset as the *boundary states* of $\mathbb{Y}_i$ and define them formally below.

Definition 19. (Boundary states of an action) *For any action $\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})$, the set of boundary states in a sub-chain $\mathbb{Y}_i \in \mathbb{Y}_{\mathbb{M}}$ is denoted by $bl(\mathbb{Y}_i, \alpha)$ and is derived as:*

$$bl(\mathbb{Y}, \alpha) = \{\, S_i \mid S_i = \langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y} \;,\; \boldsymbol{\xi}_i^s \geq \mathcal{V}_\alpha^{s,-} \;,\; \boldsymbol{\xi}_i^l \ngeq \mathcal{V}_\alpha^{l,-} \,\} \tag{4.1}$$

Definition 20. (Boundary states of a sub-chain) *The set of boundary states of a sub-chain $\mathbb{Y} \in \mathbb{Y}_{\mathbb{M}}$ is defined as $bl(\mathbb{Y}) = \bigcup_{\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} bl(\mathbb{Y}, \alpha)$.*

The boundary states show the configurations of the system where the small groups (resources) are ready to cooperate (provide the service), but the required instances in the large groups (resources users) are not asking for the service. For instance, in the



(A) Complete CTMC. (B) Aggregated CTMC.

Figure 4.2: Complete and aggregated CTMC for the client-server system with two servers and two clients (shorthand $tnk = think$ is used). Each state of the complete CTMC is a vector $\langle S_i, S_l, S_b, C_r, C_t \rangle$, where S_i counts the number of idle servers, S_l the number of logging servers, S_b the number of broken servers, C_r the number of requesting clients and C_t the number of thinking clients. Each state of the aggregated CTMC follows this state descriptor: $\langle S_i, S_l, S_b \rangle$.

client-server example the boundary states are those where action *req* is enabled by at least one server, but there are no clients to make a request. These states are highlighted in Fig. 4.2A in red.

In *large-scale* and *resource-bound* systems, the users appear in large populations and resources are unlikely to remain idle. Thus, the probability of being in the boundary states can be assumed to be negligible:

$$\forall\, \mathbb{Y} \in \mathbb{Y}_{\mathbb{M}}\ ,\ \forall\, S \in bl(\mathbb{Y})\ :\ \mathbb{P}_t(S) \approx 0 \tag{4.2}$$

4.2.3.4 Rate Regularities

If we ignore boundary states as above, then we can observe an important structural property with respect to the rate of cross-sub-chain transitions.

Proposition 4.2.1. (Rate regularity for non-boundary states). *The non-boundary states of each sub-chain exhibit the same behaviour with respect to the actions in* $\overrightarrow{\mathcal{A}}^*_s(\mathbb{M})$ *and* $\overrightarrow{\mathcal{A}}^*_{sl}(\mathbb{M})$*:*

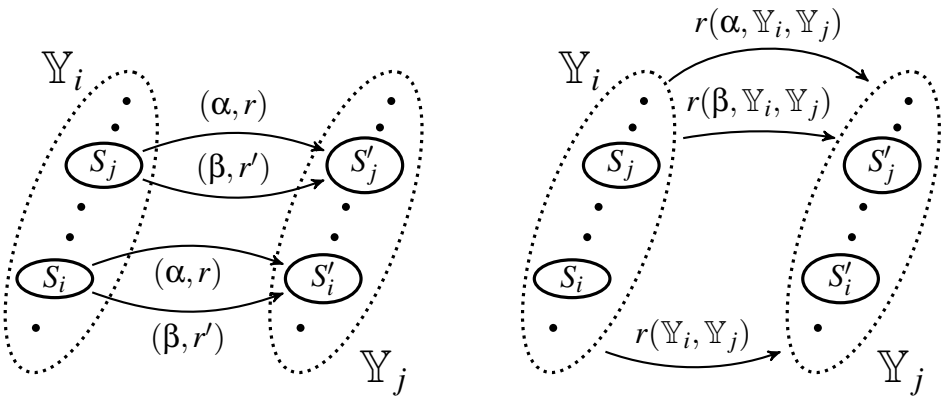
$$\forall\, \alpha \in (\ \overrightarrow{\mathcal{A}}^*_s(\mathbb{M}) \cup\ \overrightarrow{\mathcal{A}}^*_{sl}(\mathbb{M})\ )\ ,\ \forall\, S_i\ \in\ \mathbb{Y}_i\ ,\ \forall\, S_j\ \in\ \mathbb{Y}_j$$
$$\Bigg(S_i \xrightarrow{(\alpha,R)} S_j\ \Longrightarrow\ \forall\, S'_i\ \in\ (\mathbb{Y}_i - bl(Y_i,\alpha))\ ,$$
$$\Big[\, r_\alpha(S'_i > 0)\ \Longrightarrow\ \exists\, S'_j\ \in\ \mathbb{Y}_j\ \ S'_i \xrightarrow{(\alpha,R)} S'_j \,\Big] \Bigg)$$

Intuitively, the proposition expresses that in any $\mathbb{Y}_i$, all non-boundary states enable the same set of actions (see Fig. 4.3A). Also, the states offer the same rate with respect to each action. Thus, for any $\alpha \in \overrightarrow{\mathcal{A}}^*_s(\mathbb{M}) \cup \overrightarrow{\mathcal{A}}^*_{sl}(\mathbb{M})$ we define $r(\alpha, \mathbb{Y}_i, \mathbb{Y}_j)$, the rate of any α transition from any $S_i \in \mathbb{Y}_i$ to another state $S_j \in \mathbb{Y}_j$, $\mathbb{Y}_i \neq \mathbb{Y}_j$.

$$r(\alpha, \mathbb{Y}_i, \mathbb{Y}_j)\ =\ \begin{cases} r & \text{if}\ \ \exists\, S_i \in \mathbb{Y}_i\, \exists\, S_j \in \mathbb{Y}_j :\ S_i \xrightarrow{(\alpha,r)} S_j \\ 0 & \text{otherwise} \end{cases}$$

Between any two sub-chains $\mathbb{Y}_i$ and $\mathbb{Y}_j$, transitions of more than one type might connect their states (see Fig. 4.3B). We define $r(\mathbb{Y}_i, \mathbb{Y}_j) = \sum_{\alpha \in (\overrightarrow{\mathcal{A}}^*_s(\mathbb{M})\ \cup\ \overrightarrow{\mathcal{A}}^*_{sl}(\mathbb{M}))} r(\alpha, \mathbb{Y}_i, \mathbb{Y}_j)$ to represent the total rate at which non-boundary states in $\mathbb{Y}_i$ transition into a corresponding state in $\mathbb{Y}_j$ (see Fig. 4.3B).

In Fig. 4.2A, we can observe the above regularities. The transitions enabled by actions $\overrightarrow{\mathcal{A}}^*_s(CS) \cup \overrightarrow{\mathcal{A}}^*_{sl}(CS) = \{req, log, brk, fix\}$ cause the system to leave its current sub-chain to move into a new one. The rate of such a transition depends only on the



(A) The non-boundary states in a sub-chain presenting similar behaviours.

(B) The rates for any one cross-sub-chain transitions.

Figure 4.3: Rate regularities for cross-sub-chain transitions.

configuration of the servers. Since in each sub-chain the state of the servers does not change, the rates of those transitions, if enabled, is the same.

4.2.4 Construction of Aggregated CTMC

For a large-scale model $\mathbb{M}$ that satisfies Condition 1, we can construct its CTMC and use $\vec{\mathcal{A}}^*_s(\mathbb{M})$, $\vec{\mathcal{A}}^*_l(\mathbb{M})$, and $\vec{\mathcal{A}}^*_{sl}(\mathbb{M})$ to detect the sub-chains formed and observe the rate regularities for transitions between sub-chains. These regularities and the assumption that the probability of experiencing the boundary states is negligible, enable us to build an aggregated CTMC for the model. In this CTMC, each sub-chain is represented by a single aggregate state, that captures the configuration of $\boldsymbol{\xi}^s$ in that sub-chain. The aggregated CTMC captures the evolution of the small groups only and is used as the basis of evaluating the behaviour of a system's resources. The downfall is that the information about the behaviour of large groups is lost.

Fig. 4.2B shows the aggregated CTMC of the client-server model. It shows the configurations that the servers experience by performing actions $\{req, log, brk, fix\}$.

We showed the condition which makes a LSRB model amenable to an aggregation. In the next section we will describe how the aggregated state space is directly derived from the model.

4.3 Aggregation Algorithm

The structure of the aggregated CTMC of a model $\mathbb{M}$ that respects Condition 1 is independent of the size of the population levels of the large groups. Therefore, we propose an algorithm which builds the aggregated CTMC *directly* from the model. At first, in a reduction step, our algorithm transforms the system equation of the original model $\mathbb{M}$ into $\mathbb{M}_R$, a *reduced* form of the equation which captures $\mathbb{M}$'s structure only with respect to the groups in $\mathcal{G}_s(\mathbb{M})$. In the next step, using $\mathbb{M}_R$ and the count-oriented semantics developed for PEPA population models, the aggregated CTMC is generated. Note that the reduced system equation faithfully captures the synchronisation restrictions imposed on any of the small groups; the reduction rules guarantee the behaviour derived for $\boldsymbol{\xi}^s$ within $\mathbb{M}_R$ matches the one observed in the original CTMC.

4.3.1 Reduction

These rules will be applied to the system equation of an input model $\mathbb{M}$ in order to produce the reduced model $\mathbb{M}_R$:

$$red(G) = \begin{cases} red(G_1) \underset{L}{\bowtie} red(G_2), & \text{if } G \equiv G_1 \underset{L}{\bowtie} G_2 \\ H\{\cdot\}, & \text{if } G \equiv H\{\cdot\}\ ,\ H \in \mathcal{G}_s(\mathbb{M}) \\ Nil, & \text{if } G \equiv H\{\cdot\}\ ,\ H \in \mathcal{G}_l(\mathbb{M}) \\ red(X), & \text{if } G \stackrel{def}{=} X \end{cases} \tag{4.3}$$

The process *Nil* represents a sequential process which does not undertake any activity. The following rules remove *Nil* processes to find the minimal reduced system equation:

$$Nil \underset{\cdot}{\bowtie} Nil = Nil \qquad Nil \underset{\cdot}{\bowtie} P = P \qquad P \underset{\cdot}{\bowtie} Nil = P$$

4.3.2 Count-Oriented Semantics

Having built the reduced form of $\mathbb{M}$'s system equation, we apply the *count-oriented structured operational semantics* presented below to derive the model's underlying labelled transition system (LTS) *directly* in numerical vector form.

Promotion. Promotion of a sequential component's transition to the group level:

$$\frac{C_u \xrightarrow{(\alpha, r_\alpha)} C'_u}{\xi(H) \rightsquigarrow^{(\alpha, r_\alpha(H))} \Theta(\xi(H), C_u, C'_u)} \; \xi(H, C_u) > 0$$

$r_\alpha(H) =$ apparent rate of α in group H, See Def. 11

Cooperation. Cooperation between groups:

$$\frac{\xi(H_i) \overset{(\alpha, r(\xi(H_i)))}{\rightsquigarrow} \xi'(H_i)}{\xi(H_i) \underset{L}{\bowtie} \xi(H_j) \overset{(\alpha, r(\xi(H_i)))}{\rightsquigarrow} \xi'(H_i) \underset{L}{\bowtie} \xi(H_j)} \; \alpha \notin L$$

$$\frac{\xi(H_j) \overset{(\alpha, r(\xi(H_j)))}{\rightsquigarrow} \xi'(H_j)}{\xi(H_i) \underset{L}{\bowtie} \xi(H_j) \overset{(\alpha, r(\xi(H_j)))}{\rightsquigarrow} \xi(H_i) \underset{L}{\bowtie} \xi'(H_j)} \; \alpha \notin L$$

$$\frac{\xi(H_i) \overset{(\alpha, r_1(\xi(H_i)))}{\rightsquigarrow} \xi'(H_i) \;\wedge\; \xi(H_j) \overset{(\alpha, r_2(\xi(H_j)))}{\rightsquigarrow} \xi'(H_j)}{\xi(H_i) \underset{L}{\bowtie} \xi(H_j) \overset{(\alpha, R)}{\rightsquigarrow} \xi'(H_i) \underset{L}{\bowtie} \xi'(H_j)} \; \alpha \in L$$

$$R = min(r^*_\alpha(H_i), r^*_\alpha(H_j))$$

Constant. Process constants:

$$\frac{M \overset{(\alpha, r)}{\rightsquigarrow} M'}{A \overset{(\alpha, r)}{\rightsquigarrow} M'} \; A \overset{def}{=} M$$

The transition relation constructed by the above rules is denoted by $\rightsquigarrow$ and in its construction, we use the transition relation $\rightarrow$ built by the original PEPA semantics [58] (see the premise of the first rule). However, note that $\rightarrow$ needs to be constructed only at the level of the model's sequential processes; for each sequential process one automaton is constructed expressing the states and transitions each instance of that sequential process experiences.

The semantics formally captures the impact of the completion of an action on each

of the state variables by the update function Θ:

$$\Theta(\boldsymbol{\xi}(H),C_u,C'_u)=\langle\Theta'(\boldsymbol{\xi}(H),C_u,C'_u,\xi(H,C''_u))\,|\,C''_u\in ds^*(H)\rangle$$

$$\Theta'(\boldsymbol{\xi}(H),C_u,C'_u,\xi(H,C''_u)=\begin{cases}\xi(H,C'')-1, & \text{if } C''=C_u\\ \xi(H,C'')+1, & \text{if } C''=C'_u\\ \xi(H,C''), & \text{otherwise}\end{cases}$$

Such an impact was shown in Eq.(2.9); as a result of each transition, a number of state variables decrease and a number of them increase.

4.3.3 Generating the Aggregated CTMC

In the aggregation algorithm we apply the count-oriented semantics to the reduced system equation to obtain its numerical aggregated state space D^{agg}. This is formalised below.

Definition 21. (Aggregated Derivative Set) *The aggregated derivative set of a state S, denoted by $ds^{agg}(S)$, is the smallest set of states satisfying the following conditions.*

1. $S \in ds^{agg}(S)$.
2. *If* $S_1 \in ds^{agg}(S)$ *and* $S_1 \overset{(\alpha,\cdot)}{\rightsquigarrow} S_2$, *then* $S_2 \in ds^{agg}(S)$.

Intuitively, $D^{agg}(S)$ finds all the aggregated states reachable from S using $\rightsquigarrow$.

Definition 22. (Aggregated Labelled Transition System) *For a LSRB model $\mathbb{M}$ which satisfied Condition 1, its aggregated labelled transition system is defined as a tuple $(ds^{agg}(S_0),\Omega,\rightsquigarrow)$. S_0 is the model's initial aggregate state. $\Omega=(\mathcal{A}\times\mathcal{F})$ is the alphabet of transitions' labels where $\mathcal{A}$ is the set of actions defined in $\mathbb{M}$ and $\mathcal{F}$ is a function space defined over $\mathbb{Z}_+^{d_{\mathbb{M}_R}}$. $d_{\mathbb{M}_R}$ is the dimension of the reduced system equation, i.e. the number of state variables appearing in $\boldsymbol{\xi}^s$. Each function in $\mathcal{F}$ corresponds to one action type: F_α defines the apparent rate of action α given the vector space $\boldsymbol{\xi}^s$.*

The combination of $ds^{agg}(S_0)$ and transition relation $\rightsquigarrow$ enable us to build the aggregated derivation graph. Here, each node is one of the states and each arc represents one element of the transition relation. The derivation graph is then used to build the model's underlying CTMC, based on a similar approach to that explained in Sec. 2.3.2.

As an example, consider Fig. 4.2B. This shows the aggregated CTMC constructed for the client-server system when initialised with two idle servers.

4.4 Marginal Distribution over Model's Resources

The aggregated state space D^{agg} is analysed to study the behaviour of a model's resources. The analysis involves constructing the set of C-K equations underlying D^{agg} and finding the solution which captures the transient evolution of a probability distribution over $\boldsymbol{\xi}^s$. In the system of C-K equations, for each state $\boldsymbol{\gamma} \in D^{agg}$ one equation is constructed of the form:

$$\frac{d\,\mathbb{P}_t(\boldsymbol{\gamma})}{d\,t} = - \sum_{\substack{\alpha \in \vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}^{-,s}_\alpha}} r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\boldsymbol{\gamma}) \quad + \sum_{\substack{\alpha \in \vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}^{+,s}_\alpha}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \cdot \mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \quad (4.4)$$

The equations are solved subject to an initial probability distribution over D^{agg}.

4.4.1 Validity of C-K Equations of Aggregated State Space

The aggregation scheme presented above and the derivation of Eq.(4.4) depends on negligibility of boundary states. Obviously, the probability distribution obtained by solving C-K equations of D^{agg} is required to match the marginal distribution over $\boldsymbol{\xi}^s$ derived from solving the C-K equations of the original and complete state space. In this section, we formally establish the connection between the two systems of the C-K equations.

Let $S_i = \langle \boldsymbol{\xi}^s_i , \boldsymbol{\xi}^l_i \rangle$ represent a state in D and in sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}} \in \mathbb{Y}_{\mathbb{M}}$. We have $\boldsymbol{\xi}^s = \boldsymbol{\gamma}$. The C-K equations constructed for S_i is:

$$\frac{d\,\mathbb{P}_t(S_i)}{d\,t} = \underbrace{- \sum_{S_i \xrightarrow{(\alpha, r_\alpha(S_i))} S_j} r_\alpha(S_i) \cdot \mathbb{P}_t(S_i)}_{\text{outward transitions}} \quad \underbrace{+ \sum_{S_j \xrightarrow{(\alpha, r_\alpha(S_j))} S_i} r_\alpha(S_j) \cdot \mathbb{P}_t(S_j)}_{\text{inward transitions}} \quad (4.5)$$

In this equation, we have annotated the terms based on being related to the transitions coming into S_i or the ones leaving S_i.

We aim to calculate $\mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}})$, the probability of being in the sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$. This is equal to the sum of the probabilities of being in any of the states S_i in $\mathbb{Y}_{\boldsymbol{\gamma}}$:

$$\mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}}) = \sum_{S_i \in \mathbb{Y}_{\boldsymbol{\gamma}}} \mathbb{P}_t(S_i)$$

To find $\mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}})$, we sum over instances of equations Eq.(4.5) constructed for $S_i \in \mathbb{Y}_{\boldsymbol{\gamma}}$:

$$\sum_{S_i \in \mathbb{Y}_{\boldsymbol{\gamma}}} \frac{d\,\mathbb{P}_t(S_i)}{d\,t} = \sum_{S_i \in \mathbb{Y}_{\boldsymbol{\gamma}}} \left[- \sum_{S_i \xrightarrow{(\alpha, r_\alpha(S_i))} S_j} r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) \;+ \sum_{S_j \xrightarrow{(\alpha, r_\alpha(S_j))} S_i} r_\alpha(S_j) \cdot \mathbb{P}_t(S_j) \right] \tag{4.6}$$

The left hand side is transformed as:

$$\sum_{S_i \in \mathbb{Y}_{\boldsymbol{\gamma}}} \frac{d\,\mathbb{P}_t(S)}{d\,t} = \frac{d \sum_{S_i \in \mathbb{Y}_{\boldsymbol{\gamma}}} \mathbb{P}_t(S_i)}{d\,t} = \frac{d\,\mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}})}{d\,t} \tag{4.7}$$

In Eq.(4.7), the term associated with $\mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}})$ now appears. The right hand side is transformed below so that the whole equation becomes closed in terms of the probability distribution over the sub-chains.

In our first transformation, we rewrite Eq.(4.6) using the impact vectors:

$$\frac{d\,\mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}})}{d\,t} = \sum_{S_i \in \mathbb{Y}_{\boldsymbol{\gamma}}} \left[\underbrace{- \sum_{\alpha : S_i \geq \mathcal{V}_\alpha^-} r_\alpha(S_i) \cdot \mathbb{P}_t(S_i)}_{\text{outward transitions}} \underbrace{+ \sum_{\substack{\alpha : S_j \geq \mathcal{V}_\alpha^- \\ S_i = S_j + \mathcal{V}_\alpha}} r_\alpha(S_j) \cdot \mathbb{P}_t(S_j)}_{\text{inward transitions}} \right] \tag{4.8}$$

In Eq.(4.8), consider the expression related to the outward transitions. This consists of two summations. The outer sums over states $S_i \in \mathbb{Y}_{\boldsymbol{\gamma}}$, and the inner sums over the action types that enable the transitions leaving S_i. These summations are independent and we can swap their ordering. Note that when swapping, for each action we sum over only the states in which the action is enabled. A state $S_i = \langle \boldsymbol{\xi}_i^s , \boldsymbol{\xi}_i^l \rangle \in D$ enables α if $S_i \geq \mathcal{V}_\alpha^-$, i.e. $\boldsymbol{\xi}_i^s \geq \mathcal{V}_\alpha^{-,s}$ and $\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{-,l}$.

A similar step is applied to the expression related to the inward transitions. Here, the outer summation sums over the states $S_i \in \mathbb{Y}_{\boldsymbol{\gamma}}$, and the inner over the states S_j such that $S_j \xrightarrow{(\alpha,\cdot)} S_i$. For a state S_i, there is a subset of $\vec{\mathcal{A}}^*(\mathbb{M})$ entering S_i. There exists a state $S_j = S_i - \mathcal{V}_\alpha$ enabling an inward transition into S_i with action α when $S_j \geq \mathcal{V}_\alpha^-$. The equivalent form of this necessary condition is $S_i \geq \mathcal{V}_\alpha^+$. This constraint means that when swapping the position of the summations in this expression, for each action type α, we are restricted to sum only over the valid S_j states, which enable an α transition leaving S_j and entering S_i.

Given the above considerations, we swap the summations and obtain:

$$\frac{d\,\mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}})}{d\,t} = \sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \left[- \sum_{S_i : S_i \geq \mathcal{V}_\alpha^-} r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) + \sum_{\substack{S_j : S_j \geq \mathcal{V}_\alpha^- \\ S_i = S_j + \mathcal{V}_\alpha}} r_\alpha(S_j) \cdot \mathbb{P}_t(S_j) \right] \tag{4.9}$$

Abstracting Away From Transitions of Large Groups

In the state space D, we observe that the $\vec{\mathcal{A}}^*_l(\mathbb{M})$ transitions only cause the system to change its state within its current sub-chain and do not affect probability fluxes at the level of the whole sub-chain. This is used as the basis of the next transformation.

Consider Eq.(4.9). We can use the partition $\Delta_{\mathcal{A}} = \{\vec{\mathcal{A}}^*_s(\mathbb{M}), \vec{\mathcal{A}}^*_{sl}(\mathbb{M}), \vec{\mathcal{A}}^*_l(\mathbb{M})\}$ to expand the summation over the action into two separate parts. The first is related to $\alpha \in \vec{\mathcal{A}}^*_l(\mathbb{M})$ and the second to $\alpha \in \vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M})$.

$$\frac{d\,\mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}})}{d\,t} = \sum_{\alpha \in \vec{\mathcal{A}}^*_l(\mathbb{M})} \Bigg[\overbrace{- \sum_{S_i : S_i \geq \mathcal{V}^-_\alpha} r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) + \sum_{\substack{S_j : S_j \geq \mathcal{V}^-_\alpha \\ S_i = S_j + \mathcal{V}_\alpha}} r_\alpha(S_j) \cdot \mathbb{P}_t(S_j)}^{\text{reduces to zero}} \Bigg] + \sum_{\alpha \in \vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M})} \Bigg[- \sum_{S_i : S_i \geq \mathcal{V}^-_\alpha} r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) + \sum_{\substack{S_j : S_j \geq \mathcal{V}^-_\alpha \\ S_i = S_j + \mathcal{V}_\alpha}} r_\alpha(S_j) \cdot \mathbb{P}_t(S_j) \Bigg] \quad (4.10)$$

For each $\alpha \in \vec{\mathcal{A}}^*_l(\mathbb{M})$ the terms in the first line cancel each other, as each $\vec{\mathcal{A}}^*_l(\mathbb{M})$ transition in $\mathbb{Y}_{\boldsymbol{\gamma}}$ is counted twice with opposite signs. Therefore, we only consider the second line and move to the next transformations.

Negligibility of Boundary States

First, we consider the term related to the outward transitions. Recall that any state $S_i \in \mathbb{Y}_{\boldsymbol{\gamma}}$ is written as $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}^l_i \rangle$, and since our model respects Condition 1, the rate of any $\alpha \in \vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M})$ in S_i depends on $\boldsymbol{\gamma}$, $r_\alpha(S_i) = r_\alpha(\boldsymbol{\gamma})$. Thus, we transform the term related to the outward transition as:

$$\sum_{\alpha \in \vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M})} \sum_{S_i : S_i \geq \mathcal{V}^-_\alpha} r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) = \sum_{\alpha \in \vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M})} \sum_{\substack{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}^l_i \rangle \\ \boldsymbol{\gamma} \geq \mathcal{V}^{-,s}_\alpha, \underline{\boldsymbol{\xi}^l_i \geq \mathcal{V}^{-,l}_\alpha}}} r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\langle \boldsymbol{\gamma}, \boldsymbol{\xi}^l_i \rangle) \quad (4.11)$$

For any $\alpha \in \vec{\mathcal{A}}^*_s(\mathbb{M})$, $\mathcal{V}^{-,l}_\alpha = 0$. For such actions, in the inner summation it is enough to check $\boldsymbol{\gamma} \geq \mathcal{V}^{-,s}_\alpha$. However, for any $\alpha \in \vec{\mathcal{A}}^*_{sl}(\mathbb{M})$, both $\boldsymbol{\gamma} \geq \mathcal{V}^{-,s}_\alpha$ and $\boldsymbol{\xi}^l_i \geq \mathcal{V}^{-,l}_\alpha$ must hold. Here, we bring in the idea of boundary states. In $\mathbb{Y}_i$, the probability of being in $bl(\mathbb{Y}_i)$ is close to zero; these are the states where $\boldsymbol{\gamma} \geq \mathcal{V}^{-,s}_\alpha$ and $\boldsymbol{\xi}^l_i \ngeq \mathcal{V}^{-,l}_\alpha$. Thus, in the summation we can remove the restriction on $\boldsymbol{\xi}^l_i \geq \mathcal{V}^{-,l}_\alpha$ and assume that those where

$\boldsymbol{\xi}_i^l \not\geq \mathcal{V}_\alpha^{-,l}$ add negligible contribution to the equation. We then obtain:

$$\sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} \sum_{S_i : S_i \geq \mathcal{V}_\alpha^-} r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) \approx \sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle, \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{-,s}} r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle) \quad (4.12)$$

By factoring out $r_\alpha(\boldsymbol{\gamma})$ from the inner summation on the right hand side and using the substitution $\sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle, \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{-,s}} \mathbb{P}_t(\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle) = \Big(\mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}})\Big)_{\boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{-,s}}$ we have:

$$\sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} \sum_{S_i : S_i \geq \mathcal{V}_\alpha^-} r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) \approx \sum_{\substack{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{-,s}}} r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}}) \quad (4.13)$$

Eq.(4.13) is substituted into Eq.(4.8) as the term related to the outward transitions. We apply similar steps to the term related to inward transitions. By considering the probability of being in boundary states in all sub-chain to be close to zero, this term is approximated by:

$$\sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} \sum_{\substack{S_j : S_j \geq \mathcal{V}_\alpha^- \\ S_i = S_j + \mathcal{V}_\alpha}} r_\alpha(S_j) \cdot \mathbb{P}_t(S_j) \approx \sum_{\substack{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{+,s}}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \mathbb{P}_t(\mathbb{Y}_{(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)}) \quad (4.14)$$

Having applied the above transformations, Eq.(4.8) is reformulated in the following form that is closed in terms of the probability distribution defined over the sub-chains:

$$\frac{d\, \mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}})}{d\, t} \approx \sum_{\substack{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{-,s}}} r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}}) + \sum_{\substack{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{+,s}}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \mathbb{P}_t(\mathbb{Y}_{(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)}) \quad (4.15)$$

As the final step, we observe that each sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$ in D is represented by $\boldsymbol{\gamma}$ in D^{agg}. This means that $\mathbb{P}_t(\boldsymbol{\gamma}) = \mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}})$. We use this equality $\mathbb{P}_t(\boldsymbol{\gamma}) = \mathbb{P}_t(\mathbb{Y}_{\boldsymbol{\gamma}})$ in Eq.(4.15) and finally derive Eq.(4.4).

With the above transformations, we showed that the analysis of the aggregated model gives rise to a distribution that approximates the exact marginal distribution over $\boldsymbol{\xi}^s$. Since D^{agg} abstracts away from the transitions of the large groups, it can be

efficiently analysed, and we can then obtain the stochastic behaviour of the model's small groups with maximum faithfulness. In doing this, the construction of the complete state space and the transformations exhibited above are completely avoided, as we have shown for any LSRB model, the paths through the transformations and the aggregation give rise to the same set of equations.

In the next section, we consider an experiment where the accuracy of the approximate probability distribution is investigated.

4.5 Accuracy of the Aggregation

The accuracy of the marginal probability distribution derived via the aggregation depends on the the negligibility of the probability of being in the boundary states. If, at all times, the small groups in the model are under heavy load and their cooperation capacity is saturated by the demand of the large groups, one would expect to get highly accurate approximate marginal probability distribution. Conversely, the approximation method leads to an erroneous marginal distribution if the probability of being in boundary states is not negligible. In this section, we report the results of an experiment where we investigate the accuracy of the aggregation method.

We consider the following client-server model:

$$C_{think} \stackrel{def}{=} (think, r_t).C_{req} \qquad C_{req} \stackrel{def}{=} (req, \top).C_{think}$$

$$S_{idle} \stackrel{def}{=} (req, r_s).S_{log} + (brk, r_b).S_{broken}$$

$$S_{log} \stackrel{def}{=} (log, r_l).S_{idle} \qquad S_{broken} \stackrel{def}{=} (fix, r_f).S_{idle}$$

$$CS \stackrel{def}{=} Servers\ \{\ S_{idle}[n_s]\ \} \underset{\{req\}}{\bowtie} Clients\ \{\ C_{think}[n_c]\ \}$$

and assume that the small group, *Servers*, contains five servers and the large group, *Clients*, 100 clients. Recall that the state vector which captures the state of the system is $\langle S_i, S_l, S_b, C_t, C_r \rangle$ and the one for the aggregated model is $\langle S_i, S_l, S_b \rangle$. In the complete state space, the boundary states are those where at least one server enables action *req* ($S_i \geq 0$) but no client requests it ($C_r = 0$). We constructed three versions of the model and across these, the following parameters were the same: $n_s = 5$, $n_c = 100$, $r_s = 10$, $r_l = 50$, $r_b = 0.005$, $r_f = 0.005$ and $r_c = \top$. For the rate of a client' *think* action, in the first version $r_t = 15$, in the second $r_t = 0.2$ and in the third $r_t = 0.1$. The change in r_t causes a gradual increase in the probability of being in boundary states (see Fig. 4.4).

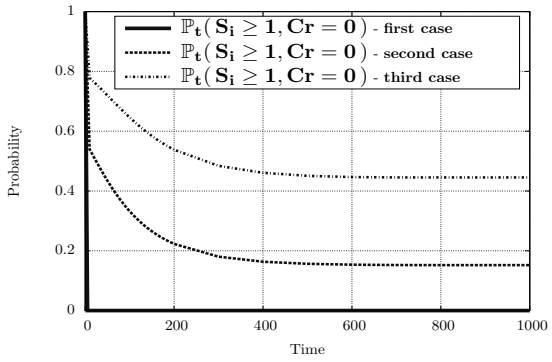
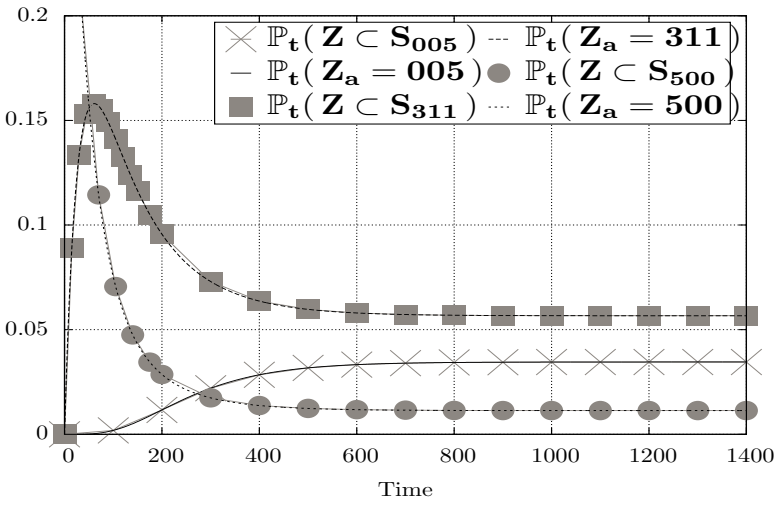


Figure 4.4: Probability of being in boundary states ($\mathbb{P}_t(S_i \geq 0, C_r = 0)$) for each case of the experiment.

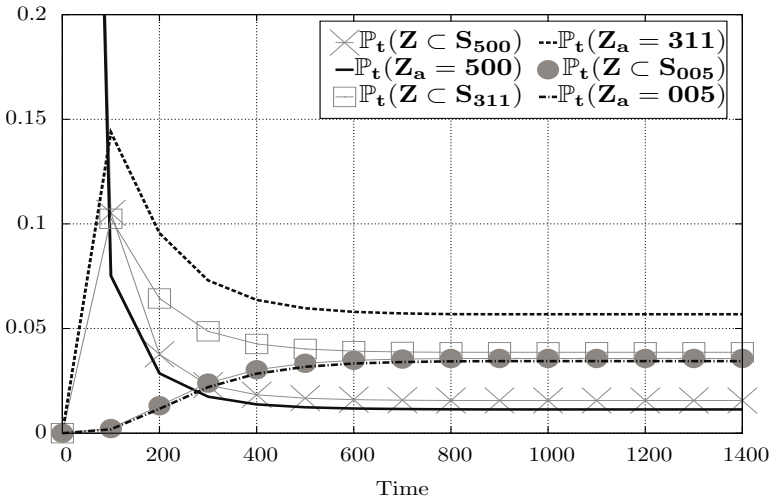
For each version, we calculated an approximate probability distribution over the state space of the servers via the aggregation and compared it with a similar distribution derived from the exact analysis of the complete state space using PRISM [68]. There are multiple ways for comparing two probability distributions. For simplicity, we chose three different representative states from the distributions and compared the distributions only with respect to those representative states. Our comparison could readily be extended to the complete distributions. In this section, Z denotes the stochastic process represented by the client-server model's complete state space and Z_a denotes the stochastic process associated with the model's aggregated state space.

The parameters of the first version ($r_t = 15$) cause the servers to be under heavy contention at all times; the probability of $C_r = 0$ is close to zero. Fig. 4.5A shows a comparison between probabilities calculated for three representative states $\langle 5,0,0\rangle$, $\langle 3,1,1\rangle$ and $\langle 0,0,5\rangle$. As an example, $\mathbb{P}_t(Z \subset (3,1,1))$, denotes the probability that in the complete state space, the system resides in a state where there are three idle, one logging and one broken server and $\mathbb{P}_t(Z_a = (3,1,1))$ captures the probability of being in the associated state in the aggregated state space.

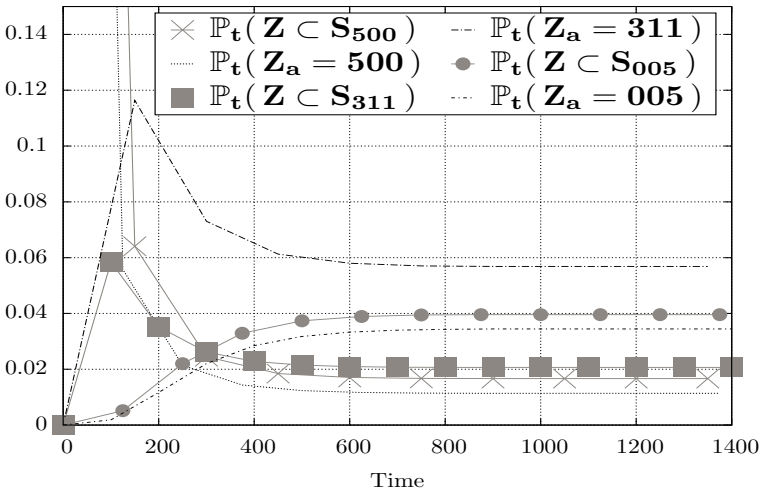
In the second case, $r_t = 0.2$, thinking has a longer duration which slows the flow of clients into the state of requesting communication. Thus, the probability of $C_r = 0$ becomes higher. The same measures were calculated for the second case and the results are reported in Fig. 4.5B. Since the probability of being in the boundary states is higher the discrepancy between the results obtained via the aggregation and the exact method is larger. In the third case $r_t = 0.1$ and the probability of $C_r = 0$ is relatively high (see Fig. 4.4). Hence, the deviation of the approximate distribution from the exact one is significantly larger than the previous cases. The outputs for this case are shown in



(A) $\mathbb{P}(Z \subset (5,0,0))$, $\mathbb{P}(Z \subset (3,1,1))$, $\mathbb{P}(Z \subset (0,0,5))$ first case.



(B) $\mathbb{P}(Z \subset (5,0,0))$, $\mathbb{P}(Z \subset (3,1,1))$, $\mathbb{P}(Z \subset (0,0,5))$ second case.



(C) $\mathbb{P}(Z \subset (5,0,0))$, $\mathbb{P}(Z \subset (3,1,1))$, $\mathbb{P}(Z \subset (0,0,5))$ third case.

Figure 4.5: Comparison of exact and approximate probabilities of being in three representative states across the experimental cases.

	First Case			Second Case			Third Case		
	exact	aprox.	error	exact	aprox.	error	exact	aprox.	error
$\mathbb{P}_{\langle Z_a=500\rangle}$	0.011	0.011	0	0.015	0.011	26	0.016	0.011	31
$\mathbb{P}_{\langle Z_a=311\rangle}$	0.056	0.056	0	0.038	0.056	47	0.020	0.056	180
$\mathbb{P}_{\langle Z_a=005\rangle}$	0.034	0.034	0	0.035	0.034	2	0.039	0.034	12
$\mathbb{P}_{\langle Z_a\subset E^5\rangle}$	0.028	0.028	0	0.023	0.028	21	0.02	0.028	40
$\mathbb{P}_{\langle Z_a\subset E^2\rangle}$	0.310	0.317	2	0.328	0.317	3	0.336	0.317	5.6
$\mathbb{P}_{\langle Z_a\subset E^1\rangle}$	0.161	0.165	2.48	0.171	0.165	3.5	0.189	0.165	12

Table 4.2: Comparion of the experiment results calculated separately by the approximate and exact methods. The error is presented as a percentage.

Fig 4.5C.

The aggregated state space can be used for deriving further performance indicators, such as dependability measures concerning systems' resources. Assume that the measure of interest is the number of working servers, i.e. those which are not broken. Formally, let E^K denote the probabilistic event that there are K servers running. For instance, when K=5:

$$\mathbb{P}\left(Z\subset E^5\right)=\mathbb{P}(Z\subset(5,0,0))+\mathbb{P}(Z\subset(4,1,0))+\ldots+\mathbb{P}(Z\subset(0,5,0))\approx$$
$$\mathbb{P}\left(Z_a\subset E^5\right)=\mathbb{P}(Z_a=(5,0,0))+\mathbb{P}(Z_a=(4,1,0))+\ldots+\mathbb{P}(Z_a=(0,5,0))$$

Using the aggregation method, the steady state values of $\mathbb{P}\left(Z\subset E^K\right), K=1,2,5$ and the previously presented outputs were calculated and compared against the corresponding exact results (see Table 4.2). The comparison provides evidence that a higher probability of being in the model's boundary states corresponds to a less accurate aggregation.

In this experiment, using a machine with 4G of RAM and 2.3GHz processors the derivation of the approximate steady state marginal distribution through the aggregated CTMC took 10–15 seconds. Deriving the same distribution by the exact analysis took 650–700 seconds. The former takes advantage of the possibility of aggregation whereas the latter, derives the marginal distribution from a state space where the detailed dynamics of the clients are also captured. The number of states in the complete state space of the model is 2121 whereas the aggregated one has 21 states. Note that the latter is invariant with respect to the number of clients, highlighting the scalability

advantage provided by the aggregation method.

Applying the aggregation method relies on prior knowledge about the model's behaviour with respect to its boundary states. Such knowledge can be supplied by the domain experts or by monitoring the real system. In large-scale resource-bound communication networks, the resources are almost continuously under contention; e.g. processing input transactions or dealing with a frequent incoming flow of packets. Thus, for such systems, the probability of resources waiting for the users is close to zero and the aggregation algorithm proposed can be safely used.

Chapter 5

Efficient Detection of Time-Scale Near-Complete Decomposability

5.1 Introduction

One important method that is suitable for the analysis of CTMCs with large state spaces is *time-scale near-complete decomposability* (TSND) [33]. The method is applicable to CTMCs whose transitions occur across slow and fast time scales. Given a CTMC, its state space is divided into a number of blocks in such a way that the transitions within the blocks capture the fast dynamics and those across the blocks are related to changes happening over the slow time scale. Then the CTMC is analysed by the TSND solution method that comprises of the following steps:

1. Decomposition. Each block is treated as one CTMC by disregarding the cross-block transitions. For each block, its CTMC is evaluated in isolation to obtain a partial solution associated with that block.

2. Block level analysis. A CTMC is constructed by disregarding the transitions within the blocks and only considering the cross-block transitions. Here, each block is represented by a single state. The CTMC is analysed to give us the block level solution.

3. Re-composition. The partial solutions and the block level solution are recomposed to derive an approximate probability distribution over the complete state space.

The main advantage of the TSND approach is that the blocks are focused upon one at a time, and therefore, the construction of the complete state space is avoided.

As we stated in Sec. 1.1, LSRB models are composed of different types of components which potentially exhibit behaviours evolving at different time scales. Additionally, the state spaces of such models are massive. These features identify the TSND method as an appropriate analysis approach for LSRB models. Note that for the method to be useful, it needs to be integrated into the high-level formalism; the slow and fast actions (which give rise to slow and fast transitions) need to be specified at the level of the model description, and then these must be automatically used for exploration of the blocks of the underlying state spaces [73].

When applying the TSND to LSRB models, we are confronted with one challenging question: given that the state space cannot be fully constructed, what is the partition on the model's action which divides them into slow and fast categories? Which actions cause the inner block transitions and which ones cause the cross-block transitions? The trivial approach is that the partition is specified manually by the modeller. However, this approach does not scale up and limits the automatic application of TSND method. Answering the question is a key part of the TSND method and any procedure which helps with efficient detection of the partitions is of great importance [5].

In this chapter, we consider the above questions and present an efficient procedure for deriving the slow / fast partition in the context of LSRB models. Our procedure relies on the aggregation method that was presented in the previous chapter and has two main steps. First, given a model $\mathbb{M}$ and its aggregated state space D^{agg}, we apply a reachability analysis based on the hierarchical clustering [12] on D^{agg}. This gives us a slow / fast partition over $\overrightarrow{\mathcal{A}}^*_s(\mathbb{M}) \cup \overrightarrow{\mathcal{A}}^*_{sl}(\mathbb{M})$, the set of actions in which the small groups are involved. Then, the partition corresponding to the original model is derived by keeping the set of slow actions the same and adding the actions related to the large group $\overrightarrow{\mathcal{A}}^*_l(\mathbb{M})$ to the set of fast transitions derived via D^{agg}. The intuition is that since the $\overrightarrow{\mathcal{A}}^*_l(\mathbb{M})$ are enabled by large populations, they continually occur at high rates and belong to the fast time scale. The output is a slow / fast partition on $\overrightarrow{\mathcal{A}}^*(\mathbb{M})$ which can then be fed into the TSND solution method described above.

Our time-scale detection algorithm is supported by a theorem regarding our aggregation method. The theorem formally proves that under some mild conditions, the same slow / fast categorisation observed for $\overrightarrow{\mathcal{A}}^*_s(\mathbb{M}) \cup \overrightarrow{\mathcal{A}}^*_{sl}(\mathbb{M})$ transitions of the aggregated state space is observable within the original state space. This means that when

searching for the decomposable TSND blocks, instead of a prohibitively expensive reachability analysis of the complete state space, one can search for them significantly more efficiently within the aggregated state space. The blocks found will be valid in the complete state space. The slow / fast partitioning algorithm is efficient and also automated. This means that it enables us to easily apply the TSND solution to LSRB models in an automatic fashion.

The structure of this chapter is the following. In Sec. 5.2, we introduce the basics of TSND. We describe how it is applied to CTMCs and LSRB models. In Sec. 5.3, we introduce our enhanced procedure for detecting the slow and fast partition. In Sec. 5.4, we report the results of an experiment, where we investigated the procedure's scalability and usefulness.

5.2 Time-Scale Near-Complete Decomposability

In this section, we review the steps involved when applying the TSND method to DTMCs and CTMCs. The method is mainly used for steady state analysis. This type of analysis will be the focus of this section. The method of TSND was originally developed for DTMC [5], where the notion of time is captured via predefined discrete time steps. In our presentation, first we review the method for the case of DTMCs and then introduce how it is adapted for CTMCs.

5.2.1 Decomposability Condition for DTMCs

In general, the steady state distribution of a DTMC with n states is obtained by solving the following system of linear equations.

$$\Pi R = \Pi \qquad \text{subject to} \quad \sum_{i=1}^{n} \Pi(i) = 1 \tag{5.1}$$

where the vector Π with dimension $1 \times n$ captures the steady state distribution of the DTMC and R is the associated transition matrix. The structure of R is important in solving the system of equations. To illustrate this, we introduce the following definitions.

Definition 23. (Block Diagonal Matrix). *A block diagonal matrix is a square matrix in which the principal diagonal consists of block matrices and the off-diagonal elements*

are zero. A block diagonal matrix E has the form

$$E = \begin{bmatrix} E_1 & 0 & \dots & 0 \\ 0 & E_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & E_{N_B} \end{bmatrix} \tag{5.2}$$

where each E_k is a square matrix.

Let $ord(\cdot)$ be a function defined on the square matrices which returns the number of rows or columns of a given matrix. Given E in the above form:

$$\sum_{i=1}^{N} ord(E_i) = ord(E)$$

Using Def. 23, we describe how a DTMC is *completely decomposable*. Consider a DTMC $X^{\mathcal{D}}$ with stochastic matrix[1] R of order n. The vector $\boldsymbol{x}^{\mathcal{D}}(k)$ describes the probability distribution of $X^{\mathcal{D}}$ at step k. $\boldsymbol{x}^{\mathcal{D}}(k+1)$ can be obtained by:

$$\boldsymbol{x}^{\mathcal{D}}(k+1) = \boldsymbol{x}^{\mathcal{D}}(k) \times R \tag{5.3}$$

Here element $x_l^{\mathcal{D}}$ in $\boldsymbol{x}^{\mathcal{D}}(k)$ is the probability of the system being in state l at step k and element r_{hl}^{D} in R shows the conditional probability that $X^{\mathcal{D}}$ is in state l at step k given that it was in state h at step $(k-1)$. In general, $\boldsymbol{x}^{\mathcal{D}}(k)$ can be obtained by repetitive multiplication of the initial state $\boldsymbol{x}^{\mathcal{D}}(0)$ by R:

$$\boldsymbol{x}^{\mathcal{D}}(k) = \boldsymbol{x}^{\mathcal{D}}(0) \times R^k \tag{5.4}$$

Definition 24. (Completely Decomposable DTMC). *The matrix R is said to be* completely decomposable *if the rows and columns of R can be identically rearranged so that $R = R^*$ where R^* is a block diagonal matrix with the following form:*

$$R^* = \begin{bmatrix} R_1^* & 0 & \dots & 0 \\ 0 & R_2^* & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{N_B}^* \end{bmatrix} \tag{5.5}$$

Moreover, a DTMC is said to be completely decomposable if its transition matrix is completely decomposable.

[1] We assume that our stochastic matrices are *right* stochastic, i.e. the matrices consist of elements between zero and one and the sum of the elements in each row is equal to one.

The state space of a completely decomposable DTMC can be partitioned into a number of blocks where the states within each block are not reachable from states of the other blocks. If $X^{\mathcal{D}}$ is completely decomposable, its state vector $\boldsymbol{x}^{\mathcal{D}} = \boldsymbol{x}^{\mathcal{D}^*}(t) = [x_l^*(t)]$ can be divided into a set of subvectors:

$$\boldsymbol{x}^{\mathcal{D}^*}(k) = [x_l^*(k)] = \Big[[x_{i_1}^*(k)], \ldots, [x_{i_I}^*(k)], \ldots, [x_{i_N}^*(k)]\Big]$$

where if, $x_{i_I}^*(k) = x_l^*(k)$, then $l = \sum_{J=1}^{I-1} ord(J) + i$. Each subvector $[x_{i_I}^*(t)]$ is related to the corresponding square matrix R_I^*. When analysing $X^{\mathcal{D}}$, instead of using Eq.(5.4), the distribution of the decomposable DTMC at any step k or at the equilibrium can be obtained by solving a number of simpler equations. The subvector $[x_{i_I}^*(k)]$ depends only on $[x_{i_I}^*(0)]$ and R_I^*, and is independent of $[x_{i_J}^*(k)]$ and R_J^*, $J \neq I$ for any step k.

Using the notion of completely decomposable DTMC, next we introduce the notion of *near-completely* decomposable DTMC.

Definition 25. (Near-completely Decomposable DTMC)-*(Chp. 2, [87]). A DTMC $X^{\mathcal{D}}$ is near-completely decomposable if the rows and columns of its transition matrix R can be identically rearranged such that:*

$$R \;=\; R^* \;+\; \varepsilon^{\mathcal{D}} \;\cdot\; C \tag{5.6}$$

where:

1. *R^* is a stochastic matrix of the same order as R in block structure.*
2. *$\varepsilon^{\mathcal{D}}$ is a real positive number, close to zero and small compared to elements of R^*.*
3. *C is a square matrix of the same order as R with the property of keeping both R and R^* stochastic (all R_I^*, $I = 1 \ldots N$ are square and stochastic).*

Intuitively, in order for R to satisfy Eq.(5.6), it should consist of principal square sub-matrices, the elements of which are significantly larger than the elements outside. Equivalently, the DTMC satisfies Eq.(5.6) when its state space can be partitioned in such a way that the probabilities related to the transitions within each block are at least an order of magnitude larger than the probabilities associated with the cross-block transitions.

Eq.(5.6) is a formal characterisation of the property of near-complete decomposition. For a DTMC whose transition matrix exhibits the required structural property an

instance of this equation with concrete values of R^* and C can be constructed. In fact, it is usually the case that for such a DTMC, many instances of Eq.(5.6) can possibly be introduced. Here, we show a systematic construction of a solution in which $\varepsilon^{\mathcal{D}}$ and C exhibit certain properties. This construction will be used in a theorem presented later.

5.2.2 Construction of a Solution

We define $\varepsilon^{\mathcal{D}}$ as a measure that captures the *maximum probabilistic coupling* between any two different blocks in R. The measure is derived by considering the probabilities associated with the cross-block transitions:

$$\varepsilon_m^{\mathcal{D}} = \max_{i_I} \left(\sum_{J=1,J\neq I}^{N} \sum_{j=1}^{ord(J)} r_{i_I j_J} \right) \tag{5.7}$$

where i_I represents the i^{th} state in block I and $r_{i_I j_j}$, an element of R, is the probability associated with the transition from state i_I into state j_J which resides in a different block J. Substituting $\varepsilon_m^{\mathcal{D}}$ in Eq.(5.6), we show that the equation has at least one solution.

Defining $\varepsilon^{\mathcal{D}} = \varepsilon_m^{\mathcal{D}}$ causes all elements of C to be between -1 and 1, $|c_{i_I j_J}| \leq 1$:

$$\max_{i_I} \left(\sum_{J=1,J\neq I}^{N} \sum_{j=1}^{ord(J)} c_{i_I j_J} \right) = -\max_{i_I} \left(\sum_{k=1}^{ord(J)} c_{i_I k_I} \right) = 1 \tag{5.8}$$

Since matrices R and R_I^*, $I = 1,2,\ldots N$ are required to be stochastic, the row sums of C are equal to zero. We choose non-positive values for all elements $c_{i_I k_I}$ and positive values for all elements $c_{i_I k_J}$, $I \neq J$ [33]. In C, for each row i_I associated with state i_I we have:

$$\sum_{k=1}^{ord(I)} c_{i_I k_I} = -\sum_{J=1,J\neq I}^{N} \sum_{j=1}^{ord(J)} c_{i_I j_J} \tag{5.9}$$

Using $\varepsilon^{\mathcal{D}}$ and C, we can construct the instance of $R = R^* + \varepsilon^{\mathcal{D}} \cdot C$ with the following entities:

$$r^*_{i_I j_J} = \begin{cases} 0 & \text{if } I \neq J \\ r_{i_I j_J} + \dfrac{r_{i_I j_J}}{\left(\sum_{j=1}^{ord(I)} r_{i_I j_J}\right)} \sum_{J=1,J\neq I}^{N} \sum_{j=1}^{ord(J)} r_{i_i j_J} & \text{if } I = J \end{cases} \tag{5.10}$$

$$c^*_{i_I j_J} = \begin{cases} +\dfrac{1}{\varepsilon^D} \times r_{i_I j_J} & \text{if } I \neq J \\ -\dfrac{1}{\varepsilon^D} \times \dfrac{r_{i_I j_J}}{\left(\sum_{j=1}^{ord(I)} r_{i_I j_J}\right)} \times \sum_{J=1,J\neq I}^{N} \sum_{j=1}^{ord(J)} r_{i_i j_J} & \text{if } I = J \end{cases} \tag{5.11}$$

The construction assumes that the cross-block transitions are disregarded. Thus, for any block I the probability flux associated with its cross-block transitions is distributed across the states within the block. In other words, the rate $r_{i_I j_J}$, $I = J$ in R^* is adjusted by adding the fraction $\frac{r_{i_I j_J}}{(\sum_{j=1}^{ord(I)} r_{i_I j_I})}$ of the total rate $\sum_{J=1,J\neq I}^{N} \sum_{j=1}^{ord(J)} r_{i_i j_J}$ associated with transitions leaving block I. We can prove that using the above solution the matrix R^{D^*} is block diagonal and each R_I^* is stochastic, thus, respecting the requirements defined for the solution. The proof is shown in Appendix. A.1.

Example

Consider the stochastic matrix:

$$R = \left[\begin{array}{cc|c} 0.5 & 0.45 & 0.05 \\ 0.6 & 0.375 & 0.025 \\ \hline 0.025 & 0.025 & 0.95 \end{array}\right]$$

Using $\varepsilon^{\mathcal{D}} = (0.025 + 0.025)$, the solution of $R = R^* + \varepsilon^{\mathcal{D}} C$ is given by:

$$\left[\begin{array}{cc|c} 0.5 & 0.45 & 0.05 \\ 0.6 & 0.375 & 0.025 \\ \hline 0.025 & 0.025 & 0.95 \end{array}\right] = \left[\begin{array}{cc|c} 0.5263 & 0.4737 & 0 \\ 0.6154 & 0.3846 & 0 \\ \hline 0 & 0 & 1 \end{array}\right] + 0.05 \left[\begin{array}{ccc} -0.5263 & -0.4737 & 1 \\ -0.3077 & -0.1923 & 0.5 \\ 0.5 & 0.5 & -1 \end{array}\right] \quad (5.12)$$

5.2.3 Decomposability Condition in CTMCs

We presented the notion of decomposability for DTMCs. PEPA models have semantics in terms of CTMCs. Using the definitions above and considering that each CTMC can be represented by its uniformised DTMC, we extend the notion for CTMCs.

Definition 26. (Uniformised DTMC of a CTMC [102]) *Consider a CTMC X^C with the infinitesimal generator matrix Q. Note that in Q, $q_{ii} = -\sum_{j=1,j\neq i}^{n} q_{ij}$. Let $\lambda(i) = -q_{ii}$ denote the exit rate of state i. The uniformised process $X^{\mathcal{D}}$ with stochastic matrix R is a DTMC derived by the linear transformation:*

$$R = I + \frac{Q}{\lambda} \quad (5.13)$$

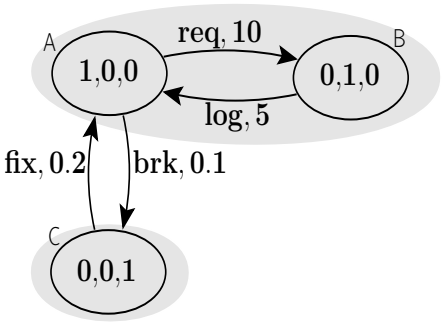


Figure 5.1: The aggregated state space of the client-server system initialised with one server. The state vector is $\langle S_i, S_l, S_b \rangle$.

where λ *is a constant satisfying* $\lambda \geq max(\lambda(i))$.

We refer to λ as the *uniformisation factor*. In this thesis, for simplicity we assume $\lambda = max(\lambda(i))$. Using Def. 25, we define the notion of decomposability for CTMCs.

Definition 27. (Near-completely decomposable CTMC). *A CTMC* X^C *is regarded to be near-completely decomposable if its uniformised process* $X^{\mathcal{D}}$ *is near-completely decomposable.*

In the infinitesimal generator matrix Q of a near-completely decomposable CTMC X^C, the rows and columns can be rearranged into a quasi block structure in a way that the rates within each diagonal block are substantially larger than the off-diagonal elements. The expression $R = R^* + \varepsilon^{\mathcal{D}} \times C$ is used to form an analogous expression for Q. By substituting the elements of the former using the uniformisation, we have:

$$R \;=\; I + \frac{Q}{\lambda} \;=\; R^* + \varepsilon^{\mathcal{D}}\, C \;\Longrightarrow\; Q \;=\; (\lambda\, R^* - \lambda\, I) + (\lambda\, \varepsilon^D)\, C$$

Then, using $I + \frac{Q^*}{\lambda} = R^*$ and $\varepsilon^C = \lambda\, \varepsilon^D$ we obtain the following for the CTMC:

$$Q \;=\; Q^* + \varepsilon^C \cdot C \tag{5.14}$$

Note that C was not required to change. Similarly to the case of DTMC, Eq.(5.14) admits multiple solutions. We can follow a procedure similar to the DTMC one to construct one solution. The details of this is not presented again. Instead, we only present the following example.

Example

Consider the client-server model of Page 44 initialised with only one server. Fig. 5.1 shows the model's aggregated state space. This characterises the process $X_{CS_{agg}}$ with

infinitesimal matrix $Q_{CS_{agg}}$ that appears on the left hand side of Eq.(5.15). X_{CS} respects the decomposability condition with the blocks highlighted in Fig. 5.1. The transitions of types *break* and fix take place on a slower time scale than those of types *req* and log. Thus $Q_{CS_{agg}}$ can be reformulated as:

$$\begin{array}{c} \\ A \\ B \\ C \end{array}\begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \left[\begin{array}{cc|c} -10.1 & 10 & 0.1 \\ 5 & -5 & 0 \\ \hline 0.2 & 0 & -0.2 \end{array}\right] \end{array} = 10.1 \left(\left[\begin{array}{ccc} 0 & 1 & 0 \\ \frac{5}{10.1} & \frac{5.1}{10.1} & 0 \\ 0 & 0 & 1 \end{array}\right] - \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] \right)$$

$$+ (10.1)(\frac{0.2}{10.1}) \left[\begin{array}{ccc} 0 & -\frac{0.1}{0.2} & \frac{0.1}{0.2} \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{array}\right] \qquad (5.15)$$

$$= \left[\begin{array}{ccc} -10.1 & 10.1 & 0 \\ 5 & -5 & 0 \\ 0 & 0 & 0 \end{array}\right] + (0.2) \left[\begin{array}{ccc} 0 & -\frac{0.1}{0.2} & \frac{0.1}{0.2} \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{array}\right]$$

In general, dealing with completely decomposable processes is trivial. In this chapter, we are focusing on those being near-completely decomposable (NCD).

5.2.4 Solution Methods - Application to LSRB Models

A DTMC (or CTMC) that satisfies the near-complete decomposability condition can be analysed by the decomposition / re-composition method that we presented on Page 79. The method can also be extended such that it can also be directly applied to the process algebra models. In this section, first we propose the details of the solution method when applied to near-completely decomposable DTMCs and then describe how the it is used for LSRB models.

5.2.4.1 Solving Nearly Decomposable Markov Chains

We consider DTMC $X^{\mathcal{D}}$ with transition matrix R that satisfies $R = R^* + \varepsilon \cdot C$. Suppose that R^* has the form shown in Eq.(5.5). Let B_I represent the set of states associated

with the block $R_I^*, I \in [1 \cdots N_B]$. Each B_I has $ord(I)$ individual states. Let $\Pi = [\pi_k]_{k=1}^{ord(R)}$ denote $X^{\mathcal{D}}$'s steady state distribution and $\Pi_I = [\pi_{i_I}]_{i=1}^{ord(I)}$ be the projection of Π onto B_I. We define θ_I to be the steady state probability of X^D being in block B_I, $\theta_I = \sum_{i=1}^{ord(I)} \pi_{i_I}$. We define the vector $\boldsymbol{p}_I$ to represent the steady state distribution of the DTMC whose states are expressed by B_I and transitions are governed by R_I^*. The decomposability of $X^{\mathcal{D}}$ means that it reaches short term quasi-equilibrium between pairs of events happening on the slow time scale (the cross-block transitions). This means that the distribution Π_I can be accurately approximated as:

$$\forall \ I \in [1 \cdots N_B] \ : \ \Pi_I \approx \boldsymbol{p}_I \times \theta_I \tag{5.16}$$

Eq. (5.16) suggests a decomposition / re-composition algorithm for obtaining an approximation to Π. First, for each $I \in [1, \cdots, N]$ we derive $\boldsymbol{p}_I$ by considering the transition matrix R_I^*. Second, we derive the block level DTMC $X_{block}^{\mathcal{D}}$ with transition probabilities:

$$r_{IJ} = \sum_{i=1}^{ord(I)} \mathbf{p}_{i_I} \times \sum_{j=1}^{ord(J)} p_{i_I j_J} \tag{5.17}$$

For any $I, J \in B$, r_{IJ} is the sum of the cross-block transitions connecting the states in I to J (the details of this is shown in Appendix. A.2). We analyse $X_{block}^{\mathcal{D}}$ and obtain θ_I. Finally, Eq.(5.16) is used to find Π.

The procedure above is more efficient than the analysis of the complete state space as at any given time, only one block is actively analysed. In the following, we show an example of this procedure.

Example

Consider Eq.(5.12). The transition matrices related to the blocks are:

$$R_1^* = \begin{bmatrix} 0.5263 & 0.4737 \\ 0.6154 & 0.3846 \end{bmatrix} \qquad R_2^* = \begin{bmatrix} 1 \end{bmatrix}$$

with equilibrium distributions $\mathbf{p}_1 = [0.5651 \quad 0.4349]$, $\mathbf{p}_2 = [1]$. The transition matrix of the block level DTMC is

$$\begin{bmatrix} 0.9609 & 0.0391 \\ 0.05 & 0.95 \end{bmatrix}$$

At the equilibrium, $\theta_1 = 0.5612$ and $\theta_2 = 0.4388$. Using Eq. (5.16) we have: $\Pi \approx [\ \mathbf{p}_{1_1} \times \theta_1 \quad \mathbf{p}_{2_1} \times \theta_1 \quad \mathbf{p}_{1_2} \times \theta_2\ ] = [\ 0.3171 \quad 0.2441 \quad 0.4388\ ]$. The exact steady state distribution is $\Pi = [\ 0.3161 \quad 0.2452 \quad 0.4387\ ]$.

5.2.4.2 Application on LSRB Models

Mertsiotakis [73] developed an algorithm that enables us to apply the solution method explained above to models expressed in stochastic process algebra. We used the algorithm's main idea to devise one suitable for LSRB models. Our algorithm is shown in Alg. 1.

The input is a model $\mathbb{M}$ and partition $\Delta_{\mathcal{T}} = \{\vec{\mathcal{A}}^*_{slow}, \vec{\mathcal{A}}^*_{fast}\}$ over $\mathbb{M}$'s set of actions $\vec{\mathcal{A}}^*(\mathbb{M})$. The actions in $\vec{\mathcal{A}}^*_{slow}$ enable the transitions that happen on the slow time scale and $\vec{\mathcal{A}}^*_{fast}$ are related to those happening on the fast time scale. The initial state is used to derive the initial block which will be the first one to be analysed. For each block its set of internal states are derived by considering the transitions of type $\vec{\mathcal{A}}^*_{fast}$. Moreover, while exploring the block we are simultaneously tracking the $\vec{\mathcal{A}}^*_{slow}$ transitions that cause the system to leave the current block, and update the list of blocks that need to be explored later. The blocks are progressively analysed and collected, with the effect that at any given time the states of at most one block are kept in the memory. When all blocks have been internally explored, the block-level CTMC is constructed and analysed. Finally, the re-composition step is performed.

5.3 Automatic Detection of Slow / Fast Categorisation

In the previous section, we proposed an algorithm for applying the TSND solution method to LSRB models and observed that one of the inputs of the solution algorithm is the partition $\Delta_{\mathcal{T}} = \{\vec{\mathcal{A}}^*_{slow}, \vec{\mathcal{A}}^*_{fast}\}$ over the set $\vec{\mathcal{A}}^*(\mathbb{M})$. Specifying such a partition is a difficult task and needs to be done rigorously, as it has consequences in terms of the accuracy of the approximation. In this section first we introduce a naive procedure for the automatic derivation of $\Delta_{\mathcal{T}}$. Although it supports automation, the procedure suffers from a high computationally cost. Then we present the enhanced procedure, that by taking advantage of our aggregation technique, eliminates the shortcomings of the first one and allows us to automatically and efficiently obtain $\Delta_{\mathcal{T}}$.

Algorithm 1 Given a model $\mathbb{M}$ and Partition $\Delta_{\mathcal{T}}$, return approximate distribution Π.

function TSND_SOLUTION(model $\mathbb{M}$, partition $\Delta_{\mathcal{T}}$)
  $init\_block \leftarrow$ // using the initial state of $\mathbb{M}$, derive the initial block;
  $explored \leftarrow \{\}$ // the set of blocks that have been explored ;
  $Q_{block} = [q_{IJ}]$ // the transition matrix capturing block level transitions;
  $agenda = \{init\_block\}$ // the set of blocks to be explored;
  // analysing internal dynamics of each block
  while $(agenda \neq \emptyset)$ **do**
    $B_I \leftarrow head(agenda)$; // data structure representing a block
    $D_I \leftarrow$ the state space of B_I derived by transitions of type $\vec{\mathcal{A}}^*_{fast}$;
    $\boldsymbol{p}_I \leftarrow$ analyse D_I and derive the steady state distribution of B_I;
    $newBlocks = \leftarrow$ derive the set of blocks that are reachable by
      $\vec{\mathcal{A}}^*_{slow}$ transitions;
    // exploring new blocks
    for $(newBlock \in newBlocks \;\&\; newBlock \notin explored)$ **do**
      if $(newBlock \notin agenda)$ **then**
        $agenda \leftarrow agenda + newBlock$;
      end if
    end for
    $Q_{block} \leftarrow$ set q_{IJ} for all $J \in (agenda \cup explored)$ using Eq.(5.17);
    $explored \leftarrow explored + B_I$;
    $agenda \leftarrow agenda - B_I$;
  end while
  $\theta \leftarrow$ derive solution of the block level CTMC using Q_{block};
  // Re-composition Step
  for $(B_I \in explored)$ **do**
    $\Pi_i \leftarrow$ Use Eq.(5.16), θ and $\boldsymbol{p}_I$ to derive the steady state associated with B_I:
    $\Pi \leftarrow$ update Π by appending the distribution Π_i to Π ;
  end for
  return Π ;
end function

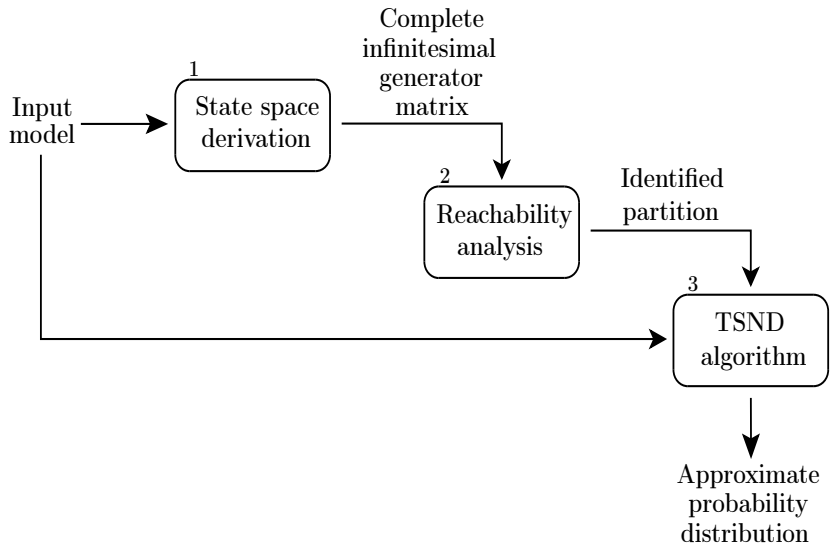


Figure 5.2: Steps of the naive application of TSND on a LSRB models.

5.3.1 Naive Procedure

The naive procedure follows a simple path illustrated in Fig. 5.2. Given a model, first we constructs its complete state space. This is then analysed by a reachability analysis that has the goal of detecting the decomposable blocks based on a given threshold that represents slow rates. Using the blocks that are identified, we infer the sets $\vec{\mathcal{A}}^*_{slow}$ and $\vec{\mathcal{A}}^*_{fast}$; the actions enabling the cross-block transitions belong to $\vec{\mathcal{A}}^*_{slow}$ and those enabling intra-block transitions are allocated to $\vec{\mathcal{A}}^*_{fast}$. The partition $\Delta_{\mathcal{T}} = \{\vec{\mathcal{A}}^*_{slow}, \vec{\mathcal{A}}^*_{fast}\}$ is then plugged into Alg. 1.

A wide range of algorithms can be used in the reachability analysis, including those that work directly on rearranging the lines and columns of the underlying infinitesimal matrix [33, Chp.1] or those based on the idea of clustering [54]. In this thesis, we employ the method formally known as *agglomerative hierarchical clustering* (AHC) [12].

5.3.2 Agglomerative Hierarchical Clustering

The method of AHC has roots in machine learning and is particularly useful for partitioning discrete state spaces based on their structural properties. An AHC clustering task is formed by specifying two functions *metric* (mtr) and *linkage criterion* (lc) on a discrete state space D. The function $mtr : D \times D \to \mathbb{R}$ captures the logical distance associated with any two states in D. The function $lc : 2^D \times 2^D \to \mathbb{R}$ captures the distance between two subsets of D. Having specified the functions and a state space D, the

clustering follows this procedure. Initially, one small cluster is constructed for each of the states in D (for n states, n clusters are formed). Then, the clusters are progressively examined and merged in a bottom-up manner; when two clusters are merged, a higher level parent cluster is agglomerated with an attribute that captures the distance between its children. As the procedure proceeds, increasingly larger clusters are generated and in each one the distance between children is derived using mtr and lc.

The iterations continue until no new cluster can be formed. The output is a binary tree that contains clusters of larger sizes (accommodating longer distances) as we move towards the root. To obtain the set of blocks, a *distance threshold* is specified, which defines a horizontal line in the tree. By construction, children of the clusters that reside above the line have distances larger than the threshold, and children of the clusters below the line have the distances less than the threshold. The state space is divided into blocks by cutting the tree at the level of the clusters immediately above the line.

The AHC can be used for deriving the partition $\Delta_{\mathcal{T}}$ in the following way. Consider a model $\mathbb{M}$ with infinitesimal generator matrix $Q = [q_{ij}]_{i,j\in D}$. The required ingredients are:

1. Function mtr. For any two states i and j we define the function mtr to represent the average time it takes to move from i to j via a one-step transition. Thus, $mtr(i,j) = \frac{1}{q_{i,j}}$.

2. Function lc. The linkage criterion between two sets of states D_1 and D_2 is defined to be the minimum of the distances across the members of D_1 and D_2: $lc(D_i, D_j) = \min_{i\in D_1, j\in D_2} mtr(i,j)$. This is known as the minimum linkage.

3. Distance threshold. We specify max_dis to capture the average time associated with actions happening on the slow time scale. For two sets of states D_1 and D_2, if $lc(D_1, D_2) \leq max\_dis$, then the sets are connected by transitions occurring on the fast time scale. Conversely, if $lc(D_1, D_2) \geq max\_dis$, then the sets are connected by slow transitions.

The output of the clustering method is a partition on D in which all blocks are separated by at least max_dis time units. In the final step, we check whether the rates of the transitions within the blocks are at least an order of magnitude larger than max_dis. If true, then the set of blocks satisfy near-complete decomposability and the partition $\Delta_{\mathcal{T}}$ is derived by examining the cross-block and intra-block transitions. In the negative

case, a higher value of *max_dis* is specified and a new clustering task is formulated. We assume that the modeller handles the task of providing *max_dis*.

It is important to comment on the computational cost of running the naive procedure. The complexity of running one clustering task is $O(n^3)$, where n is the size of the state space [34]. This means that the procedure becomes computationally expensive for large-scale models. Therefore, we developed the enhanced procedure that finds $\Delta_{\mathcal{T}}$ with more efficiency.

5.3.3 Enhanced Procedure

The enhanced approach is based on the fact that in LSRB models the rate observed for each action depends strongly on the sizes of the groups that enable the action. Recall that the groups are divided based on their sizes, $\mathcal{G}(\mathbb{M}) = \{\mathcal{G}_s(\mathbb{M}), \mathcal{G}_l(\mathbb{M})\}$, and actions are divided as $\overrightarrow{\mathcal{A}^*}(\mathbb{M}) = \{\overrightarrow{\mathcal{A}^*_s}(\mathbb{M}), \overrightarrow{\mathcal{A}^*_{sl}}(\mathbb{M}), \overrightarrow{\mathcal{A}^*_l}(\mathbb{M})\}$. Assuming that actions are to be divided into slow and fast categories, we make the following observations when a LSRB model is running:

1. The actions in $\overrightarrow{\mathcal{A}^*_l}(\mathbb{M})$ are enabled at high rates and their associated transitions occur continually with high frequencies. The rate that an individual instance in $\mathcal{G}_l(\mathbb{M})$ offers for an $\alpha \in \overrightarrow{\mathcal{A}^*_l}(\mathbb{M})$ might not be high. However, when individuals are replicated and form large populations, the rates are accumulated and consequently, the action gets frequently performed. The dense occurrence of $\overrightarrow{\mathcal{A}^*_l}(\mathbb{M})$ actions means that they belong to dynamics happening on the fast time scale.

2. There exist a subset of actions in $\overrightarrow{\mathcal{A}^*_s}(\mathbb{M}) \cup \overrightarrow{\mathcal{A}^*_{sl}}(\mathbb{M})$ that also belong to the fast time scale. These actions occur frequently due to the components in $\mathcal{G}_s(\mathbb{M})$ that in spite of being part of small populations, still manage to individually enable the actions at high rates. Note that this is in contrast with the previous case where the high frequencies were due the presence of large supportive populations.

3. There remains a subset of $\overrightarrow{\mathcal{A}^*_s}(\mathbb{M}) \cup \overrightarrow{\mathcal{A}^*_{sl}}(\mathbb{M})$ that are neither supported by large populations, nor enabled at high rates by instances in $\mathcal{G}_s(\mathbb{M})$. These actions occur considerably less frequently and give rise to the dynamics happening on the slow time scale.

These observations suggest that when categorising a model's actions, the set of fast actions $\overrightarrow{\mathcal{A}^*}_{fast}$ can be considered to consist of $\overrightarrow{\mathcal{A}^*_l}(\mathbb{M})$ and a subset of $\overrightarrow{\mathcal{A}^*_s}(\mathbb{M}) \cup \overrightarrow{\mathcal{A}^*_{sl}}(\mathbb{M})$

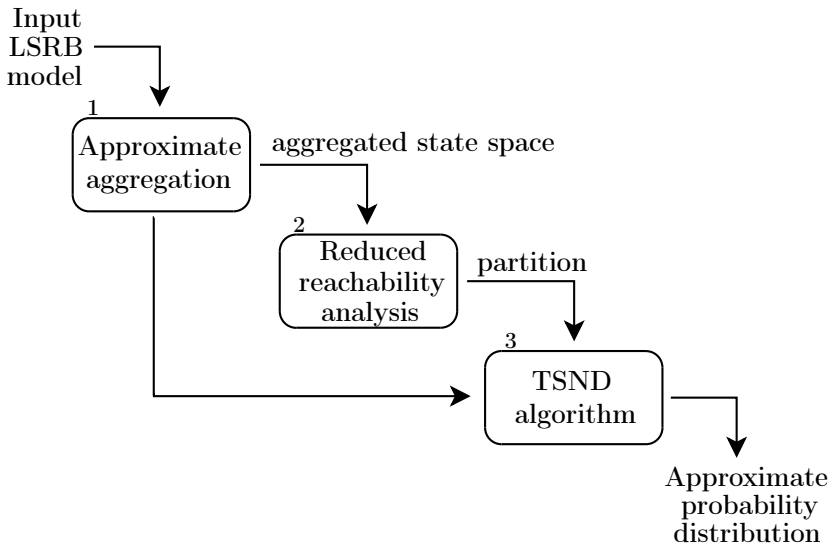


Figure 5.3: Steps of the smarter and more efficient application of TSND on a LSRB PEPA model.

related to the second case. Furthermore, the set of slow actions $\vec{\mathcal{A}}^*_{slow}$ can be regarded as what remains from $\vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M})$; i.e. $\vec{\mathcal{A}}^*_{slow} = (\vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M})) - \vec{\mathcal{A}}^*_{fast}$. Thus, assuming that $\vec{\mathcal{A}}^*_l(\mathbb{M}) \subset \vec{\mathcal{A}}^*_{fast}$ the problem of forming the partition $\Delta_{\mathcal{T}}$ over $\vec{\mathcal{A}}^*(\mathbb{M})$ can be reduced to that of finding the slow / fast partition over $\vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M})$. Now recall that the evolution of the model by $\vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M})$ actions is captured by its aggregated model $\mathbb{M}^{agg}$. Hence, finding $\Delta_{\mathcal{T}}$ can be done by a reachability analysis of D^{agg}, which is significantly smaller than the complete state space D. This is the basis of our enhanced method.

The steps of the enhanced method are shown in Fig. 5.3. Given a model $\mathbb{M}$, first the aggregation step is performed. The results is $\mathbb{M}^{agg}$ and its state space D^{agg}, which captures the evolution of $\mathbb{M}$ in terms of its sub-chains and $\vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M})$ transitions. The second step has two parts. First, a reachability analysis is performed where our clustering algorithm is applied to D^{agg}. Since D^{agg} is relatively small the clustering algorithm efficiently generates the blocks, and provides partition $\Delta^{agg}_{\mathcal{T}} = \{\vec{\mathcal{A}}^*_{agg,slow}, \vec{\mathcal{A}}^*_{agg,fast}\}$ over $\vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M})$. Second, we expand $\Delta^{agg}_{\mathcal{T}}$ to the level of the original model. Using $\Delta^{agg}_{\mathcal{T}}$, we construct the partition $\Delta_{\mathcal{T}} = \{\vec{\mathcal{A}}^*_{slow}, \vec{\mathcal{A}}^*_{fast}\}$ as:

$$\begin{aligned} \vec{\mathcal{A}}^*_{slow} &= \vec{\mathcal{A}}^*_{agg,slow} \\ \vec{\mathcal{A}}^*_{fast} &= \vec{\mathcal{A}}^*_{agg,fast} \cup \vec{\mathcal{A}}^*_l(\mathbb{M}) \end{aligned} \tag{5.18}$$

Finally, partition $\Delta_{\mathcal{T}}$ and the model are fed into the TSND solution algorithm.

Model	State Space	Process	Infinitesimal matrix	Uniformised process	Stochastic matrix	Set of blocks	A block
$\mathbb{M}$	D	X^C	Q	$X^{\mathcal{D}}$	R	B	B_I
$\mathbb{M}^{agg}$	D^{agg}	X^C_{agg}	Q_{agg}	$X^{\mathcal{D}}_{agg}$	R_{agg}	B_{agg}	$B_{agg,I}$

Table 5.1: Notation used for construction of blocks in the complete and aggregated state spaces.

Using the enhanced approach, we are essentially detecting the blocks within the aggregated state space, and then extending their validity to the complete state space. Such an expansion can be formally justified. In the next section, we present a theorem that supports our enhanced procedure.

5.3.4 Enhanced Procedure - Supporting Theorem

For convenience, the notation used in the theorem is summarised in Table 5.1. We assume that D^{agg} respects NCD, and consists of N_B blocks. This is specified as $B_{agg} = \{B_{agg,I}\}_{I\in[1\cdots N_B]}$ where each $B_{agg,I}$ consists of a number of aggregated states $\boldsymbol{\gamma} \in D^{agg}$. One other key assumption we make is that when the model is running in its stationary phase the transitions related to large groups happen on the fast time scale, i.e. transitions of types $\alpha \in \vec{\mathcal{A}}^*_l(\mathbb{M})$ occur at high rates.

Theorem 5.3.1. (Expanding NCD from $\mathbb{M}^{agg}$ to $\mathbb{M}$) *Consider the LSRB model $\mathbb{M}$ and its aggregated model $\mathbb{M}^{agg}$. If D^{agg} respects NCD with maximum probabilistic coupling ε_{agg} and the transitions of type $\vec{\mathcal{A}}^*_l(\mathbb{M})$ occur with rates at least one order of magnitude larger than ε_{agg}, then the original state space D is decomposable with the same coupling measure ε_{agg}. Furthermore, D^{agg} and D have a similar block structure, in the sense that:*

$$\forall\ B_{agg,I} \in B_{agg}\ \ \exists\ B_I \in B\ \ :\ \ B_I = \{\, S \in \mathbb{Y}_{\boldsymbol{\gamma}} \mid \mathbb{Y}_{\boldsymbol{\gamma}} \subset D\ ,\ \boldsymbol{\gamma} \in B^{agg}_I \} \tag{5.19}$$

That is, for each block $B_{agg,I}$ in D^{agg}, we have one corresponding block B_I in D that consists of states that constitute the sub-chains $\mathbb{Y}_{\boldsymbol{\gamma}} : \boldsymbol{\gamma} \in B^{agg}_I$.

Proof. The proof is by construction and its outline is shown in Fig. 5.4. We start with model $\mathbb{M}^{agg}$. Since it satisfies the decomposability condition, we can construct the

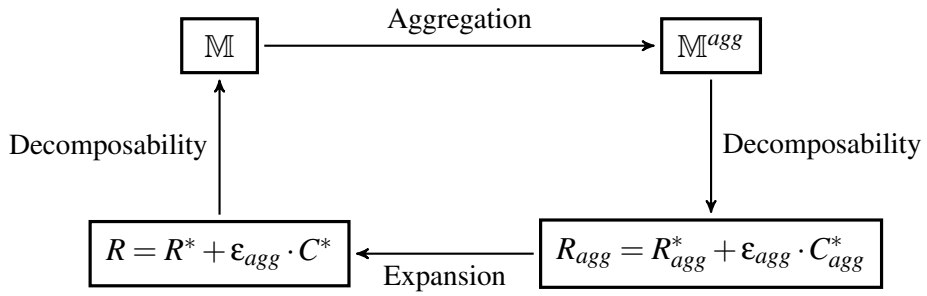


Figure 5.4: Outline of the proof for extending decomposability from D^{agg} to D.

equation:

$$R_{agg} = R^*_{agg} + \varepsilon_{agg} \cdot C^*_{agg} \tag{5.20}$$

where the elements R^*_{agg}, ε_{agg} and C^*_{agg} respect the structural conditions enumerated on Page 83 and are derived using the solution construction method of Sec. 5.2.2.

Next, we focus on $R = [r_{ij}]_{i,j \in D}$ associated with $\mathbb{M}$. Using the matrix R and the solution of Eq.(5.20) we *construct* equation:

$$R = R^* + \varepsilon_{agg} \cdot C \tag{5.21}$$

where $R^* = [r^*_{ij}]_{i,j \in D}$ and $C = [c_{ij}]_{i,j \in D}$ are respectively derived using Eq.(5.22) and Eq.(5.23):

$$r^*_{ij} = \begin{cases} r_{ij} & \text{if } i \in \mathbb{Y}_{\boldsymbol{\gamma}}, j \in \mathbb{Y}_{\boldsymbol{\gamma}} \\ & \quad \text{(a transition related to the} \\ & \quad \text{dynamics of the large populations)} \\ 0 & \text{if } i \in \mathbb{Y}_{\boldsymbol{\gamma}}, j \in \mathbb{Y}_{\boldsymbol{\beta}}, \mathbb{Y}_{\boldsymbol{\gamma}} \neq \mathbb{Y}_{\boldsymbol{\beta}}, \boldsymbol{\gamma} \in B_{agg,I}, \boldsymbol{\beta} \in B_{agg,J}, B_{agg,I} \neq B_{agg,J} \\ & \quad \text{(a cross sub-chain transition which is regarded as} \\ & \quad \text{cross-block transition in } D^{agg}. \\ r_{ij} + \dfrac{r_{ij}}{\left(\sum\limits_{\boldsymbol{\gamma} \in B_{agg,I}} \sum\limits_{h \in \mathbb{Y}_{\boldsymbol{\gamma}}} r_{ih} \right)} \times \left(\sum\limits_{\substack{B_{agg,J} \in B_{agg} \\ B_{agg,J} \neq B_{agg,I}}} \sum\limits_{\boldsymbol{\omega} \in B_{agg,J}} \sum\limits_{k \in \mathbb{Y}_{\boldsymbol{\omega}}} r_{ik} \right) & \\ \qquad \text{if } i \in \mathbb{Y}_{\boldsymbol{\gamma}}, \ j \in \mathbb{Y}_{\boldsymbol{\beta}}, \ \mathbb{Y}_{\boldsymbol{\gamma}} \neq \mathbb{Y}_{\boldsymbol{\beta}}, \ \boldsymbol{\gamma} \in B_{agg,I}, \ \boldsymbol{\beta} \in B_{agg,I} & \\ \qquad \text{(a transition from state } i \text{ to } j \text{ which is in a} & \\ \qquad \text{different sub-chain but in the same block in } D^{agg}) & \end{cases} \tag{5.22}$$

$$c_{ij} = \begin{cases} 0 & \text{if } i \in \mathbb{Y}_{\boldsymbol{\gamma}}, j \in \mathbb{Y}_{\boldsymbol{\gamma}} \\ \dfrac{1}{\varepsilon^D} \times \dfrac{r_{ij}}{\left(\sum\limits_{\substack{B_{agg,J} \in B_{agg} \\ B_{agg,J} \neq B_{agg,I}}} \sum\limits_{\boldsymbol{\omega} \in B_{agg,J}} \sum\limits_{k \in \mathbb{Y}_{\boldsymbol{\omega}}} r_{ik} \right)} & \\ \quad \text{if } i \in \mathbb{Y}_{\boldsymbol{\gamma}}, j \in \mathbb{Y}_{\boldsymbol{\beta}}, \mathbb{Y}_{\boldsymbol{\gamma}} \neq \mathbb{Y}_{\boldsymbol{\beta}}, \boldsymbol{\gamma} \in B_{agg,I}, \boldsymbol{\beta} \in B_{agg,J}, & \\ \quad B_{agg,I} \neq B_{agg,J} & \\ -\dfrac{1}{\varepsilon^D} \times \dfrac{r_{ij}}{\left(\sum\limits_{\boldsymbol{\gamma} \in B_{agg,I}} \sum\limits_{h \in \mathbb{Y}_{\boldsymbol{\gamma}}} r_{ih} \right)} \times \left(\sum\limits_{\substack{B_{agg,J} \in B_{agg} \\ B_{agg,J} \neq B_{agg,I}}} \sum\limits_{\boldsymbol{\omega} \in B_{agg,J}} \sum\limits_{k \in \mathbb{Y}_{\boldsymbol{\omega}}} r_{ik} \right) & \\ \quad \text{if } i \in \mathbb{Y}_{\boldsymbol{\gamma}}, j \in \mathbb{Y}_{\boldsymbol{\beta}}, \mathbb{Y}_{\boldsymbol{\gamma}} \neq \mathbb{Y}_{\boldsymbol{\beta}}, \boldsymbol{\gamma} \in B_{agg,I}, \boldsymbol{\beta} \in B_{agg,I} & \end{cases} \tag{5.23}$$

Our construction guarantees that:

1. R^* is a block diagonal matrix with stochastic blocks on the diagonal.
2. Each matrix block $R^*_{agg,I}$ associated with $B_{agg,I}$ is mapped to one matrix block R^*_I in R^*.
3. The elements of R^* are significantly larger than ε_{agg}.

The reformulation of R in terms of Eq.(5.21) means that $X^{\mathcal{D}}$ is near-completely decomposable with the same coupling measure ε_{agg}. Furthermore, by construction the property of Eq.(5.19) is respected.

Note that we presented the proof in terms of the model's underlying DTMC. The proof can be readily extended to the case of CTMCs using Eq.(5.14). □

Although seemingly complex, the calculations in Eq.(5.22) and Eq.(5.23) follow a simple logic. To assist with their understanding, we present the following example.

5.3.5 Extending Decomposability - Example

Consider Fig. 5.5, the complete state space of the client-server model initialised with one server, one client and parameters $r_s = 10$, $r_l = 5$, $r_b = 0.1$, $r_f = 0.2$, $r_c = \top$, $r_t = 3$. Note that the states are numbered. The underlying DTMC of this model has the

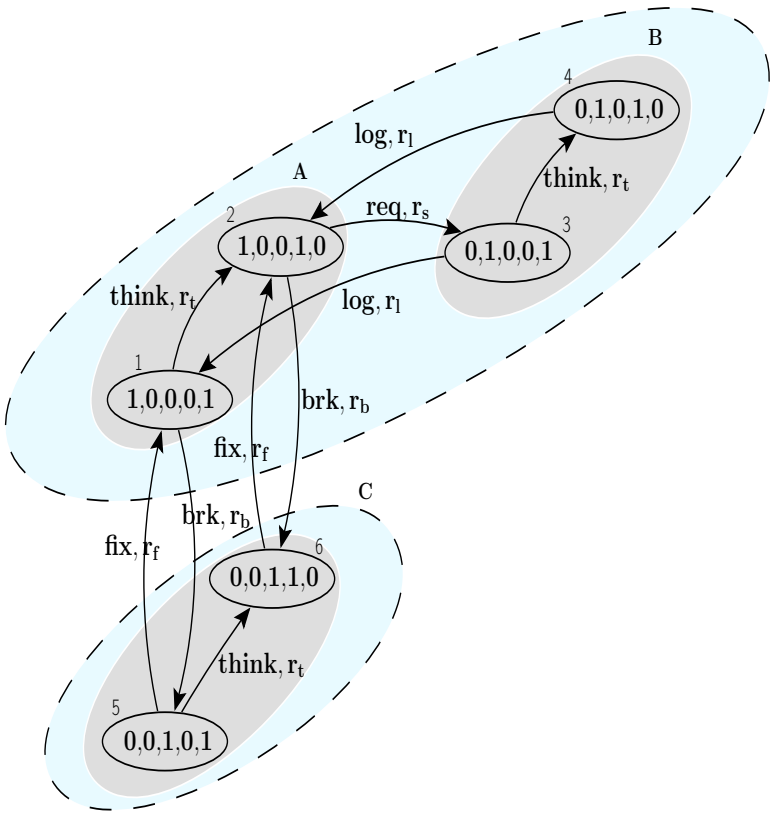


Figure 5.5: The complete state space of a client-server system initialised with one client and one server.

stochastic matrix shown on the left hand side of Eq.(5.25). The model is amenable to the aggregation and as a result we derive the aggregated state space shown in Fig. 5.1. The sub-chains are named in Fig. 5.5 by letters A, B and C. Note the correspondence between these labels and those for aggregated states in Fig. 5.1.

We showed on Page 87 that the DTMC of the aggregated model is decomposable:

$$R_{agg} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{bmatrix} 0 & \frac{10}{10.1} & \frac{0.1}{10.1} \\ \frac{5}{10.1} & \frac{5.1}{10.1} & 0 \\ \frac{0.2}{10.1} & 0 & \frac{9.9}{10.1} \end{bmatrix} \end{array} = \left[\begin{array}{cc|c} 0 & 1 & 0 \\ \frac{5}{10.1} & \frac{5.1}{10.1} & 0 \\ \hline 0 & 0 & 1 \end{array}\right]$$

$$+ \frac{0.2}{10.1} \begin{bmatrix} 0 & -\frac{0.1}{0.2} & \frac{0.1}{0.2} \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} \quad (5.24)$$

The set of blocks identified on D^{agg} is: $B_{agg} = \{B_{agg,1} = \{A,B\}, B_{agg,2} = \{C\}\}$. This means that transitions of type req, log are fast and the transitions of type $break, fix$ are slow. The maximum coupling factor is $\frac{0.2}{10.1}$ associated with the probability of *break* happening in state A.

Assuming that the rates of the transitions related to the client (the transitions of type *think*) are greater than the coupling factor, we use the blocks $B_{agg,1}$ and $B_{agg,2}$, and Eq.(5.22) and Eq.(5.23) to construct the decomposability equation of R as shown in Eq.(5.25). Here the transitions of type *think* are used to expand the blocks.

$$
\begin{array}{c}
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}
\begin{array}{c}
\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\left[\begin{array}{cccccc}
\frac{7}{10.1} & \frac{3}{10.1} & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{10}{10.1} & 0 & \frac{0.1}{10.1} & 0 \\
\frac{5}{10.1} & 0 & \frac{2.1}{10.1} & \frac{3}{10.1} & 0 & 0 \\
0 & \frac{5}{10.1} & 0 & \frac{5.1}{10.1} & 0 & 0 \\
\frac{0.2}{10.1} & 0 & 0 & 0 & \frac{6.9}{10.1} & \frac{3}{10.1} \\
0 & \frac{0.2}{10.1} & 0 & 0 & 0 & \frac{9.9}{10.1}
\end{array}\right]
\end{array}
\end{array}
=
\left[\begin{array}{cccccc|cccccc}
\frac{7}{10.1}+\frac{7}{(7+3)}\frac{0.1}{10.1} & \frac{3}{10.1}+\frac{3}{(7+3)}\frac{0.1}{10.1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{10}{10.1}+\frac{0.1}{10.1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{5}{10.1} & 0 & \frac{2.1}{10.1} & \frac{3}{10.1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{5}{10.1} & 0 & \frac{5.1}{10.1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & \frac{6.9}{10.1}+\frac{6.9}{(6.9+3)}\frac{0.2}{10.1} & \frac{3}{10.1}+\frac{3}{(3+6.9)}\frac{0.2}{10.1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{9.9}{10.1}+\frac{0.2}{10.1}
\end{array}\right]
$$

$$
+\frac{(0.2)}{10.1}
\left[\begin{array}{cccccc|cccccc}
-\frac{7}{(7+3)}\frac{0.1}{0.2} & -\frac{3}{7+3}\frac{0.1}{0.2} & 0 & 0 & 0 & 0 & \frac{0.1}{0.2} & 0 \\
0 & 0 & -\frac{0.1}{0.2} & 0 & 0 & 0 & 0 & \frac{0.1}{0.2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & -\frac{6.9}{(6.9+3)} & -\frac{3}{(6.9+3)} \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & -1
\end{array}\right]
\tag{5.25}
$$

	Version 1	Version 2	Version 3	Version 4
n_s	1	2	5	8
n_c	150	300	350	500

Table 5.2: Population parameters associated with the clients and servers.

	Time (Sec)		Memory (Byte)	
	Aggregated Model	Original Model	Aggregated Model	Original Model
Ver. 1	3	225	2336	788304
Ver. 2	5	596	4120	2281320
Ver. 3	8	1405	6448	4970784
Ver. 4	12	2833	9408	9239992

Table 5.3: Running time and memory used by clustering method when applied to four versions of the model. The measures are extracted from the Java virtual machine.

5.4 Enhanced Procedure - Experiment Results

5.4.1 Experiment Scenario

In this final section, we report the results of an experiment where we applied the enhanced procedure to four versions of our client-server model. The model is shown on Page. 44. The population parameters are shown in Table. 5.2. These are chosen in a way that the models have increasingly larger state spaces. The rate parameters are: $r_s = 10$, $r_l = 50$, $r_b = 0.01$, $r_f = 0.1$, $r_c = \top$, $r_t = 30$.

We implemented an agglomerative clustering tool using Java and applied it to our models and their aggregated versions. The coupling measure was set to $\varepsilon = 1$. We chose this as it corresponds to a value one order of magnitude smaller than the rate at which a client is served. We measured the run-time and memory usage of our clustering tool and report these measurements in Table 5.3. The evaluation of D and D^{agg} had the same result; it showed that transitions of type $\{req, log, think\}$ are fast and those of type $\{break, log\}$ are slow. However, the clustering through the aggregated state space achieved the result faster whilst consuming less memory, and showed that it is capable of being used in much larger models.

5.4.2 Final Remarks

We acknowledge that in this simple example, we could easily form partition $\Delta_{\mathcal{T}} = \{\vec{\mathcal{A}}^*_{fast} = \{req, log, tnk\}, \vec{\mathcal{A}}^*_{slow} = \{break, fix\}\}$ without going through the trouble of using the enhanced approach at all. Here, we could examine the behaviour of a single server and observe that *break* and fix happen at a much slower rate than the rest of the actions. Generalising this view, one might propose the conjecture that the infrequency of an action within one sequential component implies its occurrence over the slow time scale within the emergent behaviour. The conjecture might be true for specific models, but it turns out to be naive and generally wrong. As a counter-example, we considered a client-server system with the rate parameters as above and population parameters $n_s = 20$ and $n_c = 10000$. In this case, the group of servers is still small, but their populations has grown just enough that transitions of type fix happen at higher rates and potentially fill the gap between previously distinct time scales. Here, the model does not respect NCD, the partition $\Delta_{\mathcal{T}} = \{\vec{\mathcal{A}}^*_{fast} = \{req, log, tnk\}, \vec{\mathcal{A}}^*_{slow} = \{break, fix\}\}$ becomes invalid and applying the TSND solution results in an erroneous approximate solution. These all happen even though looking at one individual server still derives the said partition.

The observation above highlights one important aspect of our enhanced procedure. For a given model $\mathbb{M}$, the successful derivation of $\Delta_{\mathcal{T}}$ quantitatively justifies the presence of muti-scale dynamics and rigorously validates the use of the TSND method as an accurate analysis approach. However, if $\mathbb{M}$ is not decomposable, the method alerts that the TSND must not be applied. The efficiency of the method enables us to make the validity decision quickly and through the aggregated state space. In this approach, we are not limited to the behaviour of individual processes in small groups, but rather, we form our judgment more informatively by taking into account the collective dynamics of instances within the small groups.

5.4.3 Requirement Imposed on Actions of Large Groups

Our enhanced approach relies heavily on the assumption that the transitions related to large groups (resource users) occur at fast rates. One possible way to check this is to observe the behaviour of the real system under study. Perhaps we could use techniques based on machine learning to infer the distributions of the apparent rates of actions in $\vec{\mathcal{A}}^*_l(\mathbb{M})$ and compare them against the similar distributions derived for those

in $\vec{\mathcal{A}}^*_{slow}$. Alternatively, we may be able to use the model's expected behaviour (fluid flow moments and MCM conditional moments) to assert that $\alpha \in \vec{\mathcal{A}}^*_l(\mathbb{M})$ are always supported by large number of instances in $\mathcal{G}_l(\mathbb{M})$.

However, an interesting alternative is to verify the assumption syntactically by taking into account the model's structure and its parameters regarding actions' rates and populations. For instance, our preliminary work shows that it is possible to derive upper and lower bound constant for the rates of actions across the states where they are enabled. If, for an $\alpha \in \vec{\mathcal{A}}^*_l(M)$ the *lower* bound is *much larger* than the *highest* rate of the actions happening on the *slow* time scale, then one can safely assume that α belongs to the fast dynamics, as its rate only gets faster when enabled by multiple instances in large groups. Criteria such as this need to be developed to accompany our enhanced method. However, given the time constraint of this project, this aspect of our method could not be explored further and we leave the development of such criteria as a future task.

Chapter 6

Analysis of LSRB Models Using Conditional Expectations

6.1 Introduction

In this chapter, we present the method of conditional expectations, the first part of the method of conditional moments (MCM). Similarly to the near-complete decomposition, the aim of the MCM is to provide an evaluation tool for efficient and faithful analysis of LSRB models.

Given a model $\mathbb{M}$ with the set of groups $\mathcal{G}(\mathbb{M}) = \{\mathcal{G}_s(\mathbb{M}), \mathcal{G}_l(\mathbb{M})\}$ and the state vector $\boldsymbol{\xi} = \langle \boldsymbol{\xi}^s, \boldsymbol{\xi}^l \rangle$ we observe that the small groups experience a relatively small number of configurations (limited number of sub-chains, aggregated states) and the main reason causing the problem of state space explosion is the very large number of states related to the transitions of large groups within each sub-chain. Thus, to achieve efficiency while keeping faithfulness, we study the behaviour of $\boldsymbol{\xi}^s$ in terms of the discrete aggregated state space and the evolution of $\boldsymbol{\xi}^l$ by using the moments of the conditional distributions $\mathbb{P}(\boldsymbol{\xi}^l \mid \boldsymbol{\xi}^s)$. Assuming that $\boldsymbol{\xi}^s$ represents the system's resources and $\boldsymbol{\xi}^l$ the resource users, the MCM guarantees that the correlation between the internal dynamics of resources and the behaviour experienced by users is preserved. Using MCM, the construction of the internal states of the sub-chains is completely avoided. Thus, the method can be used for models which, even after applying the decomposability, give rise to blocks that are too large to be efficiently analysed.

As a preliminary example, consider the client-server system where five servers co-

operate with 100 clients on the *req* action. The aggregated state space of this system D_{CS}^{agg} is analysed to derive $\mathbb{P}_t(\langle S_i, S_l, S_b \rangle)$ for all $\langle S_i, S_l, S_b \rangle \in D_{CS}^{agg}$. In this system, the method of conditional moments allows us to derive the conditional moments of $\langle C_t, C_r \rangle$ such as the conditional expectations $\mathbb{E}_t [\langle C_t, C_r \rangle \mid \langle S_i, S_l, S_b \rangle]$ and conditional variances $\mathbb{VAR}_t [\langle C_t, C_r \rangle \mid \langle S_i, S_l, S_b \rangle]$, given the different configurations of the servers. These moments are useful for studying the stochastic behaviour of the clients given the different performance profiles associated with the servers' distinct modes of operation. We will describe the details of this example in Chap. 7.

In this chapter, we focus on the derivation of the first-order conditional moments, i.e. the conditional expectations. In Chap. 8, we describe the derivation of the higher-order conditional moments and co-moments.

The calculation of the conditional expectation is possible through the derivation of a system of differential algebraic equations (DAEs) from the model. In Sec.6.2, we formally introduce the conditional expectations and describe how the DAEs are derived. The equations are regarded as an initial value problem and can be solved when supplied with the required initial values. Finding such values is not trivial for DAEs underlying LSRB models. In Sec. 6.4, we present an algorithm for deriving the initial values.

The content presented in this chapter has not been published yet and we are in the process of preparing the related paper. The content of this chapter will be combined and integrated with those presented in chapters 5, 7, 8 and 9 and the paper will be submitted to a suitable journal.

6.2 Definitions

First, let us formally define the notion of conditional expectation related to the stochastic behaviour of the large groups. Consider the sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$ consisting of states $S_i = \langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l \rangle$ where $\boldsymbol{\xi}_i^s = \boldsymbol{\gamma}$. The probability of being in each S_i at time t is denoted by $\mathbb{P}_t(S_i) = \mathbb{P}_t(\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle)$. The probability of being in the sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$ at time t is denoted by $\mathbb{P}_t(\boldsymbol{\gamma})$ and is derived by:

$$\mathbb{P}_t(\boldsymbol{\gamma}) = \sum_{\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in D} \mathbb{P}_t(\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle) \tag{6.1}$$

Definition 28. (Conditional expectation of the vector $\boldsymbol{\xi}^l$ given a configuration $\boldsymbol{\gamma}$ for $\boldsymbol{\xi}^s$) *The* conditional *expectation of the vector of random variables $\boldsymbol{\xi}^l$ at time t, given that at t the state of the small groups is $\boldsymbol{\gamma}$, is denoted by $\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$ and is defined as:*

$$\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right] = \sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l\rangle \in Y_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \mathbb{P}_t(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}) = \sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l\rangle \in Y_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \frac{\mathbb{P}_t(\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l\rangle)}{\mathbb{P}_t(\boldsymbol{\gamma})} \tag{6.2}$$

In this definition, we use Bayes' Law [15, Chp. 1] to expand the conditional probability $\mathbb{P}_t(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma})$. According to this law, for two probabilistic events ε_1 and ε_2 with non-zero probabilities:

$$\mathbb{P}_t(\varepsilon_2 \mid \varepsilon_1) = \frac{\mathbb{P}_t(\varepsilon_1, \varepsilon_2)}{\mathbb{P}_t(\varepsilon_1)} \tag{6.3}$$

In Def. 28, we focused on the conditional expectation of the vector $\boldsymbol{\xi}^l$. A similar definition is described below, which concentrates on *reward functions* defined over $S_i = \langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l\rangle \in D$.

Let $T(\boldsymbol{\xi}^l, t) : \mathbb{N}^{n(\mathbb{M},l)} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a polynomial function defined over the Markov Chain of model $\mathbb{M}$[1]. This function represents a time dependent instantaneous reward associated with the states in D; at any point of time t, to any state $\langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l\rangle \in D$ the reward $T(\boldsymbol{\xi}_i^l, t)$ is assigned.

Definition 29. (Conditional expectation of a reward function) *At time t, the conditional expectation of a reward function $T(\boldsymbol{\xi}^l, t)$ given that at t, $\boldsymbol{\xi}^s = \boldsymbol{\gamma}$, is denoted by $\mathbb{E}_t\left[T(\boldsymbol{\xi}^l, t) \mid \boldsymbol{\gamma}\right]$ and is defined as:*

$$\mathbb{E}_t\left[T(\boldsymbol{\xi}^l, t) \mid \boldsymbol{\gamma}\right] = \sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l\rangle \in Y_{\boldsymbol{\gamma}}} T(\boldsymbol{\xi}_i^l, t) \cdot \mathbb{P}_t(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}) \tag{6.4}$$

where $\mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma})$ is derived using Eq.(6.3).

Using the definitions above, we are ready to describe the derivation of the conditional expectation of $\boldsymbol{\xi}^l$.

6.3 Conditional Expectations - Derivation of Equations

Given a model, the time evolution of the model's underlying probability distribution is derived by solving the associated system of Chapman-Kolmogorov (C-K) equations. In

[1] Recall that $n(\mathbb{M}, l)$ represents the number of state variables in $\boldsymbol{\xi}^l$.

this system, for each state $S_i \in D$ there is one ODE which captures how $\mathbb{P}_t(S_i)$ changes in an infinitesimal time interval as the result of probability fluxes into and out of S_i. We can start by considering the symbolic representation of the model's C-K equations and then derive the equations that describe the conditional expectations related to the model's large groups.

For each state S_i, one C-K equation is constructed of the form:

$$\forall\ S_i \in D\ :\ \frac{d\ \mathbb{P}_t(S_i)}{d\ t}\ =\ \underbrace{-\sum_{S_i \xrightarrow{(\alpha, r_\alpha(S_i))} S_j} r_\alpha(S_i)\cdot \mathbb{P}_t(S_i)}_{\text{outward transitions from } S_i} \underbrace{+\sum_{S_k \xrightarrow{(\alpha, r_\alpha(S_k))} S_i} r_\alpha(S_k)\cdot \mathbb{P}_t(S_k)}_{\text{inward transitions into } S_i} \quad (6.5)$$

where for each state $S_i = \langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l \rangle$, the first and second term on the right hand side describe respectively the probability fluxes out of and into S_i due to outward and inward transitions related to this state.

Let us assume that for each state $S_i = \langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l \rangle$, the two sides of Eq.(6.5) are multiplied by $\boldsymbol{\xi}_i^l$ to obtain:

$$\forall\ S_i \in D\ :\ \boldsymbol{\xi}_i^l \cdot \frac{d\ \mathbb{P}_t(S_i)}{d\ t}\ =\ -\ \boldsymbol{\xi}_i^l \cdot \sum_{S_i \xrightarrow{(\alpha, r_\alpha(S_i))} S_j} r_\alpha(S_i)\cdot\ \mathbb{P}_t(S_i) + \boldsymbol{\xi}_i^l \cdot \sum_{S_k \xrightarrow{(\alpha, r_\alpha(S_k))} S_i} r_\alpha(S_k)\cdot \mathbb{P}_t(S_k). \quad (6.6)$$

We focus on the sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$ and all states $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}$. Summing the right and left hand sides of Eq.(6.6) over $S_i \in Y_{\boldsymbol{\gamma}}$, we obtain:

$$\begin{aligned}\forall\ \boldsymbol{\gamma} \in D^{agg}\ :\ \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \frac{d\ \mathbb{P}_t(S_i)}{d\ t}\ =& \\ -\sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \sum_{S_i \xrightarrow{(\alpha, r_\alpha(S_i))} S_j} & r_\alpha(S_i)\cdot \mathbb{P}_t(S_i) \\ +\sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \sum_{S_k \xrightarrow{(\alpha, r_\alpha(S_k))} S_i} & r_\alpha(S_k)\cdot \mathbb{P}_t(S_k) \quad (6.7)\end{aligned}$$

First, we consider the left hand side of Eq.(6.7) and apply Bayes' Law:

$$\sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \frac{d\ \mathbb{P}_t(S_i)}{d\ t}$$

$$= \sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \frac{d\ \left(\mathbb{P}_t(\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle)\right)}{d\ t} = \sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \frac{d\ \left(\mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \cdot \mathbb{P}_t(\boldsymbol{\gamma})\right)}{d\ t}$$

$$= \frac{d}{d\ t}\left(\sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \cdot \mathbb{P}_t(\boldsymbol{\gamma})\right) = \frac{d}{d\ t}\left(\mathbb{P}_t(\boldsymbol{\gamma}) \sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma})\right)$$

$$= \frac{d}{d\ t}\left(\ \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\ \right) \tag{6.8}$$

We apply the product rule for differentiation to the last line of Eq.(6.8) to derive:

$$\sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \frac{d\ \mathbb{P}_t(S_i)}{d\ dt} = \frac{d\ \left(\mathbb{P}_t(\boldsymbol{\gamma})\right)}{d\ t} \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\ +\ \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \frac{d\ \left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\right)}{d\ t} \tag{6.9}$$

When we substitute Eq.(6.9) in the left hand side of Eq.(6.7) we can derive Eq.(6.10). Note that as a result of this transformation, the left hand side is now fully written in terms of the conditional expectations and the marginal distribution over D^{agg}.

$$\forall\, \boldsymbol{\gamma} \in D^{agg}\ :\ \frac{d\ \left(\mathbb{P}_t(\boldsymbol{\gamma})\right)}{d\ t} \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\ + \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \frac{d\ \left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\right)}{d\ t}\ =$$

$$\underbrace{-\sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \sum_{S_i \xrightarrow{(\alpha, r\alpha(S_i))} S_j} r_\alpha(S_i) \cdot \mathbb{P}_t(S_i)}_{\text{outward transitions}}$$

$$\underbrace{+\sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \sum_{S_k \xrightarrow{(\alpha, r\alpha(S_k))} S_i} r_\alpha(S_k) \cdot \mathbb{P}_t(S_k)}_{\text{inward transitions}} \tag{6.10}$$

Having transformed the left hand side, we now consider the right hand side with the aim of finding closed form expressions in terms of the conditional expectations and

the marginal probability distribution over D^{agg}. The derivations for the right hand side are lengthier, and for clarity a map of these transformations is shown in Fig. 6.1.

Each line on the right hand side of Eq.(6.10) has two summations; the outer over the states in $\mathbb{Y}_{\boldsymbol{\gamma}}$ and the inner over transitions. In each line, it is possible to move $\boldsymbol{\xi}_i^l$ into the inner summation. We apply this transformation to derive:

$$\forall\, \boldsymbol{\gamma} \in D^{agg} \;:\; \frac{d\left(\mathbb{P}_t(\boldsymbol{\gamma})\right)}{d\,t} \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right] \;+\; \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \frac{d\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\right)}{d\,t} \;=$$

$$\underbrace{-\sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l\rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \;\sum_{S_i \xrightarrow{(\alpha, r_\alpha(S_i))} S_j} \boldsymbol{\xi}_i^l \;\cdot\; r_\alpha(S_i) \cdot \mathbb{P}_t(S_i)}_{\text{outward transitions}}$$

$$\underbrace{+\sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l\rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \;\sum_{S_k \xrightarrow{(\alpha, r_\alpha(S_k))} S_i} \boldsymbol{\xi}_i^l \;\cdot\; r_\alpha(S_k) \cdot \mathbb{P}_t(S_k)}_{\text{inward transitions}} \tag{6.11}$$

In Eq.(6.11), consider the expression related to the outward and inward transitions. Each consists of two summations. These summations are independent and we can swap their ordering, similarly to the transformation we applied in our aggregation technique to Eq.(4.8). We transform Eq.(6.11) into the following:

$$\forall\; \boldsymbol{\gamma} \in D^{agg} \;:\; \frac{d\left(\mathbb{P}_t(\boldsymbol{\gamma})\right)}{d\,t} \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right] \;+\mathbb{P}_t(\boldsymbol{\gamma}) \cdot \frac{d\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\right)}{d\,t} \;=$$

$$\underbrace{-\sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \;\sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l\rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, S_i \geq \mathcal{V}_\alpha^-} \boldsymbol{\xi}_i^l \;\cdot\; r_\alpha(S_i) \cdot \mathbb{P}_t(S_i)}_{\text{outward transitions}}$$

$$\underbrace{+\sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \;\sum_{S_k \geq \mathcal{V}_\alpha^-,\, S_i=\langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l\rangle = S_k + \mathcal{V}_\alpha} \boldsymbol{\xi}_i^l \cdot\; r_\alpha(S_k) \cdot \mathbb{P}_t(S_k)}_{\text{inward transitions}} \tag{6.12}$$

In the next steps, we transform the expressions related to the outward and inward transitions separately and plug the results back into Eq.(6.12).

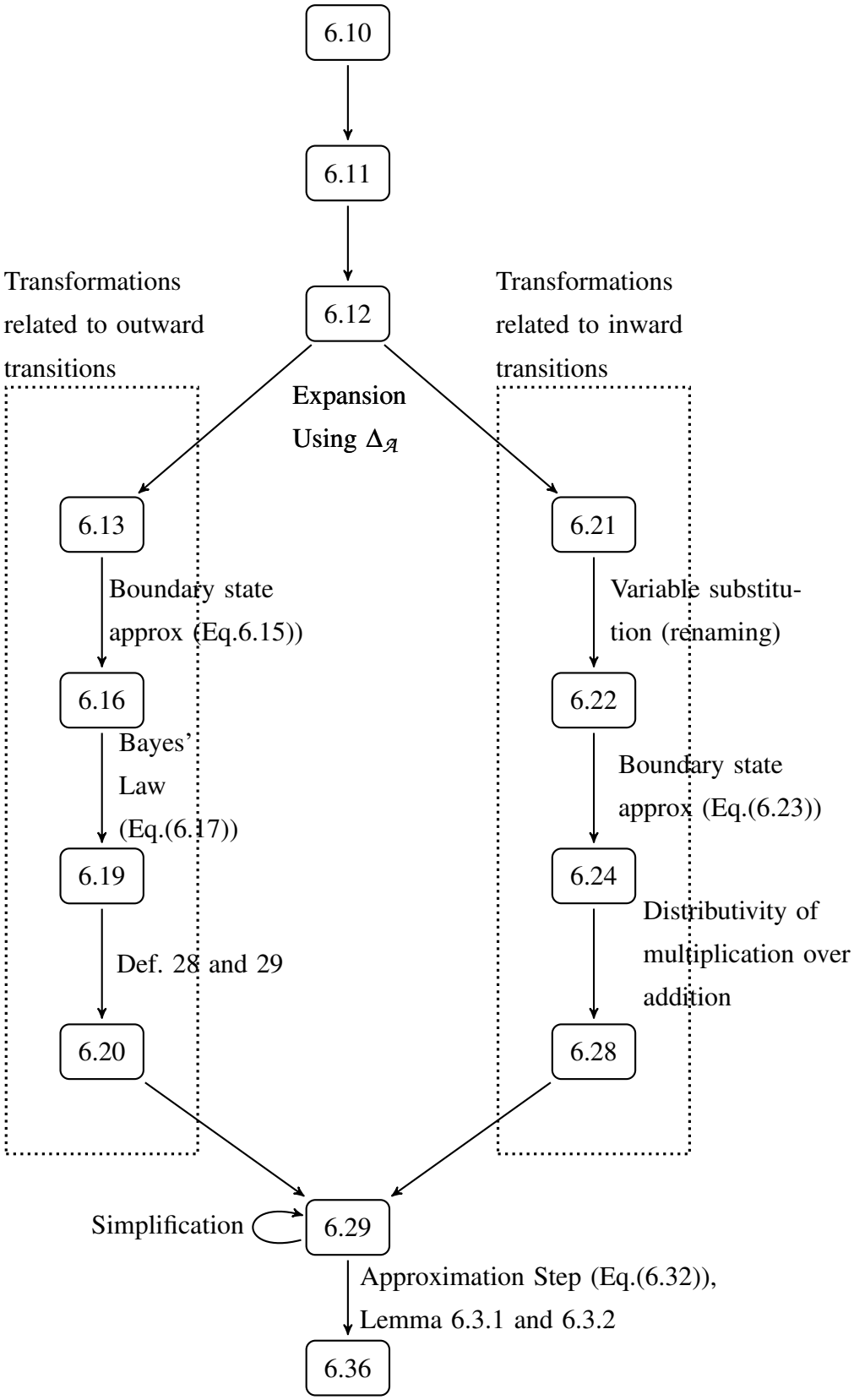


Figure 6.1: Transformations applied to the right hand side of Eq.(6.10). The boxes contain the numerical identifier of the equations which we progressively derive.

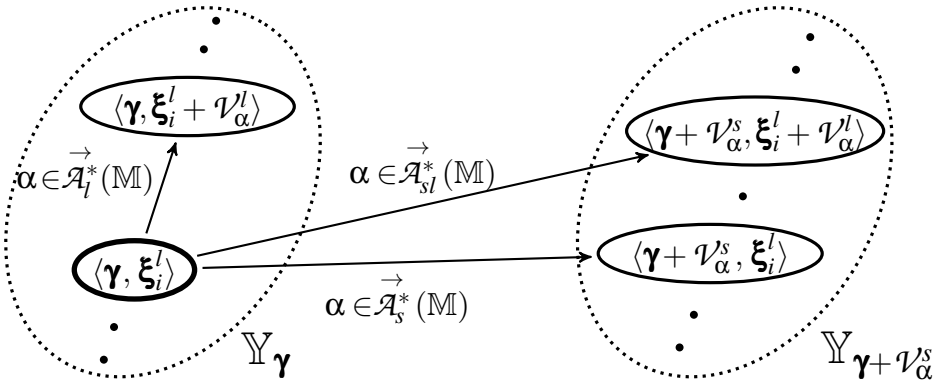


Figure 6.2: The impact of outward transitions with different types on the state vector.

6.3.1 Outward Transitions

First, we consider the expression related to the outward transitions. The impact of a transition of a type α is shown in Fig. 6.2., when $\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M})$, $\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})$ or $\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})$.

In this figure, consider the three transitions leaving the state $\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle$. If $\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M})$, then $\mathcal{V}_\alpha^s \neq 0$, $\mathcal{V}_\alpha^l = 0$, the transition only affects the configuration of the small groups and its rate is determined by this configuration: $r_\alpha(S_i) = r_\alpha(\boldsymbol{\gamma})$. For the target state $S_k = \langle \boldsymbol{\xi}_k^s, \boldsymbol{\xi}_k^l \rangle$ we have: $\boldsymbol{\xi}_k^s = \boldsymbol{\gamma} + \mathcal{V}_\alpha^s$ and $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l$. In the state space, this transition exists if $S_i \in D$ and $\boldsymbol{\gamma} > \mathcal{V}_\alpha^{s,-}$. As the second case, if $\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})$, then the transition changes the configurations of both small and large groups. In this case, $\boldsymbol{\xi}_k^s = \boldsymbol{\gamma} + \mathcal{V}_\alpha^s$ and $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l + \mathcal{V}_\alpha^l$. This transition exists if both $\boldsymbol{\xi}_i^s$ and $\boldsymbol{\xi}_i^l$ enable it, that is, when $S_i \in D$, $\boldsymbol{\xi}_i^s \geq \mathcal{V}_\alpha^{s,-}$ and $\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}$. Since the large groups are passive with respect to $\vec{\mathcal{A}}_{sl}^*(\mathbb{M})$ actions, $r_\alpha(S_i) = r_\alpha(\boldsymbol{\gamma})$. As the last case, if $\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})$, then $\mathcal{V}_\alpha^s = 0$ and $\mathcal{V}_\alpha^l \neq 0$. In this case, $\boldsymbol{\xi}_k^s = \boldsymbol{\xi}_i^s = \boldsymbol{\gamma}$, $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l + \mathcal{V}_\alpha^l$ and the rate of the transition is determined by $\boldsymbol{\xi}_i^l$. This transition exists if $S_i \in D$ and $\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}$. The properties of these transitions are summarised in Table.6.1.

In Eq.(6.12) and in the term related to the outward transitions, we can use the partition $\Delta_{\mathcal{A}} = \{\vec{\mathcal{A}}_s^*(\mathbb{M}), \vec{\mathcal{A}}_{sl}^*(\mathbb{M}), \vec{\mathcal{A}}_l^*(\mathbb{M})\}$ over $\vec{\mathcal{A}}^*(\mathbb{M})$ to expand the summations over $\vec{\mathcal{A}}^*(\mathbb{M})$ given the different categories in $\Delta_{\mathcal{A}}$:

	$S_i \xrightarrow{(\alpha,\cdot)} S_k \ , \ S_i = \langle \boldsymbol{\xi}_i^s , \boldsymbol{\xi}_i^l \rangle \ , \ S_k = \langle \boldsymbol{\xi}_k^s , \boldsymbol{\xi}_k^l \rangle$		
	$\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M})$	$\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})$	$\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})$
necessary condition	$\boldsymbol{\xi}_i^s \geq \mathcal{V}_\alpha^{s,-}$	$\boldsymbol{\xi}_i^s \geq \mathcal{V}_\alpha^{s,-}$ $\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}$	$\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}$
impact	$\boldsymbol{\xi}_k^s = \boldsymbol{\xi}_i^s + \mathcal{V}_\alpha^s$ $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l$	$\boldsymbol{\xi}_k^s = \boldsymbol{\xi}_i^s + \mathcal{V}_\alpha^s$ $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l + \mathcal{V}_\alpha^l$	$\boldsymbol{\xi}_k^s = \boldsymbol{\xi}_i^s$ $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l + \mathcal{V}_\alpha^l$
rate	$r_\alpha(\boldsymbol{\xi}_i^s)$	$r_\alpha(\boldsymbol{\xi}_i^s)$	$r_\alpha(\boldsymbol{\xi}_i^l)$

Table 6.1: The properties of the outward transitions from a state S_i

$$
\begin{aligned}
- \sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \ \cdot \ r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) \ = \\
- \sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,-}} \boldsymbol{\xi}_i^l \ \cdot \ r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(S_i) \\
- \sum_{\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} \sum_{\substack{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\xi}_i^s \geq \mathcal{V}_\alpha^{s,-} \\ \boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}}} \boldsymbol{\xi}_i^l \ \cdot \ r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(S_i) \qquad \star \\
- \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}} \boldsymbol{\xi}_i^l \cdot \ r_\alpha(\boldsymbol{\xi}_i^l) \cdot \mathbb{P}_t(S_i) \qquad (6.13)
\end{aligned}
$$

Boundary state approximation. In the next transformation, we focus on the expression in Eq.(6.13) related to $\vec{\mathcal{A}}_{sl}^*(\mathbb{M})$ action types (marked with a star). We assume that in any sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$ the probability of being in boundary states is close to zero. These are states $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle$ where there is at least one action type $\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})$ enabled by the small groups (i.e. $\boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,-}$), but not by the cooperating large groups (i.e. $\boldsymbol{\xi}_i^l \ngeq \mathcal{V}_\alpha^{l,-}$). As stated in Section 4.2.3.3, this assumption can be formally specified as:

$$
\begin{aligned}
\forall \, \alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \ , \quad \forall \, S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}} \quad : \\
\Big(\ (\boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,-}) \implies (\boldsymbol{\xi}_i^l \ngeq \mathcal{V}_\alpha^{l,-}) \ \Big) \implies \mathbb{P}_t(S_i) \approx 0 \qquad (6.14)
\end{aligned}
$$

For all states $S_i \in \mathbb{Y}_{\boldsymbol{\gamma}}$, the configuration of the small groups is identical. Therefore, for any $\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})$, if there exists any state $S_i \in \mathbb{Y}_\gamma$ which enables α at $r_\alpha(\boldsymbol{\gamma}) \neq 0$, then across all states in this sub-chain, if α is enabled, it would only be at the same rate $r_\alpha(\boldsymbol{\gamma})$. The inner summation in Eq.(6.14) sums over *only* the states $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}$

where both $\boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,-}$ and $\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}$ (both small and large groups enable the transition). Since the probability of being in boundary states ($\boldsymbol{\gamma} + \mathcal{V}_\alpha^s \geq 0$ and $\boldsymbol{\xi}_i^l + \mathcal{V}_\alpha^l \not\geq 0$) is close to zero we can drop the restriction over the configuration of the large groups and only focus only on the configuration of the small groups. Here, for the boundary states: $\boldsymbol{\xi}_i^l \cdot r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(S_i) \approx 0$, their contribution becomes negligible and therefore, such contributions can be safely added to the summation:

$$\sum_{\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} \sum_{\substack{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,-} \\ \underline{\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}}}} \xi_i^l \cdot r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(S_i) \approx$$

$$\sum_{\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,-}} \boldsymbol{\xi}_i^l \cdot r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(S_i) \quad (6.15)$$

By substituting Eq.(6.15) in Eq.(6.13) we obtain:

$$\begin{aligned} - \sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) \approx \\ - \sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,-}} \boldsymbol{\xi}_i^l \cdot r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(S_i) \\ - \sum_{\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,-}} \boldsymbol{\xi}_i^l \cdot r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(S_i) \\ - \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}} \boldsymbol{\xi}_i^l \cdot r_\alpha(\boldsymbol{\xi}_i^l) \cdot \mathbb{P}_t(S_i) \quad (6.16) \end{aligned}$$

For an action α and a state $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle$, if α is not enabled we have: $r_\alpha(S_i) = 0$; if $\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})$ then $\boldsymbol{\gamma} \not\geq \mathcal{V}_\alpha^{-,s}$ and $r_\alpha(\boldsymbol{\gamma}) = 0$, and if $\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})$ then $\boldsymbol{\xi}_i^l \not\geq \mathcal{V}_\alpha^{l,-}$ and $r_\alpha(\boldsymbol{\xi}_i^l) = 0$. In each line of Eq. 6.16, the inner summations are respectively over the states where the relevant actions are enabled. Since for each action type α the non-enabling states add zero to this expression, we can relax the constraint currently imposed on these states to include *all* states in each sub-chain; when $\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M})$, the constraint $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,-}$ is replaced by $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}$, a similar replacement is applied when $\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})$ and $\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})$; the constraint $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}$ is replaced by $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}$. We call this transformation *the inclusion of non-enabling states* which is also used in later transformations. The constraints on the summations are updated in the equations that follow.

Using Bayes' Law. Using the Bayes' Law, the joint probability $\mathbb{P}_t(S_i)$, $S_i = \langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l \rangle$,

can be written in the conditional form:

$$\mathbb{P}_t(S_i) = \mathbb{P}_t(\boldsymbol{\xi}_i^s) \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\xi}_i^s) \tag{6.17}$$

By substituting $\boldsymbol{\xi}_i^s = \boldsymbol{\gamma}$, we obtain:

$$\forall\, \boldsymbol{\gamma} \in D^{agg} \;:\; \mathbb{P}_t(\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle) = \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \tag{6.18}$$

Using Eq.(6.18), we transform Eq.(6.16) into the following:

$$\begin{aligned} &- \sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \; \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \;\cdot\; r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) \;\approx \\ &-\mathbb{P}_t(\boldsymbol{\gamma}) \cdot \Bigg[\sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma}) \;\cdot \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \xi_i^l \;\cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \\ &\qquad + \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \; \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \;\cdot r_\alpha(\boldsymbol{\xi}_i^l) \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \Bigg] \end{aligned} \tag{6.19}$$

Given Def.(28), we have:

$$\forall\, \boldsymbol{\gamma} \;:\; \mathbb{E}_t\Big[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\Big] = \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma})$$

and by Def.(29) we obtain:

$$\forall\, \boldsymbol{\gamma} \;:\; \mathbb{E}_t\Big[\boldsymbol{\xi}^l \cdot r_\alpha(\boldsymbol{\xi}_i^l) \mid \boldsymbol{\gamma}\Big] = \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot r_\alpha(\boldsymbol{\xi}_i^l) \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \;\;.$$

Thus, we transform Eq.(6.19) into:

$$\begin{aligned} &- \sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \; \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \;\cdot\; r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) \;\approx \\ &-\mathbb{P}_t(\boldsymbol{\gamma}) \;\cdot \Bigg[\sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma}) \;\cdot \mathbb{E}_t\Big[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\Big] \\ &\qquad + \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{E}_t\Big[\boldsymbol{\xi}^l \cdot r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\Big] \Bigg] \end{aligned} \tag{6.20}$$

Eq.(6.20) is substituted in Eq.(6.12) as the term related to the outward transitions.

Thus far, we have been considering the term related to the outward transitions. Now we consider the term in Eq.(6.12) related to the inward transitions.

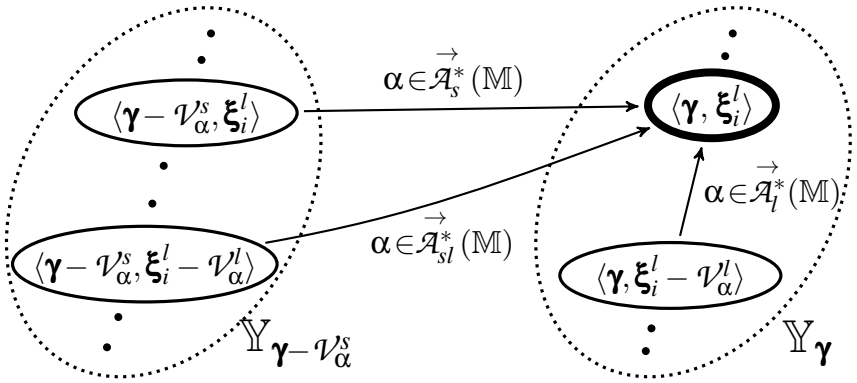


Figure 6.3: Inward transitions into a state S_i and the impact of these transitions on the state vector.

6.3.2 Inward Transitions

Let us consider Fig. 6.3. We focus on state $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle$ in sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$ and the transitions into S_i from different states $S_k = \langle \boldsymbol{\xi}_k^s, \boldsymbol{\xi}_k^l \rangle$. These transitions can be categorised based on their type. If $S_k \xrightarrow{(\alpha,\cdot)} S_i$, $\alpha \in \overrightarrow{\mathcal{A}}_s^*(\mathbb{M})$, then $\mathcal{V}_\alpha^s \neq 0$, $\mathcal{V}_\alpha^l = 0$ and $\boldsymbol{\xi}_k^s = \boldsymbol{\gamma} - \mathcal{V}_\alpha^s$, $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l$, $S_k \in \mathbb{Y}_{\boldsymbol{\gamma} - \mathcal{V}_\alpha^s}$. This transition exists if $\boldsymbol{\xi}_k^s \geq \mathcal{V}_\alpha^{s,-}$. When $\alpha \in \overrightarrow{\mathcal{A}}_{sl}^*(\mathbb{M})$, $\mathcal{V}_\alpha^s \neq 0$ and $\mathcal{V}_\alpha^l \neq 0$, $\boldsymbol{\xi}_k^s = \boldsymbol{\gamma} - \mathcal{V}_\alpha^s$, $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l - \mathcal{V}_\alpha^l$ and $S_k \in \mathbb{Y}_{\boldsymbol{\gamma} - \mathcal{V}_\alpha^s}$. This transition exists if both small and large groups enable it, $\boldsymbol{\xi}_k^s \geq \mathcal{V}_\alpha^{s,-}$ and $\boldsymbol{\xi}_k^l \geq \mathcal{V}_\alpha^{l,-}$. Finally, when $\alpha \in \overrightarrow{\mathcal{A}}_l^*(\mathbb{M})$, $\mathcal{V}_\alpha^s = 0$, $\mathcal{V}_\alpha^l \neq 0$, $\boldsymbol{\xi}_k^s = \boldsymbol{\gamma}$, $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l - \mathcal{V}_\alpha^l$ and S_k is in the same sub-chain as S_i. This transition exists if $\boldsymbol{\xi}_k^l \geq \mathcal{V}_\alpha^{l,-}$. The properties of the inward transitions into S_i are summarised in Table.6.2.

The categorisation above is useful when applying the transformations to the term related to the inward transitions in Eq.(6.12). Consider the fourth line in this equation.

	$S_k \xrightarrow{(\alpha,\cdot)} S_i$, $S_k = \langle \boldsymbol{\xi}_k^s, \boldsymbol{\xi}_k^l \rangle$, $S_i = \langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l \rangle$		
	$\alpha \in \overrightarrow{\mathcal{A}}_s^*(\mathbb{M})$	$\alpha \in \overrightarrow{\mathcal{A}}_{sl}^*(\mathbb{M})$	$\alpha \in \overrightarrow{\mathcal{A}}_l^*(\mathbb{M})$
necessary condition	$\boldsymbol{\xi}_k^s \geq \mathcal{V}_\alpha^{s,-}$ $(\boldsymbol{\xi}_i^s \geq \mathcal{V}_\alpha^{s,+})$	$\boldsymbol{\xi}_k^s \geq \mathcal{V}_\alpha^{s,-}$ $(\boldsymbol{\xi}_i^s \geq \mathcal{V}_\alpha^{s,+})$ $\boldsymbol{\xi}_k^l \geq \mathcal{V}_\alpha^{l,-}$ $(\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-})$	$\boldsymbol{\xi}_k^l \geq \mathcal{V}_\alpha^{l,-}$ $(\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,+})$
impact	$\boldsymbol{\xi}_i^s = \boldsymbol{\xi}_k^s + \mathcal{V}_\alpha^s$ $\boldsymbol{\xi}_i^l = \boldsymbol{\xi}_k^l$	$\boldsymbol{\xi}_i^s = \boldsymbol{\xi}_k^s + \mathcal{V}_\alpha^s$ $\boldsymbol{\xi}_i^l = \boldsymbol{\xi}_k^l + \mathcal{V}_\alpha^l$	$\boldsymbol{\xi}_i^s = \boldsymbol{\xi}_k^s$ $\boldsymbol{\xi}_i^l = \boldsymbol{\xi}_k^l + \mathcal{V}_\alpha^l$
rate	$r_\alpha(\boldsymbol{\xi}_k^s)$	$r_\alpha(\boldsymbol{\xi}_k^s)$	$r_\alpha(\boldsymbol{\xi}_k^l)$

Table 6.2: The properties of the inward transitions into a state S_i

The outer summation sums over the actions. By expanding this with respect to the partition $\Delta_{\mathcal{A}}$ and using the Bayes' Law we have:

$$\begin{aligned}
&\sum_{\alpha\in\vec{\mathcal{A}}^{*}(\mathbb{M})} \sum_{S_k\geq\mathcal{V}_\alpha^-,\, S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle=S_k+\mathcal{V}_\alpha} \xi_i^l\cdot\ r_\alpha(S_k)\cdot\mathbb{P}_t(S_k)\\
&= \sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^{*}(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} \sum_{S_k=\langle\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}} \boldsymbol{\xi}_k^l\cdot r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\ \cdot \underbrace{\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)}_{\mathbb{P}_t(\langle\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle)}\\
&+ \sum_{\substack{\alpha\in\vec{\mathcal{A}}_{sl}^{*}(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} \sum_{\substack{S_k=\langle\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}\\ \boldsymbol{\xi}_k^l\geq\mathcal{V}_\alpha^{l,-},\,\boldsymbol{\xi}_i^l=\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l}} (\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l)\cdot r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\ \cdot\ \mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\ \mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\\
&+ \sum_{\alpha\in\vec{\mathcal{A}}_l^{*}(\mathbb{M})} \sum_{\substack{S_k=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}\\ \boldsymbol{\xi}_k^l\geq\mathcal{V}_\alpha^{l,-}}} (\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l)\ \cdot\ r_\alpha(\boldsymbol{\xi}_k^l)\ \cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma})\ \cdot\ \mathbb{P}_t(\boldsymbol{\gamma})
\end{aligned} \tag{6.21}$$

In Eq.(6.21) we factor out the terms related to the marginal distribution over $\boldsymbol{\xi}^s$ and the apparent rates of $\vec{\mathcal{A}}_s^{*}(\mathbb{M})$ and $\vec{\mathcal{A}}_{sl}^{*}(\mathbb{M})$ actions to obtain:

$$\begin{aligned}
&\sum_{\alpha\in\vec{\mathcal{A}}^{*}(\mathbb{M})} \sum_{S_k\geq\mathcal{V}_\alpha^-,\, S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle=S_k+\mathcal{V}_\alpha} \xi_i^l\cdot\ r_\alpha(S_k)\cdot\mathbb{P}_t(S_k)\\
&= \sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^{*}(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} \mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot \underbrace{\sum_{S_k=\langle\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}} \boldsymbol{\xi}_k^l\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)}_{\mathbb{E}_t\left[\boldsymbol{\xi}^l\mid\boldsymbol{\gamma}-\mathcal{V}_\alpha^s\right]}\\
&+ \sum_{\substack{\alpha\in\vec{\mathcal{A}}_{sl}^{*}(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} \mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s) \sum_{\substack{S_k=\langle\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}\\ \boldsymbol{\xi}_k^l\geq\mathcal{V}_\alpha^{l,-}}} (\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l)\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\ \star\\
&+ \sum_{\alpha\in\vec{\mathcal{A}}_l^{*}(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma})\cdot \sum_{\substack{S_k=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}\\ \boldsymbol{\xi}_k^l\geq\mathcal{V}_\alpha^{l,-}}} (\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l)\ \cdot\ r_\alpha(\boldsymbol{\xi}_k^l)\ \cdot\ \mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma})
\end{aligned} \tag{6.22}$$

Boundary state approximation. The probability of being in boundary states is assumed to be close to zero. Thus, an approximation similar to Eq.(6.15) is possible with respect to the term in Eq.(6.22) related to $\vec{\mathcal{A}}_{sl}^{*}(\mathbb{M})$ actions (marked with a star sign). This approximation states that if there is a transition $S_k \xrightarrow{(\alpha,R)} S_i$, $\alpha\in\vec{\mathcal{A}}_{sl}^{*}(\mathbb{M})$, from a state $S_k\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}$ to a state $S_i\in\mathbb{Y}_{\boldsymbol{\gamma}}$, then we assume that all $S_k^{'}\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}$ enable

similar transitions $S'_k \xrightarrow{(\alpha,R)} S'_i$, $S'_i \in \mathbb{Y}_{\boldsymbol{\gamma}}$ to their corresponding S'_i.

$$\sum_{\substack{\alpha \in \overrightarrow{\mathcal{A}}^*_{sl}(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}^{s,+}_\alpha}} \mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \cdot r_\alpha(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \sum_{\substack{S_k = \langle \boldsymbol{\gamma} - \mathcal{V}^s_\alpha, \boldsymbol{\xi}^l_k \rangle \in \mathbb{Y}_{\boldsymbol{\gamma} - \mathcal{V}^s_\alpha} \\ \boldsymbol{\xi}^l_k \geq \mathcal{V}^{l,-}_\alpha}} (\boldsymbol{\xi}^l_k + \mathcal{V}^l_\alpha) \cdot \mathbb{P}_t(\boldsymbol{\xi}^l_k \mid \boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \approx$$

$$\sum_{\substack{\alpha \in \overrightarrow{\mathcal{A}}^*_{sl}(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}^{s,+}_\alpha}} \mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \cdot r_\alpha(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \sum_{S_k = \langle \boldsymbol{\gamma} - \mathcal{V}^s_\alpha, \boldsymbol{\xi}^l_k \rangle \in \mathbb{Y}_{\boldsymbol{\gamma} - \mathcal{V}^s_\alpha}} (\boldsymbol{\xi}^l_k + \mathcal{V}^l_\alpha) \cdot \mathbb{P}_t(\boldsymbol{\xi}^l_k \mid \boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \tag{6.23}$$

We substitute Eq.(6.23) in Eq.(6.22). Also, in the last line of Eq. (6.22), which is related to $\overrightarrow{\mathcal{A}}^*_l(\mathbb{M})$, we apply the technique of inclusion of non-enabling states to change the constraints of the summation over the states from $S_k = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}^l_k \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\xi}^l_k \geq \mathcal{V}^{l,-}_\alpha$ into $S_k = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}^l_k \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}$.

$$\begin{aligned}
&\sum_{\alpha \in \overrightarrow{\mathcal{A}}^*(\mathbb{M})} \sum_{S_k \geq \mathcal{V}^-_\alpha, S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}^l_i \rangle = S_k + \mathcal{V}_\alpha} \xi^l_i \cdot r_\alpha(S_k) \cdot \mathbb{P}_t(S_k) \\
&\approx \sum_{\substack{\alpha \in \overrightarrow{\mathcal{A}}^*_s(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}^{s,+}_\alpha}} \mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \cdot r_\alpha(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma} - \mathcal{V}^s_\alpha\right] \\
&+ \sum_{\substack{\alpha \in \overrightarrow{\mathcal{A}}^*_{sl}(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}^{s,+}_\alpha}} \mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \cdot r_\alpha(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \underline{\sum_{S_k = \langle \boldsymbol{\gamma} - \mathcal{V}^s_\alpha, \boldsymbol{\xi}^l_k \rangle \in \mathbb{Y}_{\boldsymbol{\gamma} - \mathcal{V}^s_\alpha}} (\boldsymbol{\xi}^l_k + \mathcal{V}^l_\alpha) \cdot \mathbb{P}_t(\boldsymbol{\xi}^l_k \mid \boldsymbol{\gamma} - \mathcal{V}^s_\alpha)} \\
&+ \sum_{\alpha \in \overrightarrow{\mathcal{A}}^*_l(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \underline{\sum_{S_k = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}^l_k \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} (\boldsymbol{\xi}^l_k + \mathcal{V}^l_\alpha) \cdot r_\alpha(\boldsymbol{\xi}^l_k) \cdot \mathbb{P}_t(\boldsymbol{\xi}^l_k \mid \boldsymbol{\gamma})}
\end{aligned} \tag{6.24}$$

In the final step, using the distributive property of multiplication over addition we expand the terms underlined in Eq.(6.24). Considering the first term we have:

$$\begin{aligned}
&\sum_{S_k = \langle \boldsymbol{\gamma} - \mathcal{V}^s_\alpha, \boldsymbol{\xi}^l_k \rangle \in \mathbb{Y}_{\boldsymbol{\gamma} - \mathcal{V}^s_\alpha}} (\boldsymbol{\xi}^l_k + \mathcal{V}^l_\alpha) \cdot \mathbb{P}_t(\boldsymbol{\xi}^l_k \mid \boldsymbol{\gamma} - \mathcal{V}^s_\alpha) = \\
&\sum_{S_k = \langle \boldsymbol{\gamma} - \mathcal{V}^s_\alpha, \boldsymbol{\xi}^l_k \rangle \in \mathbb{Y}_{\boldsymbol{\gamma} - \mathcal{V}^s_\alpha}} \boldsymbol{\xi}^l_k \cdot \mathbb{P}_t(\boldsymbol{\xi}^l_k \mid \boldsymbol{\gamma} - \mathcal{V}^s_\alpha) + \sum_{S_k = \langle \boldsymbol{\gamma} - \mathcal{V}^s_\alpha, \boldsymbol{\xi}^l_k \rangle \in \mathbb{Y}_{\boldsymbol{\gamma} - \mathcal{V}^s_\alpha}} \Big(\mathcal{V}^l_\alpha \cdot \mathbb{P}_t(\boldsymbol{\xi}^l_k \mid \boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \Big)
\end{aligned} \tag{6.25}$$

where:

$$\sum_{S_k=\langle \boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}} \Big(\ \mathcal{V}_\alpha^l\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\ \Big) \;=\; \mathcal{V}_\alpha^l\cdot \sum_{S_k=\langle \boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}} \mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma}-\mathcal{V}_\alpha^s) \;=$$

$$\mathcal{V}_\alpha^l\cdot \sum_{S_k=\langle \boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}} \frac{\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l)}{\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)} \;=\; \mathcal{V}_\alpha^l\cdot\frac{\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)}{\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)} \;=\; \mathcal{V}_\alpha^l \tag{6.26}$$

Also, for the second underlined term in Eq. (6.24), we have:

$$\sum_{S_k=\langle \boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}} (\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l)\ \cdot\ r_\alpha(\boldsymbol{\xi}_k^l)\ \cdot\ \mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma})\ =$$

$$\sum_{S_k=\langle \boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_k^l\ \cdot\ r_\alpha(\boldsymbol{\xi}_k^l)\ \cdot\ \mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma})\ +\ \sum_{S_k=\langle \boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}} \mathcal{V}_\alpha^l\ \cdot\ r_\alpha(\boldsymbol{\xi}_k^l)\ \cdot\ \mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma}) \tag{6.27}$$

We substitute Eq.(6.25) and Eq.(6.27) in Eq.(6.24). Then, we use the definition of conditional expectation (Def. 28) to obtain:

$$\begin{aligned}
&\sum_{S_k\xrightarrow{(\alpha,r_\alpha(S_k))}S_i}\ \sum_{S_i\in\mathbb{Y}_{\boldsymbol{\gamma}}} \xi_i^l\cdot\ r_\alpha(S_k)\cdot\mathbb{P}_t(S_k)\\
&\approx \sum_{\substack{\alpha\in\overrightarrow{\mathcal{A}}_s^*(\mathbb{M})\cup\overrightarrow{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} \mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\ \cdot\mathbb{E}_t\Big[\boldsymbol{\xi}^l\mid\boldsymbol{\gamma}-\mathcal{V}_\alpha^s\Big]\\
&+\sum_{\substack{\alpha\in\overrightarrow{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} \mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\ \cdot\mathcal{V}_\alpha^l\\
&+\sum_{\alpha\in\overrightarrow{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma})\ \cdot\ \mathbb{E}_t\Big[\boldsymbol{\xi}^l\cdot r_\alpha(\boldsymbol{\xi}^l)\mid\boldsymbol{\gamma}\Big]\\
&+\sum_{\alpha\in\overrightarrow{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma})\ \cdot\ \mathcal{V}_\alpha^l\ \cdot\ \mathbb{E}_t\Big[r_\alpha(\boldsymbol{\xi}^l)\mid\boldsymbol{\gamma}\Big]
\end{aligned} \tag{6.28}$$

The transformations applied to the term related to the inward transitions in Eq.(6.12)

led to Eq.(6.28). Now we substitute this result into Eq.(6.12) to obtain:

$$\forall\, \boldsymbol{\gamma} \in D^{agg} : \frac{d\left(\mathbb{P}_t(\boldsymbol{\gamma})\right)}{d\,t} \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right] \;+\; \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \frac{d\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\right)}{d\,t} \;\approx$$

$$\left.\begin{aligned} &- \sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \;\cdot\; r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right] \\ &- \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \;\cdot\; \mathbb{E}_t\left[\boldsymbol{\xi}^l \cdot r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\right] \qquad \star \end{aligned}\right\} \text{outward transitions}$$

$$\left.\begin{aligned} &+ \sum_{\substack{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+}}} \mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma} - \mathcal{V}_\alpha^s\right] \\ &+ \sum_{\substack{\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+}}} \mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \mathcal{V}_\alpha^l \\ &+ \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \;\cdot\; \mathbb{E}_t\left[\boldsymbol{\xi}^l \cdot r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\right] \qquad \star \\ &+ \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \;\cdot\; \mathcal{V}_\alpha^l \;\cdot\; \mathbb{E}_t\left[r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\right] \end{aligned}\right\} \text{inward transitions} \tag{6.29}$$

In Eq.(6.29), one important simplification is possible; the two terms marked by $\star$ are equal but with opposite signs and therefore, reduce to zero. As this will be highlighted later in the case of higher-order moments, this simplification implies that the system can become closed in terms of the conditional expectations (first-order conditional moments) and the equations for the higher-order moments are not required for obtaining these first-order moments.

In Eq.(6.29) – after simplification – all expressions are closed in terms of the marginal probability distribution over D^{agg} and the conditional expectations of $\boldsymbol{\xi}^l$, *except the term underlined*, which is related to the conditional expectations of the apparent rate functions of the actions in $\vec{\mathcal{A}}_l^*(\mathbb{M})$. This term needs to be transformed so that the whole system becomes closed in terms of the said primitives. For this transformation, we present Lemma 6.3.1, one approximation step and Lemma 6.3.2.

Lemma 6.3.1. (Expectation of a linear function of a random variable) *Let X be a*

random variable and $F : X \to \mathbb{R}$ *be a* linear *function defined on* X*. The expectation of* F*, denoted by* $\mathbb{E}[F(X)]$*, is equal to the function* F *evaluated at* $\mathbb{E}[X]$*:*

$$\mathbb{E}[F(X)] = F(\mathbb{E}[X]) \tag{6.30}$$

The proof of this lemma is simple and presented in Appendix B.1. We use the Taylor expansion of $F(X)$ around $\mathbb{E}[X]$ and in this sense, our proof is different from the more straightforward one usually presented in the literature. Reading the proof helps with understanding of our more complex derivations in the next sections.

Next, we present an approximation step required for our transformation.

Approximating rate functions. Consider a shared action type $\alpha \in \overrightarrow{\mathcal{A}}_l^*(\mathbb{M})$. Since α is shared, the apparent rate function of α contains the minimum expression: $r_\alpha(\boldsymbol{\xi}^l) = \min_{(H,C)\in en_\alpha(\mathbb{M})} r_\alpha(\boldsymbol{\xi}(H))$. The *conditional* expectation of $r_\alpha(\boldsymbol{\xi}^l)$, given that $\boldsymbol{\xi}^s = \boldsymbol{\gamma}$, is denoted by $\mathbb{E}_t\left[r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\right]$:

$$\mathbb{E}_t\left[r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\right] = \mathbb{E}_t\left[\min_{(H,C)\in en_\alpha(\mathbb{M})} \big(r_\alpha(\boldsymbol{\xi}(H)) \big) \mid \boldsymbol{\gamma}\right] \tag{6.31}$$

We will approximate $\mathbb{E}_t\left[r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\right]$ by swapping the position of the min and expectation operators in Eq.(6.31):

$$\mathbb{E}_t\left[r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\right] \approx \min_{(H,C)\in en_\alpha(\mathbb{M})} \big(\mathbb{E}_t\left[r_\alpha(\boldsymbol{\xi}(H)) \mid \boldsymbol{\gamma}\right] \big)$$

Since our models respect split-freeness, $\forall\ \alpha \in \overrightarrow{\mathcal{A}}^*(\mathbb{M})\ ,\ \forall\ H \in \widehat{en}(\alpha)$, the function $r_\alpha(\boldsymbol{\xi}(H))$, i.e. the apparent rate of α with respect to group H, is a *linear* function. Thus, using Lemma 6.3.1:

$$\mathbb{E}_t\left[r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\right] \approx \min_{(H,C)\in en_\alpha(\mathbb{M})} \big(\mathbb{E}_t\left[r_\alpha(\boldsymbol{\xi}(H)) \mid \boldsymbol{\gamma}\right] \big) = \min_{(H,C)\in en_\alpha(\mathbb{M})} \Big(r_\alpha\big(\mathbb{E}_t\left[\boldsymbol{\xi}(H) \mid \boldsymbol{\gamma}\right] \big) \Big) \tag{6.32}$$

For each shared action type $\alpha \in \overrightarrow{\mathcal{A}}_l^*(\mathbb{M})$, Eq.(6.32) approximates $\mathbb{E}_t\left[r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\right]$ by an expression closed in terms of the conditional expectations.

Lemma 6.3.1 and the approximation step above lead us to Lemma 6.3.2. In Lemma 6.3.2, we propose an approximation to the conditional expectation of the rate functions.

Lemma 6.3.2. (Approximating conditional expectation of apparent rate function) *Let* α *be an action type in* $\overrightarrow{\mathcal{A}}_l^*(\mathbb{M})$ *and* $\mathbb{Y}_{\boldsymbol{\gamma}}$ *a sub-chain in* D*. At time* t*, given that* $\boldsymbol{\xi}^s = \boldsymbol{\gamma}$*,*

the conditional expectation $\mathbb{E}_t\left[r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\right]$ *can be approximated as:*

$$\mathbb{E}_t\left[r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\right] \approx r_\alpha\Big(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\Big) \tag{6.33}$$

Proof. The proof follows from Lemma 6.3.1 and Eq.(6.32). In the case when α is an individual action, $r_\alpha(\boldsymbol{\xi}^l)$ is a linear function and Eq.(6.33) follows from Lemma 6.3.1. In this case, the approximation is exact. In the case when α is a shared action, we use Eq.(6.32) to obtain Eq.(6.33):

$$\mathbb{E}_t\left[r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\right] \approx \min_{(H,C)\in en_\alpha(\mathbb{M})}\Big(\, r_\alpha\big(\,\mathbb{E}_t\left[\boldsymbol{\xi}(H) \mid \boldsymbol{\gamma}\right]\,\big)\,\Big) = r_\alpha\Big(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\Big) \tag{6.34}$$

□

Now we consider the expression underlined in Eq.(6.29). Using Lemma 6.3.2 we can transform the expression as:

$$\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \mathcal{V}_\alpha^l \cdot \mathbb{E}_t\left[r_\alpha(\boldsymbol{\xi}^l) \mid \boldsymbol{\gamma}\right] \approx \sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \mathcal{V}_\alpha^l \cdot r_\alpha\Big(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\Big) \tag{6.35}$$

By substituting Eq.(6.35) in Eq.(6.29), we derive:

$$\begin{aligned}
\forall\ \boldsymbol{\gamma}\in D^{agg}:\quad & \mathbb{P}_t(\boldsymbol{\gamma})\cdot\frac{d\,\Big(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\Big)}{d\,t} + \frac{d\,\Big(\mathbb{P}_t(\boldsymbol{\gamma})\Big)}{d\,t}\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right] \approx \\
& - \sum_{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma})\cdot r_\alpha(\boldsymbol{\gamma})\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right] \\
& + \sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} \mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}-\mathcal{V}_\alpha^s\right] \\
& + \sum_{\substack{\alpha\in\vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} \mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\mathcal{V}_\alpha^l \\
& + \sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma})\cdot\mathcal{V}_\alpha^l\cdot r_\alpha\Big(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\Big)
\end{aligned} \tag{6.36}$$

As a result of this substitution, Eq.(6.36) finally becomes closed in terms of the marginal distribution over D^{agg} and the conditional expectations of $\boldsymbol{\xi}^l$.

In Eq.(6.36), the conditional expectations of $\boldsymbol{\xi}^l$ are presented in the vector form; given that $\boldsymbol{\xi}^l$ consists of a number of state variables the equation illustrates how the expectation of the whole vector $\boldsymbol{\xi}^l$ evolves. It is also useful to derive an alternative system of equations that is at a finer granularity and captures the conditional expectation of *each of the state variables* in $\boldsymbol{\xi}^l$. The new system is particularly useful for software implementation purposes.

Assuming $\boldsymbol{\xi}^l = \langle \xi(H,C) \rangle$, $H \in \mathcal{G}_l(\mathbb{M})$, $C \in ds^*(H)$, at any time t the conditional expectation of each state variable $\xi(H,C)$, given that at t, the configuration of small groups is $\boldsymbol{\gamma}$, is denoted by $\mathbb{E}_t\Big[\xi(H,C) \,|\, \boldsymbol{\gamma}\Big]$:

$$\mathbb{E}_t\Big[\xi(H,C) \,|\, \boldsymbol{\gamma}\Big] \;=\; \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \xi_i(H,C) \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \tag{6.37}$$

Def. 28 implies that $\mathbb{E}_t\Big[\xi(H,C) \,|\, \boldsymbol{\gamma}\Big]$ is the element in the expectation vector $\mathbb{E}_t\Big[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\Big]$ corresponding to the state variable $\xi(H,C)$. Therefore, $\mathbb{E}_t\Big[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\Big]$ can be written as the indexed vector of element-wise conditional expectations:

$$\mathbb{E}_t\Big[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\Big] = \Big\langle\, \mathbb{E}_{(H,C),t}\Big[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\Big] \,\Big\rangle \;,\; H \in \mathcal{G}_l(\mathbb{M}) \;,\; C \in ds^*(H) \;,$$

$$\mathbb{E}_{(H,C),t}\Big[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\Big] = \mathbb{E}_t\Big[\xi(H,C) \,|\, \boldsymbol{\gamma}\Big] \tag{6.38}$$

Using Eq.(6.29) and Eq.(6.38) we now present our alternative system of equations. Here, the following notation is used:

$$\mu(\boldsymbol{\gamma},t) \;=\; \mathbb{E}_t\Big[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\Big] \qquad\qquad \forall\; \boldsymbol{\gamma} \in D^{agg}$$

$$\mu_{(H,C)}(\boldsymbol{\gamma},t) \;=\; \mathbb{E}_{(H,C),t}\Big[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\Big] \qquad \forall\, \boldsymbol{\gamma} \in D^{agg} \;,\; \forall\, H \in \mathcal{G}_l(\mathbb{M}) \;,\forall\, C \in ds^*(H)$$

$$\forall\ \boldsymbol{\gamma} \in D^{agg}\ ,\ \forall\ (H,C)\ :\ \xi(H,C) \in \boldsymbol{\xi}^l :$$

$$\begin{aligned}
&\mathbb{P}_t(\boldsymbol{\gamma}) \cdot \frac{d\left(\mu_{(H,C)}(\boldsymbol{\gamma},t)\right)}{d\,t} + \frac{d\left(\mathbb{P}_t(\boldsymbol{\gamma})\right)}{d\,t} \cdot \mu_{(H,C)}(\boldsymbol{\gamma},t) \approx \\
&- \sum_{\alpha \in \vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot r_\alpha(\boldsymbol{\gamma}) \cdot \mu_{(H,C)}(\boldsymbol{\gamma},t) \\
&+ \sum_{\substack{\alpha \in \vec{\mathcal{A}}^*_s(\mathbb{M}) \cup \vec{\mathcal{A}}^*_{sl}(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}^{s,+}_\alpha}} \mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \cdot r_\alpha(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \cdot \mu_{(H,C)}(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha, t) \\
&+ \sum_{\substack{\alpha \in \vec{\mathcal{A}}^*_{sl}(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}^{s,+}_\alpha}} \mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \cdot r_\alpha(\boldsymbol{\gamma} - \mathcal{V}^s_\alpha) \cdot \mathcal{V}^l_{\alpha,(H,C)} \\
&+ \sum_{\alpha \in \vec{\mathcal{A}}^*_l(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \mathcal{V}^l_{\alpha,(H,C)} \cdot r_\alpha\Big(\mu(\boldsymbol{\gamma},t)\Big)
\end{aligned} \tag{6.39}$$

6.4 Construction of Initial Values

The analysis through the MCM gives rise to a system of DAEs where for each $\boldsymbol{\gamma} \in D^{agg}$ one equation is formed (in the vector form) capturing the evolution of the conditional expectation $\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$. The DAEs can be regarded as an initial value problem and in order to numerically obtain the solution a set of initial values needs to be specified. In this case, the initial values required consist of the *initial conditional expectations*, i.e. $\mathbb{E}_{t=t_0}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$, $\boldsymbol{\gamma} \in D^{agg}$, and the *initial derivatives* of these expectations, i.e. $\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]}{d\,t}\Big|_{t=0}$.

Typically, the initial values are obtained by a simple procedure which processes an *initial probability distribution* that the modeller specifies over the complete state space. This procedure, which will be called *simple initial value extraction* (SIVE), has the following steps:

1. Using the initial distribution, for each $\boldsymbol{\gamma} \in D^{agg}$ the initial marginal distribution $\mathbb{P}_{t_0}(\boldsymbol{\gamma})$ and the initial conditional distribution $\mathbb{P}_{t=t_0}(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma})$ are derived.

2. Using the definition of the conditional expectation (Def. 28) and evaluating it at $t = t_0$, $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right]$ is obtained.

3. Finally $\mathbb{P}_{t_0}(\boldsymbol{\gamma})$ and $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right]$ are substituted in the instance of Eq.(6.35) that captures the evolution of $\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$ to derive the initial derivative $\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\,t}|_{t=t_0}$.

Although simple, in practice the application of SIVE is not trivial. The potential difficulty arises when there are sub-chains in the initial distribution with zero initial probability. This makes calculation of their conditional probability distributions (first step) impossible. In this section we discuss this issue and propose a solution.

6.4.1 Types of the initial distribution

Let us consider the application of SIVE under two scenarios. Depending on the initial distribution given, one of the following is the case.

Case 1 - $\forall\, \boldsymbol{\gamma} \in D^{agg} : \mathbb{P}_{t_0}(\boldsymbol{\gamma}) \neq 0$: At $t = t_0$ each aggregated state is allocated a *non-zero* probability mass. In this case, no difficulty arises when applying SIVE and it successfully constructs the initial values. As an example, suppose that at $t = t_0$, $\boldsymbol{\xi}^s$ and $\boldsymbol{\xi}^l$ are independently distributed, where $\boldsymbol{\xi}^s$ follows a normal distribution $\mathcal{N}_{t_0}^{\boldsymbol{\gamma}}$ (truncated to have bounded support) with f as its probability density function (PDF), and in each sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$, $\boldsymbol{\xi}^l$ follows a multivariate normal distribution $\mathcal{N}(\mu^{\boldsymbol{\gamma}}, \Sigma^{\boldsymbol{\gamma}})$ (again, with bounded support) with $\mu^{\boldsymbol{\gamma}}$ as its mean vector, $\Sigma^{\boldsymbol{\gamma}}$ as its covariance matrix and $g_{\boldsymbol{\gamma}}(\boldsymbol{\xi}^l)$ as its PDF. Given the independence of $\boldsymbol{\xi}^s$ and $\boldsymbol{\xi}^l$, in the initial distribution, for all $\langle \boldsymbol{\gamma}, \boldsymbol{\xi}^l \rangle \in D : \mathbb{P}_{t_0}(\langle \boldsymbol{\gamma}, \boldsymbol{\xi}^l \rangle) = f(\boldsymbol{\gamma}) \cdot g_{\boldsymbol{\gamma}}(\boldsymbol{\xi}^l)$. Also, since $\mathbb{P}_{t_0}(\boldsymbol{\gamma}) \neq 0$ the conditional probability distribution $\mathbb{P}_{t_0}(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma})$ is derivable by Bayes' Law as:

$$\forall\, \boldsymbol{\gamma} \in D^{agg} \;:\; \mathbb{P}_{t_0}(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}) = \frac{\mathbb{P}_{t_0}(\boldsymbol{\gamma}, \boldsymbol{\xi}^l)}{\mathbb{P}_{t_0}(\boldsymbol{\gamma})} = \frac{f(\boldsymbol{\gamma}) \cdot g_{\boldsymbol{\gamma}}(\boldsymbol{\xi}^l)}{f(\boldsymbol{\gamma})} = g_{\boldsymbol{\gamma}}(\boldsymbol{\xi}^l) \tag{6.40}$$

Next, using the conditional probabilities, the initial conditional expectations are derived as:

$$\forall\, \boldsymbol{\gamma} \in D^{agg} \;:\; \mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right] = \sum_{\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot \mathbb{P}_{t_0}(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) = \sum_{\langle \boldsymbol{\gamma}, \boldsymbol{\xi}^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \boldsymbol{\xi}_i^l \cdot g_{\boldsymbol{\gamma}}(\boldsymbol{\xi}_i^l) = \mu^{\boldsymbol{\gamma}} \tag{6.41}$$

Having $\mathbb{P}_{t_0}(\boldsymbol{\gamma})$ and $\mu^{\boldsymbol{\gamma}}$, for a $\boldsymbol{\gamma}$ these are substituted in the conditional expectation equation related to $\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$ and the equation is evaluated at t_0. As a result, the only

unknown variable remaining is $\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\,t}\Big|_{t=t_0}$, which can be easily obtained by the re-ordering of the terms. Repeating for all $\boldsymbol{\gamma} \in D^{agg}$, we obtain all the required initial values.

Case 2 - $\exists\ \boldsymbol{\gamma} \in D^{agg} : \mathbb{P}_{t_0}(\boldsymbol{\gamma}) = 0$: In the initial distribution, there are a number of aggregated states with *zero* initial probability. This case, for instance, includes the situation where the system is assumed to start from one *initial state* S_{init} and in the initial distribution, all probability mass is allocated to that state. Here, when building the initial marginal distribution over the aggregated state space, the aggregated state $\boldsymbol{\gamma} : S_{init} \in \mathbb{Y}_{\boldsymbol{\gamma}}$ (whose related sub-chain in the complete state space encompasses S_{init}) has $\mathbb{P}_{t_0}(\boldsymbol{\gamma}) = 1$ and all other aggregated states have zero probability.

For an analysis of this case, we construct the partition $\Delta_Z = \{D_z^{agg}, D_{nz}^{agg}\}$ over D^{agg} where D_z^{agg} is the set of aggregated states with zero initial probability and D_{nz}^{agg} is the set of aggregated states $\boldsymbol{\gamma}$ such that $\mathbb{P}_{t_0}(\boldsymbol{\gamma}) \neq 0$. When constructing the initial values, for any state in D_{nz}^{agg} we follow SIVE, and obtain the related initial values straightforwardly. However, for any aggregated state $\boldsymbol{\gamma} \in D_z^{agg}$ the conditional probabilities $\mathbb{P}_{t_0}(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma})$ cannot be determined by Bayes's Law, and consequently, neither the initial conditional expectations nor their derivatives may be obtained by considering the initial distribution. For $\boldsymbol{\gamma} \in D_z^{agg}$, due to the division by zero, the initial values $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$ and $\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\,t}\Big|_{t=t_0}$ become *undetermined*.

6.4.2 Solver Requirement

The numerical solvers used for solving DAEs require a complete set of initial values; i.e. conditional expectations and their derivative for all aggregated states and at $t = t_0$. This is a necessary requirement for the iterative integration step within the solvers [24]. For analyses in Case 1, the complete set of initial values is derived from the initial distribution. However, as stated, in Case 2 these initial values are only partially constructed. This issue is circumvented by a standard technique where for $\boldsymbol{\gamma} \in D_z^{agg}$ instead of the extracting the values $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$ and $\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\,t}\Big|_{t=t_0}$ from the initial distribution, we specify initial values in such a way that they are consistent with the system of equations and the determined conditional expectations, and lie within the DAEs solution space (a manifold in $\mathbb{R}^d$ where d is the dimension of the state vector). The derivation of these additional values is not trivial and depends on the structure of

the equations. Here we propose a schema suitable for DAEs constructed when MCM is applied to LSRB models.

6.4.3 Deriving Consistent Initial Values

In order to calculate the consistent initial values we need the following definition and proposition.

Definition 30. (Initial distance of a state in D^{agg} from D^{agg}_{nz} states) *For any aggregated state $\boldsymbol{\gamma} \in D^{agg}$, we define $\nabla_{\boldsymbol{\gamma}}$ as the* minimum number of transitions *required to reach $\boldsymbol{\gamma}$ from any one of the states in D^{agg}_{nz}. For $\boldsymbol{\gamma} \in D^{agg}_{nz}$: $\nabla_{\boldsymbol{\gamma}} = 0$.*

Using Def. 30, a distance measure is assigned to each state $\boldsymbol{\gamma} \in D^{agg}$. This captures the distance of $\boldsymbol{\gamma}$ from the probability sources within D^{agg}. This notion has two important properties. First, a state $\boldsymbol{\gamma} \in D^{agg}_{z}$ has distance $\nabla_{\boldsymbol{\gamma}}$ if and only if, for all transitions $\boldsymbol{\gamma}^{'} \rightarrow \boldsymbol{\gamma}$, $\boldsymbol{\gamma}^{'} \in D^{agg}$, the state $\boldsymbol{\gamma}^{'}$ has the distance $\nabla_{\boldsymbol{\gamma}^{'}} \geq \nabla_{\boldsymbol{\gamma}} - 1$. Moreover, for $\boldsymbol{\gamma}$ to have the distance $\nabla_{\boldsymbol{\gamma}}$, it is necessary that there exists at least one state $\boldsymbol{\gamma}^{'}$ with distance $\nabla_{\boldsymbol{\gamma}^{'}} = \nabla_{\boldsymbol{\gamma}} - 1$ enabling a transition into $\boldsymbol{\gamma}$. An example will be presented shortly.

The distance measure $\nabla_{\boldsymbol{\gamma}}$, $\boldsymbol{\gamma} \in D^{agg}_{z}$ allows us to express the following important property about the probability of being in D^{agg}_{z} states at $t = t_0$.

Proposition 6.4.1. *For any aggregated state $\boldsymbol{\gamma} \in D^{agg}_{z}$ with initial distance $\nabla_{\boldsymbol{\gamma}}$, the first $(\nabla_{\boldsymbol{\gamma}} - 1)$-th derivatives of $\mathbb{P}_t(\boldsymbol{\gamma})$, evaluated at t_0, are equal to* zero*, and the $(\nabla_{\boldsymbol{\gamma}})$-th derivative is* non-zero*. More formally, for $k = 0, 1, \cdots, (\nabla_{\boldsymbol{\gamma}} - 1)$: $\frac{d^k\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\ t^k}\Big|_{t_0} = 0$, and when $k = \nabla_{\boldsymbol{\gamma}}$, $\frac{d^k\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\ t^k}\Big|_{t_0} \neq 0$.*

Proof. The proof is straightforward and is done by an induction on $\nabla_{\boldsymbol{\gamma}}$ and considering the Chapman-Kolmogorov equation that captures the evolution of $\mathbb{P}_t(\boldsymbol{\gamma})$. □

Prop. 6.4.1 enables us to develop a technique for finding initial values for D^{agg}_{z} states. Due to its complexity, first we will illustrate the technique using a simple example and then present it formally.

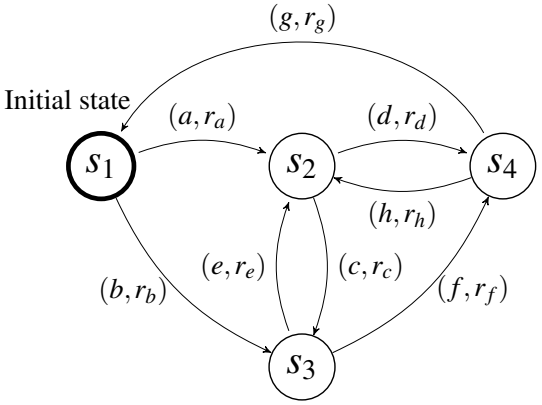


Figure 6.4: The aggregated state space of the system we consider in the example. For the action types $\overrightarrow{\mathcal{A}}_s^*(\mathbb{M}) = \{b,c,d,e,f,g,h\}$, $\overrightarrow{\mathcal{A}}_{sl}^*(\mathbb{M}) = \{a\}$ and $\overrightarrow{\mathcal{A}}_l^*(\mathbb{M}) = \{q\}$ (note that action q is not visible in D^{agg}).

6.4.3.1 Derivation of Initial Conditional Expectations - Example

Consider a model with the complete state space D, the underlying state vector $\boldsymbol{\xi} = \langle \boldsymbol{\xi}^s, \boldsymbol{\xi}^l \rangle$ and the aggregated state space D^{agg} as shown in Fig. 6.4. The model's action types are $\overrightarrow{\mathcal{A}}^*(\mathbb{M}) = \{a,b,c,d,e,f,g,h,q\}$ where $\overrightarrow{\mathcal{A}}_s^*(\mathbb{M}) = \{b,c,d,e,f,g,h\}$, $\overrightarrow{\mathcal{A}}_{sl}^*(\mathbb{M}) = \{a\}$ and $\overrightarrow{\mathcal{A}}_l^*(\mathbb{M}) = \{q\}$.

With respect to the model's initial configuration, we assume that at $t = t_0$ the model starts from one singly specified initial state $S_{init} = \langle \boldsymbol{\xi}_{init}^s, \boldsymbol{\xi}_{init}^l \rangle$ where $\boldsymbol{\xi}_{init}^s = S_1$ is the initial configuration of the small groups and $\boldsymbol{\xi}_{init}^l$ is the initial configuration of the large ones. Since at t_0, all probability mass is allocated at $S_{init} = \langle S_1, \boldsymbol{\xi}_{init}^l \rangle$, essentially, the initial distribution $\mathbb{P}_{t_0}(S)$, $S \in D$ has been specified as:

$$\forall\ \langle \boldsymbol{\xi}^s, \boldsymbol{\xi}^l \rangle \in D\ :\ \mathbb{P}_{t_0}(\langle \boldsymbol{\xi}^s, \boldsymbol{\xi}^l \rangle) = \begin{cases} 1 & \boldsymbol{\xi}^s = S_1\ \wedge\ \boldsymbol{\xi}^l = \boldsymbol{\xi}_{init}^l \\ 0 & \text{otherwise} \end{cases} \tag{6.42}$$

This initial distribution is used to construct the corresponding initial marginal distribution over D^{agg}:

$$\mathbb{P}_{t_0}(S_1) = 1 \quad , \quad \mathbb{P}_{t_0}(S_2) = \mathbb{P}_{t_0}(S_3) = \mathbb{P}_{t_0}(S4) = 0 \tag{6.43}$$

With respect to the calculation of the initial values, the partition $\Delta_{\mathcal{Z}} = \{D_z^{agg}, D_{nz}^{agg}\}$ is constructed where $D_{nz}^{agg} = \{S_1\}$ and $D_z^{agg} = \{S_2, S_3, S_4\}$. For S_1, the procedure SIVE is followed and $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_1\right]$ and $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^l | S_1\right]}{d\ t}\Big|_{t_0}$ are obtained. However, when applying the

	Distance	Initial probability	First derivative	Second derivative
S_1	$\nabla_{S_1} = 0$	$\mathbb{P}_{t_0}(S_1) \neq 0$	-	-
S_2	$\nabla_{S_2} = 1$	$\mathbb{P}_{t_0}(S_2) = 0$	$\frac{d\,\mathbb{P}_t(S_2)}{d\,t}\Big\vert_{t_0} \neq 0$	-
S_3	$\nabla_{S_3} = 1$	$\mathbb{P}_{t_0}(S_3) = 0$	$\frac{d\,\mathbb{P}_t(S_3)}{d\,t}\Big\vert_{t_0} \neq 0$	-
S_4	$\nabla_{S_4} = 2$	$\mathbb{P}_{t_0}(S_4) = 0$	$\frac{d\,\mathbb{P}_t(S_4)}{d\,t}\Big\vert_{t_0} = 0$	$\frac{d^2\,(\mathbb{P}_t(S_4))}{d\,t^2}\Big\vert_{t_0} \neq 0$

Table 6.3: The derivatives of $\mathbb{P}_t(\boldsymbol{\gamma}), \boldsymbol{\gamma} \in D^{agg}$ at $t = t_0$.

procedure, for $\boldsymbol{\gamma} \in \{S_2, S_3, S_4\}$, the initial values $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right]$ and $\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\,t}\Big\vert_{t_0}$ become undetermined.

Using Def. 30, a distance measure is assigned to each aggregated state. Here, $\nabla_{S_1=0}$, $\nabla_{S_2} = \nabla_{S_3} = 1$ and $\nabla_{S_4} = 2$. Focusing, for instance, on S_2, for its inward transitions $S_1 \to S_2$, $S_3 \to S_2$ and $S_4 \to S_2$, we have: $\nabla_{S_1} < \nabla_{S_2}$, $\nabla_{S_3} = \nabla_{S_2}$ and $\nabla_{S_4} > \nabla_{S_2}$. Using these distances, we can check the validity of Prop. 6.4.1 (checking which initial probabilities $\mathbb{P}_{t_0}(\boldsymbol{\gamma})$ and the associated derivatives are zero). The result is shown in Table C.1.

Let us illustrate how Prop.6.4.1 is applied to derive the initial values $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_2\right]$, $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_3\right]$ and $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_4\right]$. First, we consider state $S_2 \in D_z^{agg}$, and present the derivation of $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_2\right]$. Assuming that $\overrightarrow{\mathcal{A}}_s^*(\mathbb{M}) = \{b,c,d,e,f,g,h\}$, $\overrightarrow{\mathcal{A}}_{sl}^*(\mathbb{M}) = \{a\}$ and $\overrightarrow{\mathcal{A}}_l^*(\mathbb{M}) = \{q\}$, using Eq.(6.36) the conditional expectation equation constructed for S_2 is:

$$
\begin{aligned}
&\underbrace{\mathbb{P}_t(S_2) \cdot \frac{d\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right]\right)}{d\,t}}_{\text{I}} + \frac{d\left(\mathbb{P}_t(S_2)\right)}{d\,t} \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right] \approx \\
&- \underbrace{\mathbb{P}_t(S_2) \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right]\left(r_c + r_d\right)}_{\text{II}} \\
&+ \mathbb{P}_t(S_1) \cdot r_a \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_1\right] + \underbrace{\mathbb{P}_t(S_3) \cdot r_e \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_3\right]}_{\text{III}} + \underbrace{\mathbb{P}_t(S_4) \cdot r_h \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right]}_{\text{IV}} \\
&+ \mathbb{P}_t(S_1) \cdot r_a \cdot \mathcal{V}_a^l + \underbrace{\mathbb{P}_t(S_2) \cdot \mathcal{V}_q^l \cdot r_q \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right]}_{\text{V}}
\end{aligned}
\tag{6.44}
$$

Let us evaluate Eq.(6.44) at $t = t_0$ and simplify it using the information in Table C.1.

Here, $\mathbb{P}_{t_0}(S_2) = 0$. Therefore, in the left hand side, the term marked by (I) is zero. On the right hand side, the terms marked by (II) and (V) are zero. Moreover, we have: $\mathbb{P}_{t_0}(S_3) = 0$ and $\mathbb{P}_{t_0}(S_4) = 0$. Therefore, the terms marked by (III) and (IV) are zero. Removing these zero terms from the equation, we have:

$$\frac{d\left(\mathbb{P}_t(S_2)\right)}{d\,t}\Big|_{t_0} \cdot \mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_2\right] \approx \mathbb{P}_{t_0}(S_1)\cdot r_a \cdot \mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_1\right] + \mathbb{P}_{t_0}(S_1)\cdot r_a \cdot \mathcal{V}_a^l \quad (6.45)$$

Assuming that for $S_1 \in D_{nz}^{agg}$, the initial conditional expectation $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_1\right]$ is already known, in this equation, the only variable that is unknown is $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_2\right]$, which can be obtained by the reordering of the terms:

$$\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_2\right] \approx \frac{\mathbb{P}_{t_0}(S_1)\cdot r_a \cdot \left(\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_1\right] + \mathcal{V}_a^l\right)}{\frac{d\left(\mathbb{P}_t(S_2)\right)}{d\,t}\Big|_{t_0}} \quad (6.46)$$

As illustrated by Eq.(6.46), $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_2\right]$ with $\nabla_{S_2} = 1$ only depends on the conditional expectation of the states with distance $(\nabla_{S_2} - 1)$, that is S_1.

Now we focus on state S_3 which has the same distance $\nabla_{S_3} = 1 = \nabla_{S_2}$. We can follow a similar process for deriving $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_3\right]$. By setting up the equation capturing $\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_3\right]$, evaluating it at $t = t_0$ and removing the zero terms we derive:

$$\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_3\right] \approx \frac{\mathbb{P}_{t_0}(S_1)\cdot r_b \cdot \mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_1\right]}{\frac{d\left(\mathbb{P}_t(S_3)\right)}{d\,t}\Big|_{t_0}} \quad (6.47)$$

Again, $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_3\right]$ with $\nabla_{S_3} = 1$ only depends on the initial conditional expectations of the states within D_{nz}^{agg}.

Finally, we consider S_4 with distance $\nabla_{S_4} = 2$. For this state, the inward transitions are from states S_2 and S_3 with distances $\nabla_{S_2} = \nabla_{S_3} = 1$ and there is no such transition from D_{nz}^{agg} states. The equation capturing $\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right]$ is:

$$\begin{aligned}
&\mathbb{P}_t(S_4)\cdot\frac{d\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right]\right)}{d\,t} + \frac{d\left(\mathbb{P}_t(S_4)\right)}{d\,t}\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right] \approx \\
&\quad - \mathbb{P}_t(S_4)\cdot\left(r_h + r_g\right)\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right] + \mathbb{P}_t(S_2)\cdot r_d\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right] + \mathbb{P}_t(S_3)\cdot r_f\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_3\right] \\
&\quad + \mathbb{P}_t(S_4)\cdot\mathcal{V}_q^l\cdot r_q\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right] \quad (6.48)
\end{aligned}$$

For state S_4, due to Prop. 6.4.1, both $\mathbb{P}_{t_0}(S_4) = 0$ and $\frac{d\,\mathbb{P}_t(S_4)}{d\,t}\Big|_{t_0} = 0$ (See Table C.1). By evaluating Eq.(6.48) at $t = t_0$, we derive: $0 = 0$, indicating that this equation is not sufficient for deriving $\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l \,|\, S_4\Big]$. To resolve this issue, first, we differentiate Eq.(6.48) with respect to time once, and then evaluate it again at t_0. Taking the derivative of this equation we have:

$$\underbrace{\mathbb{P}_t(S_4)\cdot\frac{d^2\,\Big(\mathbb{E}_t\Big[\boldsymbol{\xi}^l \,|\, S_4\Big]\Big)}{d\,t^2}}_{\text{I}} + \underbrace{2\cdot\frac{d\,\Big(\mathbb{P}_t(S_4)\Big)}{d\,t}\cdot\frac{d\,\Big(\mathbb{E}_t\Big[\boldsymbol{\xi}^l \,|\, S_4\Big]\Big)}{d\,t}}_{\text{II}} + \frac{d^2\,\big(\mathbb{P}_t(S_4)\big)}{d\,t^2}\cdot\mathbb{E}_t\Big[\boldsymbol{\xi}^l \,|\, S_4\Big] \;\approx$$

$$-\Big(r_h + r_g\Big)\cdot\left(\underbrace{\frac{d\,\mathbb{P}_t(S_4)}{d\,t}\cdot\mathbb{E}_t\Big[\boldsymbol{\xi}^l \,|\, S_4\Big]}_{\text{III}} + \underbrace{\mathbb{P}_t(S_4)\cdot\frac{d\,\mathbb{E}_t\Big[\boldsymbol{\xi}^l \,|\, S_4\Big]}{d\,t}}_{\text{IV}}\right)$$

$$+ r_d\cdot\left(\frac{d\,\mathbb{P}_t(S_2)}{d\,t}\cdot\mathbb{E}_t\Big[\boldsymbol{\xi}^l \,|\, S_2\Big] + \underbrace{\mathbb{P}_t(S_2)\cdot\frac{d\,\mathbb{E}_t\Big[\boldsymbol{\xi}^l \,|\, S_2\Big]}{d\,t}}_{\text{V}}\right)$$

$$+ r_f\cdot\left(\frac{d\,\mathbb{P}_t(S_3)}{d\,t}\cdot\mathbb{E}_t\Big[\boldsymbol{\xi}^l \,|\, S_3\Big] + \underbrace{\mathbb{P}_t(S_3)\cdot\frac{d\,\mathbb{E}_t\Big[\boldsymbol{\xi}^l \,|\, S_3\Big]}{d\,t}}_{\text{VI}}\right)$$

$$+ \mathcal{V}_q^l\cdot r_q\cdot\left(\underbrace{\frac{d\,\mathbb{P}_t(S_4)}{d\,t}\cdot\mathbb{E}_t\Big[\boldsymbol{\xi}^l \,|\, S_4\Big]}_{\text{VII}} + \underbrace{\mathbb{P}_t(S_4)\cdot\frac{d\,\mathbb{E}_t\Big[\boldsymbol{\xi}^l \,|\, S_4\Big]}{d\,t}}_{\text{VIII}}\right) \tag{6.49}$$

Next, by evaluating Eq.(6.49) at t_0 and removing the terms that are zero (terms marked by $(\text{I}-\text{VIII})$), we derive:

$$\frac{d^2\,\big(\mathbb{P}_t(S_4)\big)}{d\,t^2}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l \,|\, S_4\Big] \;\approx\; r_d\cdot\frac{d\,\mathbb{P}_{t_0}(S_2)}{d\,t}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l \,|\, S_2\Big] + r_f\cdot\frac{d\,\mathbb{P}_t(S_3)}{d\,t}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l \,|\, S_3\Big] \tag{6.50}$$

In Eq.(6.50) and on the left hand side the initial value $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_4\right]$ is preserved. Assuming that $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_2\right]$ and $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_3\right]$ have been derived in the previous steps, the only unknown variable is $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_4\right]$. Thus:

$$\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_4\right] \approx \frac{r_d \cdot \frac{d\,\mathbb{P}_{t_0}(S_2)}{d\,t}\Big|_{t_0} \cdot \mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_2\right] + r_f \cdot \frac{d\,\mathbb{P}_t(S_3)}{d\,t}\Big|_{t_0} \cdot \mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_3\right]}{\frac{d^2\,(\mathbb{P}_t(S_4))}{d\,t^2}\Big|_{t_0}} \tag{6.51}$$

Note that in Eq(6.51), the initial conditional expectation $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, S_4\right]$ depends on the conditional expectations of only the states $\boldsymbol{\gamma}$ with $\nabla_{\boldsymbol{\gamma}} = (\nabla_{S_4} - 1)$, that is $\{S_2, S_3\}$. This was the last step in the derivation of the initial conditional expectations for our example.

6.4.3.2 Initial Conditional Expectations

Our observations in the context of the above example are generalised in the following proposition.

Proposition 6.4.2. *For any aggregated state $\boldsymbol{\gamma} \in D_z^{agg}$, with $\nabla_{\boldsymbol{\gamma}}$ as its initial distance, the $(\nabla_{\boldsymbol{\gamma}} - 1)$-th derivative of its associated conditional expectation equation is sufficient for calculating $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right]$. Moreover, by removing the terms that are zero in the differentiated equation, we derive:*

$$\mathbb{E}_t\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right] = \frac{\sum\limits_{\alpha \in \overrightarrow{\mathcal{A}}_s^*(\mathbb{M}) \cup \overrightarrow{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\,(\mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s))}{d\,t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0} \left(\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma} - \mathcal{V}_\alpha^s\right] + \mathcal{V}_\alpha^l\right)}{\boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+}, \nabla_{(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)} = \nabla_{\boldsymbol{\gamma}} - 1 \quad \frac{d^{(\nabla_{\boldsymbol{\gamma}})}\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{(\nabla_{\boldsymbol{\gamma}})}}\Big|_{t_0}} \tag{6.52}$$

Proof. The proof is presented in Appendix B.2.

□

Eq.(6.52) shows that one factor which $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right]$ depends on is $\frac{d^{\nabla_{\boldsymbol{\gamma}}}\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}$. This measure is obtained by differentiating the equation capturing the evolution of $\mathbb{P}_t(\boldsymbol{\gamma})$ up to the order $(\nabla_{\boldsymbol{\gamma}})$ and then evaluating at $t = 0$. Focusing on the aggregated states in consequence, we define the set Sol_∇^{agg}, which for each $\boldsymbol{\gamma} \in D^{agg}$ consists of the derivatives of $\mathbb{P}_{t_0}(\boldsymbol{\gamma})$ up to the order $\nabla_{\boldsymbol{\gamma}}$. Forming the set is a necessary step in finding the initial conditional expectations.

Prop. 6.4.2 also shows another important property regarding $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right]$; the initial value $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right]$ depends on the initial conditional expectations of only the states $\boldsymbol{\gamma}^{'} \in D^{agg}$ which have initial distances $\nabla_{\boldsymbol{\gamma}^{'}} = (\nabla_{\boldsymbol{\gamma}} - 1)$. The dependence on the states with *strictly less* distance means that the sequence in which the initial conditional expectations of D_z^{agg} states are derived depends on the structure of the aggregated state space and is specific to each problem. By taking into account this feature, we are able to propose a recursive algorithm that finds the initial values. The pseudo-code of this algorithm is presented in Alg.2. Assuming that an initial distribution is specified over the complete state space D and the solution Sol_{∇}^{agg} is obtained, this algorithm traverses through the states in D^{agg} and by taking into account their dependency, the associated initial conditional expectations are derived.

Algorithm 2 Given an aggregated state $\boldsymbol{\gamma} \in D^{agg}$, return $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right]$

function INIT_COND_EXP(aggregated_state $\boldsymbol{\gamma}$)
  $\mathbb{P}_{t_0}(D)$ // initial distribution over D ;
  Sol_{∇}^{agg} // solution of the analysis of D^{agg} ;
  if $(\boldsymbol{\gamma} \in D_{nz}^{agg})$ **then** // Base case
    $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right] \leftarrow$ derive $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right]$ using $\mathbb{P}_{t_0}(D)$; // Case 1
    return $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right]$;
  end if
  if $(\boldsymbol{\gamma} \in D_z^{agg})$ **then** // Case2
    $origin\_states \leftarrow$ set of $\boldsymbol{\gamma}^{'} \in D^{agg}$ that enable transition $\boldsymbol{\gamma}^{'} \rightarrow \boldsymbol{\gamma}$, $\nabla_{\boldsymbol{\gamma}^{'}} = \nabla_{\boldsymbol{\gamma}} - 1$;
    $origin\_cond\_expecs \leftarrow \{\}$;
    for $(\boldsymbol{\gamma}^{'} \in origin\_states)$ **do**
      $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}^{'}\right] \leftarrow$ INIT_COND_EXP($\boldsymbol{\gamma}^{'}$) ; // Recursive step
      append $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}^{'}\right]$ to $origin\_cond\_expecs$;
    end for
    $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right] \leftarrow$ apply Eq.(6.52) using $origin\_cond\_expecs$ and Sol^{agg} ;
    return $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\right]$;
  end if
end function

The presentation of this algorithm concludes the description of our technique for the derivation of the required initial conditional expectations. In the next section, we will illustrate how we derive the initial values related to the derivatives of these condi-

tional expectation.

6.4.3.3 The Derivatives of the Initial Conditional Expectations

The calculations in this section are similar to those in the previous one and therefore, here we provide less detail. First, we illustrate the technique in the context of our example and then a general proposition is proposed. In our example, consider state S_2 with distance $\nabla_{S_2} = 1$. Evaluating Eq.(6.44) at t_0 was sufficient to find $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid S_2\right]$, and no differentiation was needed. However, the evaluation of Eq.(6.44) does not allow us to obtain $\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|S_2\right]}{d\,t}\Big|_{t_0}$, since in the left hand side our initial derivative of interest has the coefficient zero. This is the result of the S_2 having a non-zero distance from D_{nz}^{agg} states. In order to resolve this issue, we differentiate Eq.(6.44) once with respect to time and then evaluate again. By the differentiation step, we get:

$$
\begin{aligned}
&\frac{d^2\ (\mathbb{P}_t(S_2))}{d\,t^2}\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right] + 2\cdot\frac{d\,\mathbb{P}_t(S_2)}{d\,t}\cdot\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right]}{d\,t} + \underbrace{\mathbb{P}_t(S_2)\cdot\frac{d^2\,\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right]\right)}{d\,t^2}}_{I} \approx \\
&\quad \left(r_d + r_c\right)\cdot\left(\frac{d\,\mathbb{P}_t(S_2)}{d\,t}\cdot\mathbb{E}_t\left[\boldsymbol{\gamma} \mid S_2\right] + \underbrace{\mathbb{P}_t(S_2)\cdot\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right]}{d\,t}}_{\text{II}}\right) \\
&+\ r_a\cdot\left(\frac{d\,\mathbb{P}_t(S_1)}{d\,t}\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_1\right] + \mathbb{P}_t(S_1)\cdot\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_1\right]}{d\,t}\right) \\
&+\ r_e\cdot\left(\frac{d\,\mathbb{P}_t(S_3)}{d\,t}\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_3\right] + \underbrace{\mathbb{P}_t(S_3)\cdot\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_3\right]}{d\,t}}_{\text{III}}\right) \\
&+\ r_h\cdot\left(\underbrace{\frac{d\,\mathbb{P}_t(S_4)}{d\,t}\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right]}_{\text{IV}} + \underbrace{\mathbb{P}_t(S_4)\cdot\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right]}{d\,t}}_{V}\right) \\
&+\ r_a\cdot\mathcal{V}_a^l\cdot\frac{d\,\mathbb{P}_t(S_1)}{d\,t} + \mathcal{V}_q^l\cdot r_q\cdot\left(\frac{d\,\mathbb{P}_t(S_2)}{d\,t}\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right] + \underbrace{\mathbb{P}_t(S_2)\cdot\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right]}{d\,t}}_{VI}\right)
\end{aligned}
\tag{6.53}
$$

By evaluating Eq.(6.53) at $t = t_0$ and using Table C.1, the terms marked by $(\mathrm{I}-\mathrm{VI})$ reduce to zero. Therefore, the equation is simplified to:

$$\frac{d^2\ (\mathbb{P}_t(S_2))}{d\,t^2}\Big|_{t_0} \cdot \mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l \,|\, S_2\Big] + 2 \cdot \frac{d\ \mathbb{P}_t(S_2)}{d\,t}\Big|_{t_0} \cdot \underline{\frac{d\ \mathbb{E}_t\Big[\boldsymbol{\xi}^l \mid S_2\Big]}{d\,t}\Big|_{t_0}} \approx$$

$$+ \Big(r_d + r_c\Big) \cdot \frac{d\ \mathbb{P}_t(S_2)}{d\,t}\Big|_{t_0} \cdot \mathbb{E}_{t_0}[\boldsymbol{\gamma} | \, S_2]$$

$$+ r_a \cdot \left(\frac{d\ \mathbb{P}_t(S_1)}{d\,t}\Big|_{t_0} \cdot \mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l \,|\, S_1\Big] \ + \ \mathbb{P}_{t_0}(S_1) \cdot \frac{d\ \mathbb{E}_t\Big[\boldsymbol{\xi}^l \mid S_1\Big]}{d\,t}\Big|_{t_0} \right)$$

$$+ r_e \cdot \frac{d\ \mathbb{P}_t(S_3)}{d\,t}\Big|_{t_0} \cdot \mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l \,|\, S_3\Big] + r_a \cdot \mathcal{V}_a^l \cdot \frac{d\ \mathbb{P}_t(S_1)}{d\,t}\Big|_{t_0} + \mathcal{V}_q^l \cdot r_q \cdot \frac{d\ \mathbb{P}_t(S_2)}{d\,t}\Big|_{t_0} \cdot \mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l \,|\, S_2\Big] \tag{6.54}$$

In Eq.(6.54), $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^l | S_2\right]}{d\,t}\Big|_{t_0}$ has a non-zero coefficient (see Table C.1). Moreover, $\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l \,|\, S_2\Big]$ was derived in the previous section, and for state S_1 the initial values $\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l \,|\, S_1\Big]$ and $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^l | S_1\right]}{d\,t}\Big|_{t_0}$ are calculated directly from the initial distribution. Therefore, in the equation above the only unknown variable is $\underline{\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^l | S_2\right]}{d\,t}\Big|_{t_0}}$ (underlined), which can be obtained by a simple reordering of the terms. Note here an important property about the structure of Eq.(6.54); for S_2 with $\nabla_{S_2} = 1$, the initial derivative $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^l | S_2\right]}{d\,t}\Big|_{t_0}$ depends on the initial derivative related to only the state $S_1 \in D_{nz}^{agg}$, with distance $\nabla_{S_1} = 0$.

A similar process is applied for state S_3 with $\nabla_{S_3} = \nabla_{S_2} = 1$. By differentiating the equation that captures $\mathbb{E}_t\Big[\boldsymbol{\xi}^l \mid S_3\Big]$ once and evaluating at t_0, the initial value $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^l | S_3\right]}{d\,t}\Big|_{t_0}$ is obtained. Due to its similarity, we skip the details of this calculation.

Finally, we consider S_4. Here, the calculations are more complicated. The equation used for calculating $\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l \,|\, S_4\Big]$ is Eq.(6.50), which itself is the result of differentiating the equation capturing $\mathbb{E}_t\Big[\boldsymbol{\xi}^l \mid S_4\Big]$, i.e. Eq.(6.49), once with respect to time. Using Eq.(6.50), the initial value $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^l | S_4\right]}{d\,t}\Big|_{t_0}$ cannot be obtained, since in the left hand side the corresponding coefficients are zero. This is the result of $\nabla_{S_4} = 2$; for $k = 0, 1 : \frac{d^k\ (\mathbb{P}_t(S_4))}{d\,t^k}\Big|_{t_0} = 0$. Here, unlike the states S_2 and S_3, a single differentiation of

the conditional expectation equation is not sufficient. To resolve the issue, the *second* derivative of Eq.(6.49) is constructed and evaluated at t_0. By differentiating Eq.(6.49) twice, we get:

$$\underbrace{\mathbb{P}_t(S_4)\frac{d^3\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right]\right)}{d\,t^3}}_{\text{I}} + \underbrace{3\frac{d\,\mathbb{P}_t(S_4)}{d\,t}\frac{d^2\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right]\right)}{d\,t^2}}_{\text{II}} +$$

$$3\frac{d^2\,(\mathbb{P}_t(S_4))}{d\,t^2}\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right]}{d\,t} + \frac{d^3\,(\mathbb{P}_t(S_4))}{d\,t^3}\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right] \approx$$

$$-\left(r_h+r_g\right)\left(\frac{d^2\,(\mathbb{P}_t(S_4))}{d\,t^2}\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right] + \underbrace{2\frac{d\,\mathbb{P}_t(S_4)}{d\,t}\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right]}{d\,t}}_{\text{III}} + \underbrace{\mathbb{P}_t(S_4)\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right]}{d\,t}}_{\text{IV}}\right)$$

$$+r_d\cdot\left(\frac{d^2\,(\mathbb{P}_t(S_2))}{d\,t^2}\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right] + 2\frac{d\,\mathbb{P}_t(S_2)}{d\,t}\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right]}{d\,t} + \underbrace{\mathbb{P}_t(S_2)\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right]}{d\,t}}_{\text{V}}\right)$$

$$+r_f\cdot\left(\frac{d^2\,(\mathbb{P}_t(S_3))}{d\,t^2}\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_3\right] + 2\frac{d\,\mathbb{P}_t(S_3)}{d\,t}\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_3\right]}{d\,t} + \underbrace{\mathbb{P}_t(S_3)\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_3\right]}{d\,t}}_{\text{VI}}\right)$$

$$+\mathcal{V}_q^l\cdot r_q\cdot\left(\frac{d^2\,(\mathbb{P}_t(S_4))}{d\,t^2}\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right] + \underbrace{2\frac{d\,\mathbb{P}_t(S_4)}{d\,t}\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right]}{d\,t}}_{\text{VII}} + \underbrace{\mathbb{P}_t(S_4)\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right]}{d\,t}}_{\text{VIII}}\right) \tag{6.55}$$

Next, by evaluating the above equation at t_0 and using Prop. 6.4.1 to remove the terms that reduce to zero, we get:

$$3\cdot\frac{d^2\,(\mathbb{P}_t(S_4))}{d\,t^2}\Big|_{t_0}\cdot\underline{\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_4\right]}{d\,t}\Big|_{t_0}} + \frac{d^3\,(\mathbb{P}_t(S_4))}{d\,t^3}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid S_4\right] \approx$$

$$-\left(r_h+r_g\right)\left(\frac{d^2\,(\mathbb{P}_t(S_4))}{d\,t^2}\Big|_{t_0}\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid S_4\right]\right) + \mathcal{V}_q^l\cdot r_q\cdot\left(\frac{d^2\,(\mathbb{P}_t(S_4))}{d\,t^2}\Big|_{t_0}\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid S_4\right]\right)$$

$$+r_d\cdot\left(\frac{d^2\,(\mathbb{P}_t(S_2))}{d\,t^2}\Big|_{t_0}\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid S_2\right] + 2\frac{d\,\mathbb{P}_t(S_2)}{d\,t}\Big|_{t_0}\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_2\right]}{d\,t}\Big|_{t_0}\right)$$

$$+r_f\cdot\left(\frac{d^2\,(\mathbb{P}_t(S_3))}{d\,t^2}\Big|_{t_0}\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid S_3\right] + 2\frac{d\,\mathbb{P}_t(S_3)}{d\,t}\Big|_{t_0}\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid S_3\right]}{d\,t}\Big|_{t_0}\right) \tag{6.56}$$

In the previous steps, for the states S_2 and S_3 the initial conditional expectations $\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\Big]$, $\boldsymbol{\gamma} \in \{S_2, S_3\}$ and their initial derivatives $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\ t}\Big|_{t_0}$ have been derived. In Eq.(6.56) the only unknown variable is $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^l|S_4\right]}{d\ t}\Big|_{t_0}$, which can be obtained by re-ordering of the terms. Again, note the important structural property that the equation above exhibits. Given that $\nabla_{S_4} = 2$, the related initial value $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^l|S_4\right]}{d\ t}\Big|_{t_0}$ depends on the initial derivative of only the states with $(\nabla_{S_4} - 1)$ distance, which are S_2 and S_3.

Having described our technique in the context of the example, now we present the following proposition which enables us to calculate the initial derivatives of the conditional expectations for states $\boldsymbol{\gamma} \in D_z^{agg}$.

Proposition 6.4.3. *For any aggregated state $\boldsymbol{\gamma} \in D_z^{agg}$ with initial distance $\nabla_{\boldsymbol{\gamma}}$, the $\nabla_{\boldsymbol{\gamma}}$-th derivative of the equation that captures the evolution of $\mathbb{E}_t\Big[\boldsymbol{\xi}^l \,|\, \boldsymbol{\gamma}\Big]$ is sufficient to calculate $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\ t}\Big|_{t_0}$. Moreover, by removing the terms that reduce to zero in the differentiated equation, we obtain Eq.(6.57) where $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\ t}\Big|_{t_0}$ can be obtained by re-ordering of the terms.*

$$
\begin{aligned}
(\nabla_{\boldsymbol{\gamma}}+1)\cdot\frac{d^{\nabla_{\boldsymbol{\gamma}}}\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\ t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}\cdot\underline{\frac{d\ \mathbb{E}_t\Big[\boldsymbol{\xi}^l\ |\ \boldsymbol{\gamma}\Big]}{d\ t}\Big|_{t_0}} \;+\; \frac{d^{(\nabla_{\boldsymbol{\gamma}}+1)}\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\ t^{(\nabla_{\boldsymbol{\gamma}}+1)}}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}\Big] \;=\; & \\
-\sum_{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma})\cdot\frac{d^{\nabla_{\boldsymbol{\gamma}}}\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\ t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}\Big] & \qquad I \\
+\sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}\\ \nabla_{(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)}<\nabla_{\boldsymbol{\gamma}}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\Bigg(\frac{d^{(\nabla_{\boldsymbol{\gamma}})}\ (\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s))}{d\ t^{(\nabla_{\boldsymbol{\gamma}})}}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}-\mathcal{V}_\alpha^s\Big] \;+ & \qquad II \\
\nabla_{\boldsymbol{\gamma}}\cdot\frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\ (\mathbb{P}(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s))}{d\ t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0}\cdot\frac{d\ \mathbb{E}_t\Big[\boldsymbol{\xi}^l\ |\ \boldsymbol{\gamma}-\mathcal{V}_\alpha^s\Big]}{d\ t}\Big|_{t_0}\Bigg) & \qquad III \\
+\sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}\\ \nabla_{(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)}=\nabla_{\boldsymbol{\gamma}}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\Bigg(\frac{d^{(\nabla_{\boldsymbol{\gamma}})}\ (\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s))}{d\ t^{(\nabla_{\boldsymbol{\gamma}})}}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}-\mathcal{V}_\alpha^s\Big] & \qquad IV
\end{aligned}
$$

$$+ \sum_{\substack{\alpha \in \vec{\mathcal{A}}^{*}_{sl}(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}^{s,+}_{\alpha} \\ \nabla_{(\boldsymbol{\gamma}-\mathcal{V}^{s}_{\alpha})} \leq \nabla_{\boldsymbol{\gamma}}-1}} r_{\alpha}(\boldsymbol{\gamma}-\mathcal{V}^{s}_{\alpha}) \cdot \mathcal{V}^{l}_{\alpha} \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}})} \ (\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}^{s}_{\alpha}))}{d\, t^{(\nabla_{\boldsymbol{\gamma}})}}\Big|_{t_0}$$

$$+ \sum_{\alpha \in \vec{\mathcal{A}}^{*}_{l}(\mathbb{M})} \mathcal{V}^{l}_{\alpha} \cdot \frac{d^{\nabla_{\boldsymbol{\gamma}}} \ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\, t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0} \cdot r_{\alpha}\Big(\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^{l} \,|\, \boldsymbol{\gamma}\Big]\Big) \qquad V \qquad (6.57)$$

Proof. The proof is presented in Appendix B.3. □

Let us briefly focus on the terms in Eq.(6.57).

By focusing on the left hand side, we observe that one factor which $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^{l}|\boldsymbol{\gamma}\right]}{d\ t}\Big|_{t_0}$ depends on is $\frac{d^{(\nabla_{\boldsymbol{\gamma}}+1)}\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\ t^{(\nabla_{\boldsymbol{\gamma}}+1)}}\Big|_{t_0}$. Similarly to the case for $\mathbb{E}_{t_0}\Big[\boldsymbol{\xi}^{l} \,|\, \boldsymbol{\gamma}\Big]$, this measure is obtained by differentiating the equation capturing the evolution of $\mathbb{P}_t(\boldsymbol{\gamma})$ and then evaluating at $t = 0$. By performing the derivation for the aggregated states in sequence, we obtain the set $Sol^{agg}_{(\nabla+1)}$, which for each $\boldsymbol{\gamma} \in D^{agg}$ consists of the derivatives of $\mathbb{P}_{t_0}(\boldsymbol{\gamma})$ up to the order $(\nabla_{\boldsymbol{\gamma}}+1)$. This set is then used for finding the derivatives of the initial conditional expectations.

In Eq.(6.57), the expressions marked by (I), (II), (IV) and (V) consist of initial conditional expectations, which are already known using Prop. 6.4.2. Moreover, the term (III) consists of the derivative of initial conditional expectations for states $\boldsymbol{\gamma}' \in D^{agg}$ which enable transitions into $\boldsymbol{\gamma}$ and have initial distances *strictly less* than $\boldsymbol{\gamma}$. We can assume that these initial derivatives have already been obtained using a similar version of Eq.(6.57). By these considerations, we conclude that in Eq.(6.57) the only unknown variable is $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^{l}|\boldsymbol{\gamma}\right]}{d\ t}\Big|_{t_0}$ and the equation is sufficient for calculating this initial value.

Similarly to the previous section, the important structural property that Eq.(6.57) exhibits allows us to propose an iterative algorithm, similar to Alg. 2, which progressively finds the initial derivatives $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^{l}|\boldsymbol{\gamma}\right]}{d\ t}\Big|_{t_0}$, for all $\boldsymbol{\gamma} \in D^{agg}$. The pseudo code for this algorithm is shown in Alg. 3.

Algorithm 3 Given $\boldsymbol{\gamma} \in D^{agg}$, return $\left.\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\,t}\right|_{t_0}$.

function INIT_COND_EXP_DER(aggregated_state $\boldsymbol{\gamma}$)
  $\mathbb{P}_{t_0}(D)$ // initial distribution over D ;
  $Sol^{agg}_{\nabla+1}$;
  if $(\boldsymbol{\gamma} \in D^{agg}_{nz})$ **then** // Base case
    $\left.\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\,t}\right|_{t_0} \leftarrow$ derive $\left.\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\,t}\right|_{t_0}$ using $\mathbb{P}_{t_0}(D)$; // Case 1
    return $\left.\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\,t}\right|_{t_0}$;
  end if
  if $(\boldsymbol{\gamma} \in D^{agg}_{z})$ **then** // Case2
    $origin\_states \leftarrow$ states $\boldsymbol{\gamma}' \in D^{agg}$ that enable transition $\boldsymbol{\gamma}' \rightarrow \boldsymbol{\gamma}$, $\nabla_{\boldsymbol{\gamma}'} = \nabla_{\boldsymbol{\gamma}} - 1$;
    $origin\_cond\_expecs \leftarrow \{\}$; // required init. cond. expectations
    $origin\_cond\_expecs\_der \leftarrow \{\}$; // required cond. expec. derivatives
    $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}\right] \leftarrow$ INIT_COND_EXP($\boldsymbol{\gamma}$) ;
    append $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}\right]$ to $origin\_cond\_expecs$;
    for $(\boldsymbol{\gamma}' \in origin\_states)$ **do**
      $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}'\right] \leftarrow$ INIT_COND_EXP($\boldsymbol{\gamma}'$) ; // Recursive step
      append $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}'\right]$ to $origin\_con\_expecs$;
      if $\nabla_{\boldsymbol{\gamma}'} < \nabla_{\boldsymbol{\gamma}}$ **then**
        $\left.\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}'\right]}{d\,t}\right|_{t_0} \leftarrow$ INIT_COND_EXP_DER($\boldsymbol{\gamma}'$) ;
        append $\left.\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}'\right]}{d\,t}\right|_{t_0}$ to $origin\_cond\_expecs\_der$;
      end if
    end for
    $\left.\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\,t}\right|_{t_0} \leftarrow$ apply Eq.(B.23) using Sol^{agg}, $origin\_cond\_expecs$ and $origin\_cond\_expecs\_der$;
    return $\left.\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\boldsymbol{\gamma}\right]}{d\,t}\right|_{t_0}$;
  end if
end function

Chapter 7

Analysis of Client-Server Model Using Conditional Expectations

7.1 Introduction

In this chapter, we apply the analysis by conditional expectations to a version of the client-server system. For convenience, the model and some of its description is repeated here. Table 7.1 shows the parameters used. Here, the two servers are serving 150 clients and given the parameters, the servers are continually performing under heavy contention.

$$
\begin{aligned}
C_{think} &\stackrel{def}{=} (think, r_t).C_{req} \\
C_{req} &\stackrel{def}{=} (req, \top).C_{think} \\
\\
S_{idle} &\stackrel{def}{=} (req, r_s).S_{log} + (brk, r_b).S_{broken} \\
S_{log} &\stackrel{def}{=} (log, r_l).S_{idle} \\
S_{broken} &\stackrel{def}{=} (fix, r_f).S_{idle} \\
\\
CS &\stackrel{def}{=} Servers\ \{\ S_{idle}[n_s]\ \} \underset{\{req\}}{\bowtie} Clients\ \{\ C_{think}[n_c]\ \}
\end{aligned}
$$

parameter	r_s	r_l	r_f	r_b	r_t	n_s	n_c
values	120	240	0.001	0.0001	1.7	2	150

Table 7.1: The parameters of the client-server system.

We use the state vector $\boldsymbol{\xi} = \langle S_i, S_l, S_b, C_t, C_r \rangle$ to capture the state of the model and D_{CS} to represent the model's state space. The model has two groups; $\mathcal{G}(CS) = \{Servers, Clients\}$. Given the total number of servers and clients, the group *Servers* is regarded to be small and the group *Clients* as large. Therefore, the partition $\Delta_{\mathcal{G}} = \{\{Servers\}, \{Clients\}\}$ is defined over the set of the groups of the model $\mathcal{G}(CS) = \{Servers, Clients\}$. Accordingly, the state vector is partitioned: $\boldsymbol{\xi} = \langle \boldsymbol{\xi}^s, \boldsymbol{\xi}^l \rangle$ where $\boldsymbol{\xi}^s = \langle S_i, S_l, S_b \rangle$ and $\boldsymbol{\xi}^l = \langle C_t, C_r \rangle$.

As the initial state of the system, we assume that at $t_0 = 0$, the system starts at the state $\langle S_i, S_l, S_b, C_t, C_r \rangle = \langle 2, 0, 0, 150, 0 \rangle$ where both servers are idle and all the clients are in state C_{think}. This means that at t_0, a probability distribution is defined over the system's complete state space as:

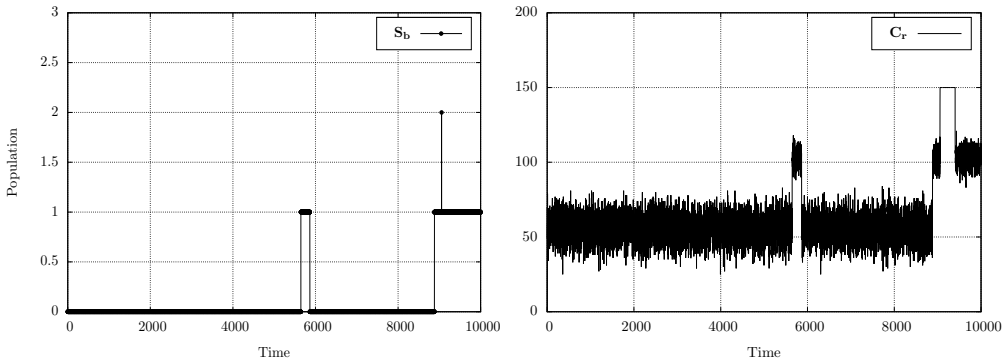
$$\mathbb{P}_{t_0}(s_i, s_l, s_b, c_t, c_r) = \begin{cases} 1 & \text{if } \langle s_i, s_l, s_b, c_t, c_r \rangle = \langle 2, 0, 0, 150, 0 \rangle \\ 0 & \text{otherwise} \end{cases} \tag{7.1}$$

In this client-server system the stochastic behaviour of the servers has significant impact on the behaviour of the clients. This is illustrated in Fig. 7.1, which shows two trajectories of the system with respect to state variables S_b and C_r. The dependence of C_r on S_b is also illustrated in Fig. 7.2, which shows the steady state distributions associated with these state variables. The system has three distinct modes of behaviour associated with $S_b = 0$, $S_b = 1$ and $S_b = 2$, and depending on the number of servers currently broken, the variable C_r clusters around distinct values.

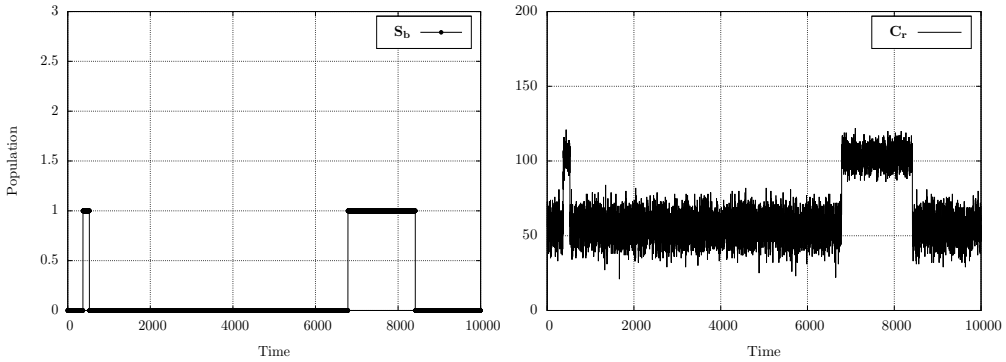
7.2 Aggregation Method

The model satisfies the conditions required for the aggregation method we presented in Chap. 4. First, there is a heavy load on the servers and the probability of being in states where no client requires service (boundary states) is close to zero. Second, the model respects the syntactic aggregation condition; the clients undertake activity *req* passively. The aggregated model captures the evolution of the servers only and its state space is illustrated in Fig. 7.3. This aggregation relies on the partition $\Delta_{\mathcal{A}} = \{\overrightarrow{\mathcal{A}}^*_s(CS), \overrightarrow{\mathcal{A}}^*_{sl}(CS), \overrightarrow{\mathcal{A}}^*_l(CS)\}$ defined over the set of actions of the model $\overrightarrow{\mathcal{A}}^*(CS)$, where $\overrightarrow{\mathcal{A}}^*_s(CS) = \{log, brk, fix\}$, $\overrightarrow{\mathcal{A}}^*_{sl}(CS) = \{req\}$ and $\overrightarrow{\mathcal{A}}^*_l(CS) = \{think\}$.

Let D^{agg}_{CS} denote the aggregated state space of Fig. 7.3. By the analysis of D^{agg} we derive a probability distribution over the states of the servers. Let $\mathbb{P}_t(\boldsymbol{\beta})$, $\boldsymbol{\beta} \in D^{agg}_{CS}$

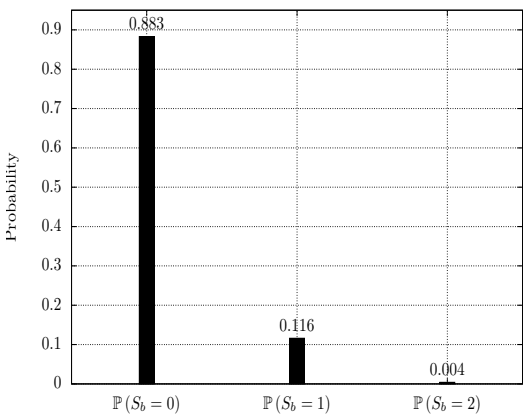


(A) Evolution of S_b in first trajectory. (B) Evolution of C_r in first trajectory.

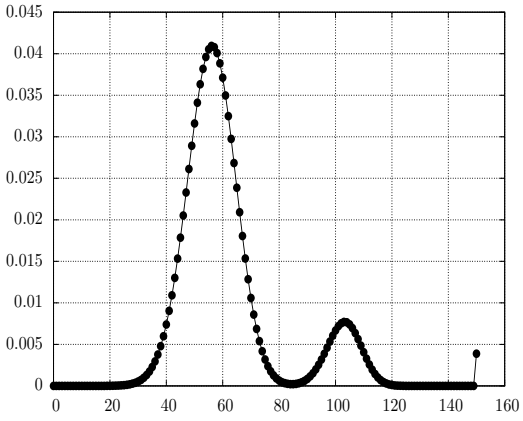


(C) Evolution of S_b in second trajectory. (D) Evolution of C_r in second trajectory.

Figure 7.1: Two individual trajectories of the system with respect to S_b and C_r derived using stochastic simulation. The result of the event that a server breaks down is the lower service rate experienced by the clients and the increase in the variable C_r.



(A) Probability distribution of S_b at equilibrium.



(B) Probability distribution of C_r at equilibrium.

Figure 7.2: The probability distribution for two state variables C_r, S_b derived using the analysis of complete state space. Depending on the available number of servers, C_r clusters around different values.

denote this distribution; at any given time t, $\mathbb{P}_t(\boldsymbol{\beta}), \boldsymbol{\beta} = \langle x_1, x_2, x_3 \rangle \in D^{agg}$, denotes the probability that at t, $S_i = x_1$, $S_l = x_2$ and $S_b = x_3$. This distribution is the solution of the Chapman-Kolmogorov equations underlying D^{agg}_{CS}.

Let us focus on two representative states $\langle 1,1,0 \rangle$ and $\langle 1,0,1 \rangle$. The ODEs associated with these two states are presented in Eq.(7.2) and Eq.(7.3).

$$\begin{aligned}\frac{d\,\mathbb{P}_t(\langle 1,1,0\rangle)}{d\,t} = & -\mathbb{P}_t(\langle 1,1,0\rangle) \cdot \Big(r_{req}(\langle 1,1,0\rangle) + r_{brk}(\langle 1,1,0\rangle) + r_{log}(\langle 1,1,0\rangle)\Big) \\ & + r_{req}(\langle 2,0,0\rangle) \cdot \mathbb{P}_t(\langle 2,0,0\rangle) + r_{log}(\langle 0,2,0\rangle) \cdot \mathbb{P}_t(\langle 0,2,0\rangle) \\ & + r_{fix}(\langle 0,1,1\rangle) \cdot \mathbb{P}_t(\langle 0,1,1\rangle) \end{aligned} \tag{7.2}$$

$$\begin{aligned}\frac{d\,\mathbb{P}_t(\langle 1,0,1\rangle)}{d\,t} = & -\mathbb{P}_t(\langle 1,0,1\rangle) \cdot \Big(r_{req}(\langle 1,0,1\rangle) + r_{brk}(\langle 1,0,1\rangle) + r_{fix}(\langle 1,0,1\rangle)\Big) \\ & + r_{brk}(\langle 2,0,0\rangle) \cdot \mathbb{P}_t(\langle 2,0,0\rangle) + r_{log}(\langle 0,1,1\rangle) \cdot \mathbb{P}_t(\langle 0,1,1\rangle) \\ & + r_{fix}(\langle 0,0,2\rangle) \cdot \mathbb{P}_t(\langle 0,0,2\rangle) \end{aligned} \tag{7.3}$$

To solve the complete set of equations, an initial distribution over D^{agg}_{CS} is needed.

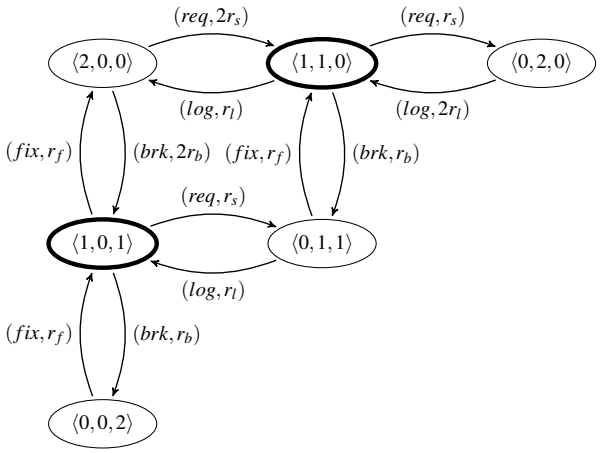


Figure 7.3: The aggregated state space of the client-server system with two servers. Abstracting away from the clients, this captures the configurations that the servers experience. Note that the representative states are highlighted.

This can be extracted from Eq.(7.1):

$$\forall\ \boldsymbol{\beta} \in D_{CS}^{agg}\ ,\ \mathbb{P}_{t_0}(\boldsymbol{\beta}) = \begin{cases} 1 & \text{if} \quad \boldsymbol{\beta} = \langle 2,0,0 \rangle \\ 0 & \text{otherwise} \end{cases} \tag{7.4}$$

Using the underlying C-K equations of D_{CS}^{agg} and the initial distribution above, the solution is obtained. The projection of this solution with respect to the representative states, i.e. the time evolution of $\mathbb{P}_t(\langle 1,1,0 \rangle)$ and $\mathbb{P}_t(\langle 1,0,1 \rangle)$ is shown in Fig. 7.4.

Using the aggregation, all configurations that the servers experience are captured. A probability distribution is obtained over such configurations, which can be used to understand the randomness caused by the servers and any aberrations from their mean evolution. For example, we can derive the distribution of S_b, shown in Fig. 7.2A, to measure the probability of the servers being in different modes of operation including the configuration where both servers are broken, or the impact of changing their parameters on their throughput.

7.3 Conditional Expectations

Now, we now focus on capturing the impact of the servers' randomness on the clients. Using the MCM, the conditional expectation of $\langle C_t, C_r \rangle$ is derived across the states $\boldsymbol{\beta} \in D_{CS}^{agg}$. Since D_{CS}^{agg} contains six states and $\boldsymbol{\xi}^l = \langle C_t, C_r \rangle$ has two state variables,

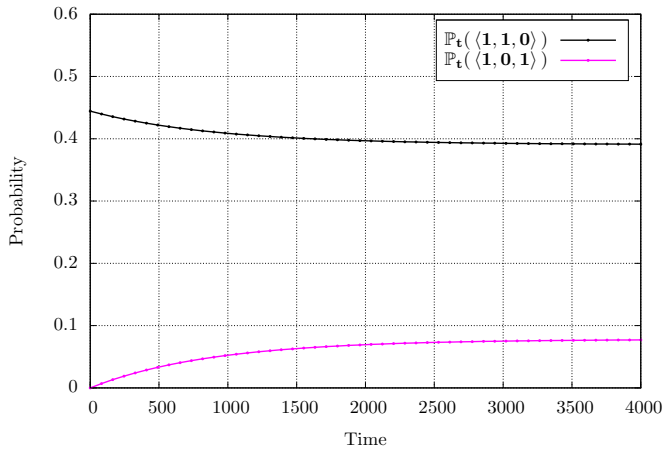


Figure 7.4: The evolution of $\mathbb{P}_t(\langle 1,1,0\rangle)$ and $\mathbb{P}_t(\langle 1,0,1\rangle)$, derived by analysing D_{CS}^{agg}.

$\boldsymbol{\beta} \in D_{CS}^{agg}$	C_t	C_r
$\langle 2,0,0\rangle$	$\mathbb{E}[C_t \mid \langle 2,0,0\rangle]$	$\mathbb{E}[C_r \mid \langle 2,0,0\rangle]$
$\langle 1,1,0\rangle$	$\mathbb{E}[C_t \mid \langle 1,1,0\rangle]$	$\mathbb{E}[C_r \mid \langle 1,1,0\rangle]$
$\langle 0,2,0\rangle$	$\mathbb{E}[C_t \mid \langle 0,2,0\rangle]$	$\mathbb{E}[C_r \mid \langle 0,2,0\rangle]$
$\langle 1,0,1\rangle$	$\mathbb{E}[C_t \mid \langle 1,0,1\rangle]$	$\mathbb{E}[C_r \mid \langle 1,0,1\rangle]$
$\langle 0,1,1\rangle$	$\mathbb{E}[C_t \mid \langle 0,1,1\rangle]$	$\mathbb{E}[C_r \mid \langle 0,1,1\rangle]$
$\langle 0,0,2\rangle$	$\mathbb{E}[C_t \mid \langle 0,0,2\rangle]$	$\mathbb{E}[C_r \mid \langle 0,0,2\rangle]$

Table 7.2: The conditional expectation variables related to the large group *Clients* in the client-server model.

there are 12 conditional expectations $\mathbb{E}_t[\xi \mid \boldsymbol{\beta}]$, $\boldsymbol{\beta} \in D^{agg}$, $\xi \in \langle C_t, C_r\rangle$, to be calculated. These expectations are presented in Table. 7.2. To find their evolution, a system of DAEs is constructed where for each $\mathbb{E}_t[\xi \mid \boldsymbol{\beta}]$, one equation is formed with the structure shown in Eq.(6.39). The equations in this system rely on the probability distribution $\mathbb{P}_t(\boldsymbol{\beta})$ derived over D_{CS}^{agg}. Therefore, the modeller must have obtained $\mathbb{P}_t(\boldsymbol{\beta})$ before solving the MCM equations.

Before presenting the DAEs, two pieces of notation are reviewed. First, we present the impact vectors associated with the actions defined in the model. These vectors are shown in Table 7.3. Second, we review the apparent rate functions of these actions. These are presented in Table 7.4, the second column. Note that, as illustrated in Eq.(6.39), one factor which the evolution of conditional expectations depends on is the conditional expectation of these rate functions, which can be obtained using Lemma 6.3.2. The result of applying this Lemma to the rate functions is shown in Table 7.4,

	Impact vector : $\mathcal{V}_\alpha$				
	$\mathcal{V}_\alpha^s$			$\mathcal{V}_\alpha^l$	
Action α	S_i	S_l	S_b	C_t	C_r
req	-1	$+1$	0	$+1$	-1
log	$+1$	-1	0	0	0
brk	-1	0	$+1$	0	0
fix	$+1$	0	-1	0	0
think	0	0	0	-1	$+1$

Table 7.3: Impact vectors of the actions defined in the client-server model.

Action	Rate function	Conditional expectation of rate function
α	$r_\alpha(S), S = \langle S_i, S_l, S_b, C_t, C_r \rangle$	$\mathbb{E}[r_\alpha(S_i) \mid \boldsymbol{\beta}]$, $\boldsymbol{\beta} = \langle S_i, S_l, S_b \rangle$
req	$r_s \times S_i$	$\mathbb{E}[(r_s \times S_i) \mid \boldsymbol{\beta}] = r_s \times \mathbb{E}[S_i \mid \boldsymbol{\beta}]$
log	$r_l \times S_l$	$\mathbb{E}[(r_l \times S_l) \mid \boldsymbol{\beta}] = r_l \times \mathbb{E}[S_l \mid \boldsymbol{\beta}]$
brk	$r_b \times S_i$	$\mathbb{E}[(r_b \times S_i) \mid \boldsymbol{\beta}] = r_b \times \mathbb{E}[S_i \mid \boldsymbol{\beta}]$
fix	$r_f \times S_i$	$\mathbb{E}\big[(r_f \times S_b) \mid \boldsymbol{\beta}\big] = r_f \times \mathbb{E}[S_b \mid \boldsymbol{\beta}]$
think	$r_t \times C_t$	$\mathbb{E}[(r_t \times C_t) \mid \boldsymbol{\beta}] = r_t \times \mathbb{E}[C_t \mid \boldsymbol{\beta}]$

Table 7.4: The apparent rate functions and their conditional expectations.

the third column.

As stated, for our client-server model, the system of DAEs consists of 12 equations. The complete list of equations is presented in Appendix C.1. Here, again we focus only on the representative states $\boldsymbol{\beta} = \langle 1,1,0 \rangle$ and $\boldsymbol{\beta} = \langle 1,0,1 \rangle$ and show their equations related to $\mathbb{E}_t\,[\,\langle C_t, C_r \rangle \mid \langle 1,1,0 \rangle\,]$ and $\mathbb{E}_t\,[\,\langle C_t, C_r \rangle \mid \langle 1,0,1 \rangle\,]$ in the vector form. The equations for other $\boldsymbol{\beta} \in D_{CS}^{agg}$ are similarly constructed. For $\boldsymbol{\beta} = \langle 1,1,0 \rangle$, the equa-

tion capturing the evolution of $\mathbb{E}_t\,[\,\langle C_t, C_r\rangle \mid \langle 1,1,0\rangle\,]$ is:

$$\mathbb{P}_t(\langle 1,1,0\rangle)\cdot\frac{d\,\left(\,\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle\,]\,\right)}{d\,t}\;+\;\frac{d\,\mathbb{P}_t(\langle 1,1,0\rangle)}{d\,t}\cdot\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle\,]\;\approx$$

$$\begin{aligned}
&-\;\mathbb{P}_t(\langle 1,1,0\rangle)\cdot\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle\,]\cdot\Big[r_{log}(\langle 1,1,0\rangle)+r_{brk}(\langle 1,1,0\rangle)+r_{req}(\langle 1,1,0\rangle)\Big]\\
&+\;\mathbb{P}_t(\langle 2,0,0\rangle)\cdot r_{req}(\langle 2,0,0\rangle)\cdot\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 2,0,0\rangle\,]\\
&+\;\mathbb{P}_t(\langle 0,1,1\rangle)\cdot r_{fix}(\langle 0,1,1\rangle)\cdot\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 0,1,1\rangle\,]\\
&+\;\mathbb{P}_t(\langle 0,2,0\rangle)\cdot r_{log}(\langle 0,2,0\rangle)\cdot\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 0,2,0\rangle\,]\\
&+\;\mathbb{P}_t(\langle 2,0,0\rangle)\cdot r_{req}(\langle 2,0,0\rangle)\cdot\mathcal{V}^{l}_{req}\\
&+\;\mathbb{P}_t(\langle 1,1,0\rangle)\cdot\mathcal{V}^{l}_{think}\cdot r_{req}\Big(\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle\,]\Big)
\end{aligned}\tag{7.5}$$

For $\boldsymbol{\beta}=\langle 1,0,1\rangle$, the evolution of $\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 1,0,1\rangle\,]$, is captured by:

$$\mathbb{P}_t(\langle 1,0,1\rangle)\cdot\frac{d\,\left(\,\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 1,0,1\rangle\,]\,\right)}{d\,t}\;+\;\frac{d\,\mathbb{P}_t(\langle 1,0,1\rangle)}{d\,t}\cdot\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 1,0,1\rangle\,]\;\approx$$

$$\begin{aligned}
&-\;\mathbb{P}_t(\langle 1,0,1\rangle)\cdot\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 1,0,1\rangle\,]\cdot\Big[r_{brk}(\langle 1,0,1\rangle)+r_{fix}(\langle 1,0,1\rangle)+r_{req}(\langle 1,0,1\rangle)\Big]\\
&+\;\mathbb{P}_t(\langle 2,0,0\rangle)\cdot r_{brk}(\langle 2,0,0\rangle)\cdot\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 2,0,0\rangle\,]\\
&+\;\mathbb{P}_t(\langle 0,0,2\rangle)\cdot r_{fix}(\langle 0,0,2\rangle)\cdot\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 0,0,2\rangle\,]\\
&+\;\mathbb{P}_t(\langle 0,1,1\rangle)\cdot r_{log}(\langle 0,1,1\rangle)\cdot\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 0,1,1\rangle\,]\\
&+\;\mathbb{P}_t(\langle 1,0,1\rangle)\cdot\mathcal{V}^{l}_{think}\cdot r_{think}(\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \mid \langle 1,0,1\rangle\,])
\end{aligned}\tag{7.6}$$

7.4 Construction of Initial Values

The model's underlying DAEs are treated as an initial value problem and solving the equations requires a set of initial values. These consists of the initial conditional expectations $\mathbb{E}_{t_0}[\,\langle C_t,C_r\rangle\,|\,\boldsymbol{\beta}]$, $\boldsymbol{\beta}\in D^{agg}_{CS}$ and the derivatives $\frac{d\;\mathbb{E}_t[\,\langle C_t,C_r\rangle|\boldsymbol{\beta}]}{d\;t}\Big|_{t_0}$.

As illustrated in Sec. 6.4, the derivation of the initial values depends on the initial distribution defined over the aggregated state space. Using the initial distribution of Eq.(7.4), the set of aggregated states with *non-zero* initial probability consists of

only one state $D_{nz}^{agg} = \{\langle 2,0,0\rangle\}$. For this state, the initial conditional expectation and its derivative are calculated directly from the initial distribution. Conversely, the set of states with *zero* initial probability is $D_z^{agg} = \{\langle 1,1,0\rangle, \langle 0,2,0\rangle, \langle 1,0,1\rangle, \langle 0,1,1\rangle, \langle 0,0,2\rangle\}$. For these states, their initial conditions are derived using Def. 30, Prop. 6.4.1, Prop. 6.4.2 and Prop. 6.4.3. The description for these calculations follows.

Initial distance from probability sources. Using the initial distribution presented in Eq.(7.4), Def. 30 is used to assign an initial distance from D_{nz}^{agg} states to each aggregated state present in D_{CS}^{agg}. Such distances are shown in Table 7.5, the second column. We can see that the state $\langle 2,0,0\rangle$ has distance zero, the states $\langle 1,1,0\rangle$ and $\langle 1,0,1\rangle$ have distance one and $\langle 0,2,0\rangle$, $\langle 0,1,1\rangle$ and $\langle 0,0,2\rangle$ have distance two.

In Sec. 6.4.3.3, we observed that when calculating the initial values, for each aggregated state $\boldsymbol{\beta} \in D_{CS}^{agg}$ with initial distance $\nabla_{\boldsymbol{\beta}}$ the first $(\nabla_{\boldsymbol{\beta}}+1)$-th derivatives of its probability function, i.e. $\frac{d^k\ (\mathbb{P}_t(\boldsymbol{\beta}))}{d\ t^k}\Big|_{t_0}$, $k = 0,1,\cdot,(\nabla_{\boldsymbol{\beta}}+1)$, are required. Such derivatives can be obtained by repeated differentiation of the C-K equations underlying D_{CS}^{agg}. A series of equations that were constructed for this purpose are presented in Appendix C.2 and the result is shown in Table 7.5, in the second to the fourth column.

States in D_{nz}^{agg}. For the only state $\langle 2,0,0\rangle \in D_z^{agg}$, we find $\mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \langle 2,0,0\rangle\,]$ directly from the initial distribution using Bayes' Law:

$$\begin{aligned}\mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \langle 2,0,0\rangle\,] &= \sum \langle C_t, C_r\rangle \cdot \mathbb{P}_{t_0}(\,\langle C_t, C_r\rangle \mid \langle 2,0,0\rangle\,) \\ &= \sum \langle C_t, C_r\rangle \cdot \frac{\mathbb{P}_{t_0}(\langle 2,0,0, C_t, C_r\rangle)}{\mathbb{P}_{t_0}(\langle 2,0,0\rangle)} = [150,0] \qquad (7.7)\end{aligned}$$

Furthermore, $\frac{d\ \mathbb{E}_t[\,\langle C_t, C_r\rangle | \langle 2,0,0\rangle\,]}{d\ t}\Big|_{t_0}$ is derived by substituting $\mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \langle 2,0,0\rangle\,]$ as above into the equation that captures the evolution of $\mathbb{E}_t\,[\,\langle C_t, C_r\rangle \mid \langle 2,0,0\rangle\,]$) and evaluating at t_0. By this substitution, we have: $\frac{d\ \mathbb{E}_t[\,\langle C_t, C_r\rangle | \langle 2,0,0\rangle\,]}{d\ t}\Big|_{t_0} = 150 r_t \cdot \mathcal{V}_{think}^{l}$.

States in D_z^{agg} - initial conditional expectations Prop. 6.4.2 shows that for each aggregated state $\boldsymbol{\beta} \in D_z^{agg}$ with the initial distance $\nabla_{\boldsymbol{\beta}}$, the initial condition $\mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \boldsymbol{\beta}\,]$ can be obtained by differentiating the equation that captures $\mathbb{E}_t\,[\,\langle C_t, C_r\rangle \mid \boldsymbol{\beta}\,]$ up to the order $(\nabla_{\boldsymbol{\beta}} - 1)$ and evaluating the result at t_0. Using the proposition, for the states $\langle 1,1,0\rangle$ and $\langle 1,0,1\rangle$ with distance one, no differentiation is needed and the initial values $\mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \langle 1,1,0\rangle\,]$ and $\mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \langle 1,0,1\rangle\,]$ are obtained by evaluating their respective conditional expectation equations at t_0. The result is $\mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \langle 1,1,0\rangle\,] = \langle 151,-1\rangle$ and $\mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \langle 1,0,1\rangle\,] = \langle 150,0\rangle$. For the states $\langle 0,2,0\rangle$, $\langle 0,1,1\rangle$ and $\langle 0,0,2\rangle$, their equations, i.e. Eq.(C.3), Eq(C.4) and Eq.(C.5) – (in Appendix C.2) –

$\boldsymbol{\beta} \in D_{CS}^{agg}$	$\nabla_{\boldsymbol{\beta}}$	$\mathbb{P}_{t_0}(\boldsymbol{\beta})$	$\frac{d\ \mathbb{P}_t(\boldsymbol{\beta})}{d\ t}\Big\vert_{t_0}$	$\frac{d^2\ (\mathbb{P}_t(\boldsymbol{\beta}))}{d\ t^2}\Big\vert_{t_0}$	$\frac{d^3\ (\mathbb{P}_t(\boldsymbol{\beta}))}{d\ t^3}\Big\vert_{t_0}$
$\langle 2,0,0\rangle$	0	1	$-2(r_s+r_b)$	-	-
$\langle 1,1,0\rangle$	1	0	$2r_s$	$-2r_s(3r_s+3r_b+r_l)$	-
$\langle 0,2,0\rangle$	2	0	0	$2r_s^2$	$-6r_s^2(r_s+r_b+r_l)$
$\langle 1,0,1\rangle$	1	0	$2r_b$	$-2r_b(3r_s+3r_b+r_f)$	-
$\langle 0,1,1\rangle$	2	0	0	$4r_sr_b$	$-6r_sr_b(2r_s+2r_b+r_l+r_f)$
$\langle 0,0,2\rangle$	2	0	0	$2r_b^2$	$-6r_b^2(r_s+r_b+r_f)$

Table 7.5: The derivatives of $\mathbb{P}_t(\boldsymbol{\gamma}), \boldsymbol{\gamma} \in D^{agg}$ at $t = t_0$ obtained by applying Prop.6.4.1 and repeated differentiation of C-K equations of the aggregated state space.

$\boldsymbol{\beta} \in D_{CS}^{agg}$	$\mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \boldsymbol{\beta}\,]$	$\frac{d\ \mathbb{E}_t[\langle C_t,C_r\rangle \vert \boldsymbol{\beta}]}{d\ t}\Big\vert_{t_0}$
$\langle 2,0,0\rangle$	$\langle 150,0\rangle$	$150r_t \cdot \mathcal{V}_{think}^{l}$
$\langle 1,1,0\rangle$	$\langle 151,-1\rangle$	$(r_s+r_b)\cdot \mathcal{V}_{req}^{l} + (150.5r_t)\cdot \mathcal{V}_{think}^{l}$
$\langle 0,2,0\rangle$	$\langle 152,-2\rangle$	$\frac{(8r_s+8r_b+2r_l)}{3}\cdot \mathcal{V}_{req}^{l} + 151r_t \cdot \mathcal{V}_{think}^{l}$
$\langle 1,0,1\rangle$	$\langle 150,0\rangle$	$150r_t \cdot \mathcal{V}_{think}^{l}$
$\langle 0,1,1\rangle$	$\langle 151,-1\rangle$	$\frac{(r_s+r_b)}{3}\cdot \mathcal{V}_{req}^{l} + \frac{301r_t}{3}\cdot \mathcal{V}_{think}^{l}$
$\langle 0,0,2\rangle$	$\langle 150,0\rangle$	$150r_t \cdot \mathcal{V}_{think}^{l}$

Table 7.6: Initial values for conditional expectations and their derivatives.

are differentiated once ($(\nabla_{\langle 0,2,0\rangle}-1) = (\nabla_{\langle 0,1,1\rangle}-1) = (\nabla_{\langle 0,0,2\rangle}-1=1)$) and the resulting equations are evaluated at t_0. For these states we get: $\mathbb{E}_{t_0}[\,\langle C_t,C_r\rangle \,|\, \langle 0,2,0\rangle\,] = \langle 152,-2\rangle$, $\mathbb{E}_{t_0}[\,\langle C_t,C_r\rangle \,|\, \langle 0,1,1\rangle\,] = \langle 151,-1\rangle$ and $\mathbb{E}_{t_0}[\,\langle C_t,C_r\rangle \,|\, \langle 0,0,2\rangle\,] = \langle 150,0\rangle$. The initial conditional expectations calculated for this model are summarised in Table 7.6, the second column.

States in D_z^{agg}, the derivative of the initial conditional expectations. For the states in D_z^{agg}, we use Prop. 6.4.3 to find the derivative of their initial conditions. According to this proposition, for any state $\boldsymbol{\beta} \in D_z^{agg}$ with distance $\nabla_{\boldsymbol{\beta}}$, in order to derive $\frac{d\ \mathbb{E}_t[\langle C_t,C_r\rangle \vert \boldsymbol{\beta}]}{d\ t}$ we differentiate the equation capturing $\mathbb{E}_t\,[\,\langle C_t,C_r\rangle \,|\, \boldsymbol{\beta}\,]$ up to the order $\nabla_{\boldsymbol{\beta}}$ and evaluate the resulting equation at t_0. The states $\langle 1,1,0\rangle$ and $\langle 1,0,1\rangle$ have distance one. Therefore, by differentiating Eq.(C.2) and Eq.(C.3) once and evaluating at t_0, we have: $\frac{d\ \mathbb{E}_t[\langle C_t,C_r\rangle \vert \langle 1,1,0\rangle]}{d\ t}\Big\vert_{t_0} = (r_s+r_b)\cdot \mathcal{V}_{req}^{l} + (150.5r_t)\cdot \mathcal{V}_{think}^{l}$ and $\frac{d\ \mathbb{E}_t[\langle C_t,C_r\rangle \vert \langle 1,0,1\rangle]}{d\ t}\Big\vert_{t_0} = 150r_t \cdot \mathcal{V}_{think}^{l}$. Note that these initial derivatives depend on the initial values $\mathbb{E}_{t_0}[\,\langle C_t,C_r\rangle \,|\, \boldsymbol{\beta}\,], \boldsymbol{\beta} \in \{\langle 2,0,0\rangle, \langle 1,1,0\rangle, \langle 1,0,1\rangle\}$ and $\frac{d\ \mathbb{E}_t[\langle C_t,C_r\rangle \vert \langle 2,0,0\rangle]}{d\ t}\Big\vert_{t_0}$, which have previously been calculated. For the states $\langle 0,2,0\rangle$, $\langle 0,1,1\rangle$ and $\langle 0,0,2\rangle$ with distances $\nabla_{\langle 0,2,0\rangle} = \nabla_{\langle 0,1,1\rangle} = \nabla_{\langle 0,0,2\rangle} = 2$, their conditional expectation equations Eq.(C.3), Eq.(C.4) and Eq.(C.5) are differentiated to the second order and evaluated at t_0 to yield: $\frac{d\ \mathbb{E}_t[\langle C_t,C_r\rangle \vert \langle 0,2,0\rangle]}{d\ t}\Big\vert_{t_0}$, $\frac{d\ \mathbb{E}_t[\langle C_t,C_r\rangle \vert \langle 0,1,1\rangle]}{d\ t}\Big\vert_{t_0}$ and $\frac{d\ \mathbb{E}_t[\langle C_t,C_r\rangle \vert \langle 0,0,2\rangle]}{d\ t}\Big\vert_{t_0}$. We summarise these initial values in Table 7.6, the third column.

Solution. To solve the set of DAEs, we used the Matlab Suite's ode solver. This solver is sufficiently powerful to deal with a large number of equations and provides parameters to adjust the accuracy to the accepted level. The solution, with respect to the representative states $\langle 1,1,0\rangle$ and $\langle 1,0,1\rangle$ is shown in Fig.7.5.

	Conditional expectation	MCM	PRISM	Error (%)
$\langle S_i, S_l, S_b \rangle = \langle 1,1,0 \rangle$	$\mathbb{E}[C_t \mid \langle 1,1,0 \rangle]$	94.336	94.867	0.55
	$\mathbb{E}[C_r \mid \langle 1,1,0 \rangle]$	55.664	55.979	0.56
$\langle S_i, S_l, S_b \rangle = \langle 1,0,1 \rangle$	$\mathbb{E}[C_t \mid \langle 1,0,1 \rangle]$	46.863	46.576	0.61
	$\mathbb{E}[C_r \mid \langle 1,0,1 \rangle]$	103.14	102.51	0.61

Table 7.7: Comparison between the steady state values of $\mathbb{E}_t[\langle C_t, C_r \rangle \mid \langle 1,1,0 \rangle]$ and $\mathbb{E}_t[\langle C_t, C_r \rangle \mid \langle 1,0,1 \rangle]$ derived from the MCM and PRISM. The error is acceptably low.

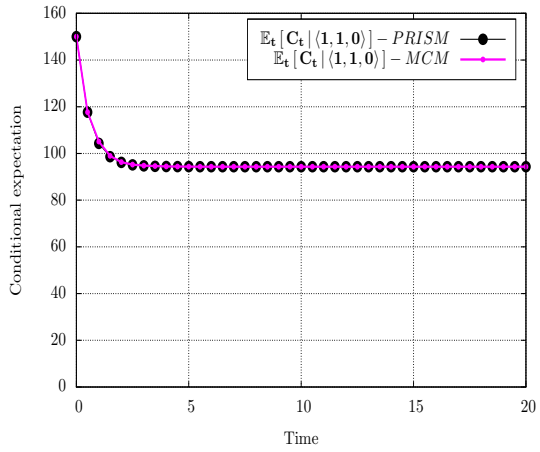
In order to check the accuracy of our MCM approximation, we compared the MCM solution against that obtained from the exact analysis of the model using PRISM. The comparison shows the MCM approximation is highly accurate. We also compared the approximate steady state conditional expectations obtained from MCM and those derived from PRISM. The result of this comparison and the relative error for the representative states is reported in Table. 7.7. The comparison showed that across all $\boldsymbol{\beta} \in D_{CS}^{agg}$ the error is less than 1 percent.

7.5 Capturing the Effect of Server Breakdown

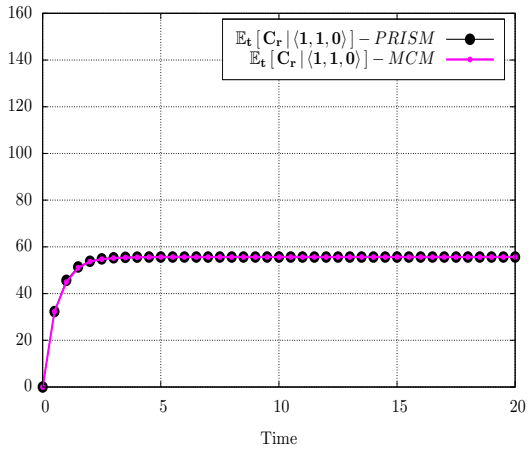
The distribution $\mathbb{P}_t(\boldsymbol{\beta})$, $\boldsymbol{\beta} \in D_{CS}^{agg}$, and the conditional expectations $\mathbb{E}[\langle C_t, C_r \rangle \mid \boldsymbol{\beta}]$ across these $\boldsymbol{\beta}$ enable us to capture the impact of the fluctuations of the servers on the clients. Here we aim to obtain the conditional expectation of $\langle C_t, C_r \rangle$ *given the servers' different modes of operation*. These different modes are shown in Fig. 7.6. Essentially, using $\mathbb{P}_t(\boldsymbol{\beta})$ and $\mathbb{E}[\langle C_t, C_r \rangle \mid \boldsymbol{\beta}]$ we derive the conditional expectations $\mathbb{E}_t[\langle C_t, C_r \rangle \mid S_b = 0]$, $\mathbb{E}_t[\langle C_t, C_r \rangle \mid S_b = 1]$ and $\mathbb{E}_t[\langle C_t, C_r \rangle \mid S_b = 2]$ and the MCM conditional expectations are sufficient for this purpose.

The client-server system of this example satisfies the time-scale near-complete decomposability (TSND) condition. The transitions of type *req*, *log* and *think* occur on the fast time scale and those of types *brk* and *fix* happen over the slow time scale. The modes are related to the subsets of servers' configurations that are separated by slow transitions. Although here the modes were detected by eye, they could also be automatically detected using the enhanced TSND detection algorithm that we presented in Chapter 5.

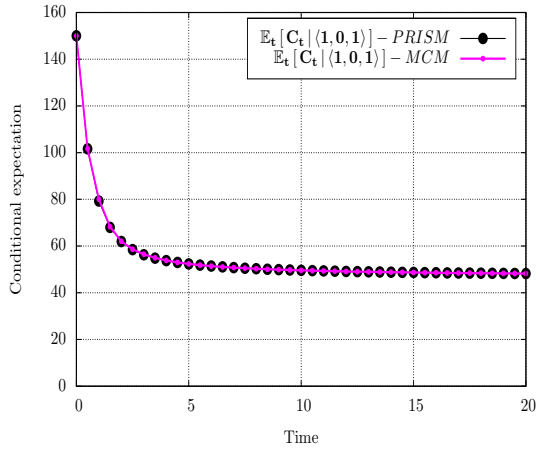
Indeed, the TSND detection algorithm can be fruitfully combined with the MCM



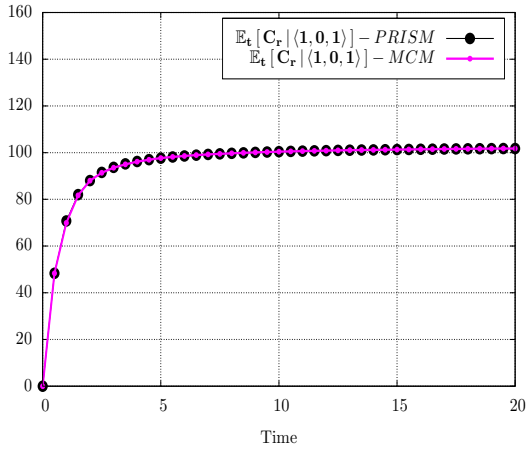
(A) Conditional expectation of C_t in aggregated state $\langle 1,1,0 \rangle$.



(B) Conditional expectation of C_r in aggregated state $\langle 1,1,0 \rangle$.



(C) Conditional expectation of C_r in aggregated state $\langle 1,0,1 \rangle$.



(D) Conditional expectation of C_r in aggregated state $\langle 1,0,1 \rangle$.

Figure 7.5: Time evolution of the conditional expectation of $\langle C_t, C_r \rangle$ across two representative aggregated states $\langle 1,1,0 \rangle$ and $\langle 1,0,1 \rangle$, as derived by PRISM and MCM.

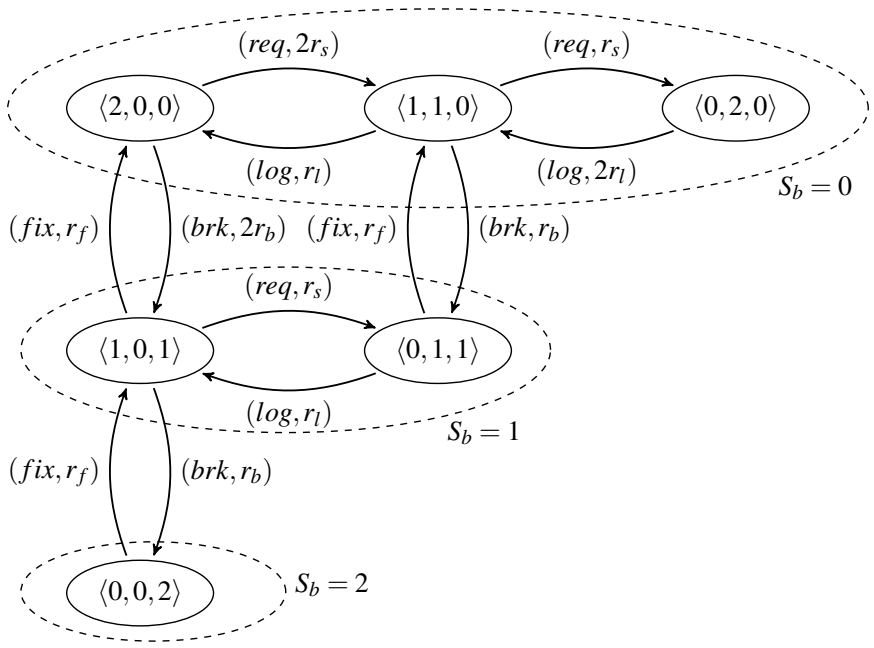


Figure 7.6: The aggregated state space of the client-server system with two servers. The different modes of operation associated with the different number of active servers are highlighted.

for the derivation of the modes for which the conditional moments are calculated. The TSND algorithm outputs the slow / fast partition over the model's actions, which induces disjoint sub-sets of aggregated states (i.e. modes) that are separated by the slow (and potentially significant) transitions of the model's resources. The set of modes are fed into the MCM, which then calculates the expectation (and higher-order moments) of the quasi-steady-state behaviour of the resource users for each mode. This way, the model is comprehensively analysed for both short and long term phenomena; on the one hand, the possibility of having mode switches is accounted for by incorporating the resources' state space (aggregated state space), and on the other, we are able to capture the behaviour of resource users for any mode of resources that if visited, the system stays there for a relatively long time. The TSND detection algorithm is capable of dealing with any large and complex model, and can provide the significant operational modes automatically.

We now return to the client-server system. The mode of operation where $S_b = 0$ corresponds to the states $\langle 2,0,0\rangle$, $\langle 1,1,0\rangle$ and $\langle 0,2,0\rangle$. The measures, $\mathbb{P}_t(\langle 2,0,0\rangle)$, $\mathbb{P}_t(\langle 1,1,0\rangle)$, $\mathbb{P}_t(\langle 0,2,0\rangle)$ (derived by the analysis of D_{CS}^{agg}) and $\mathbb{E}_t[\langle C_t, C_r\rangle \mid \langle 2,0,0\rangle]$, $\mathbb{E}_t[\langle C_t, C_r\rangle \mid \langle 1,1,0\rangle]$ and $\mathbb{E}_t[\langle C_t, C_r\rangle \mid \langle 0,2,0\rangle]$, (derived by the MCM equations) are

sufficient for derivation of the conditional expectation $\mathbb{E}_t[\langle C_t, C_r\rangle \mid S_b = 0]$.

By Def.28 and using Bayes' Law, $\mathbb{E}_t[\langle C_t, C_r\rangle \mid S_b = 0]$ is expanded as:

$$\begin{aligned}
\mathbb{E}_t[\langle C_t, C_r\rangle \mid S_b = 0] &= \sum_{\langle S_i,S_l,S_b,C_t,C_r\rangle \in D} \langle C_t, C_r\rangle \cdot \mathbb{P}_t(\langle S_i, S_l, S_b, C_t, C_r\rangle \mid S_b = 0) \\
&= \sum_{\langle S_i,S_l,0,C_t,C_r\rangle \in D} \langle C_t, C_r\rangle \cdot \frac{\mathbb{P}_t(\langle S_i, S_l, 0, C_t, C_r\rangle)}{\mathbb{P}_t(S_b = 0)} \\
&= \frac{\sum_{\langle S_i,S_l,0,C_t,C_r\rangle \in D} \langle C_t, C_r\rangle \cdot \mathbb{P}_t(\langle S_i, S_l, 0, C_t, C_r\rangle)}{\mathbb{P}_t(S_b = 0)}
\end{aligned}$$

In the summation of the last line, we sum over the states in the original state space where $S_b = 0$. Since the model satisfies the aggregation condition, its state space can be divided into a number of sub-chains each of which is represented as a single state in the aggregation state space. Here, the sub-chains where $S_b = 0$ are $\mathbb{Y}_{\langle 2,0,0\rangle}$, $\mathbb{Y}_{\langle 1,1,0\rangle}$ and $\mathbb{Y}_{\langle 0,2,0\rangle}$, which are represented by aggregated states $\langle 2,0,0\rangle, \langle 1,1,0\rangle, \langle 0,2,0\rangle \in D_{CS}^{agg}$. We will expand the summation above so that the states within each sub-chain are summed separately:

$$\begin{aligned}
\mathbb{E}_t[\langle C_t, C_r\rangle \mid S_b = 0] &= \frac{\sum_{\langle 2,0,0,C_t,C_r\rangle \in \mathbb{Y}_{\langle 2,0,0\rangle}} \langle C_t, C_r\rangle \cdot \mathbb{P}_t(\langle 2, 0, 0, C_t, C_r\rangle)}{\mathbb{P}_t(S_b = 0)} \\
&+ \frac{\sum_{\langle 1,1,0,C_t,C_r\rangle \in \mathbb{Y}_{\langle 1,1,0\rangle}} \langle C_t, C_r\rangle \cdot \mathbb{P}_t(\langle 1, 1, 0, C_t, C_r\rangle)}{\mathbb{P}_t(S_b = 0)} \\
&+ \frac{\sum_{\langle 0,2,0,C_t,C_r\rangle \in \mathbb{Y}_{\langle 0,2,0\rangle}} \langle C_t, C_r\rangle \cdot \mathbb{P}_t(\langle 0, 2, 0, C_t, C_r\rangle)}{\mathbb{P}_t(S_b = 0)}
\end{aligned} \tag{7.8}$$

Using Bayes's Law, for each state $\langle S_i, S_l, S_b, C_t, C_r\rangle \in \mathbb{Y}_{\langle S_i,S_l,S_b\rangle}$ we can transform its probability term $\mathbb{P}(\langle S_i, S_l, S_b, C_t, C_r\rangle)$ to a conditional form as:

$$\mathbb{P}(\langle S_i, S_l, S_b, C_t, C_r\rangle) = \mathbb{P}(\langle C_t, C_r\rangle \mid \langle S_i, S_l, S_b\rangle) \cdot \mathbb{P}(\langle S_i, S_l, S_b\rangle) \tag{7.9}$$

We apply this transformation to Eq.(7.8) to derive:

$$\begin{aligned}
\mathbb{E}_t\left[\langle C_t, C_r\rangle \mid S_b = 0\right] &= \frac{\sum_{\langle 2,0,0,C_t,C_r\rangle \in \mathbb{Y}_{\langle 2,0,0\rangle}} \langle C_t, C_r\rangle \cdot \mathbb{P}_t(\langle C_t, C_r\rangle \mid \langle 2,0,0\rangle) \cdot \mathbb{P}_t(\langle 2,0,0\rangle)}{\mathbb{P}_t(S_b = 0)} \\
&+ \frac{\sum_{\langle 1,1,0,C_t,C_r\rangle \in \mathbb{Y}_{\langle 1,1,0\rangle}} \langle C_t, C_r\rangle \cdot \mathbb{P}_t(\langle C_t, C_r\rangle \mid \langle 1,1,0\rangle) \cdot \mathbb{P}_t(\langle 1,1,0\rangle)}{\mathbb{P}_t(S_b = 0)} \\
&+ \frac{\sum_{\langle 0,2,0,C_t,C_r\rangle \in \mathbb{Y}_{\langle 0,2,0\rangle}} \langle C_t, C_r\rangle \cdot \mathbb{P}_t(\langle C_t, C_r\rangle \mid \langle 0,2,0\rangle) \cdot \mathbb{P}_t(\langle 0,2,0\rangle)}{\mathbb{P}_t(S_b = 0)}
\end{aligned}$$

By factoring out the probability terms related to the aggregated states we have:

$$\begin{aligned}
\mathbb{E}_t\left[\langle C_t, C_r\rangle \mid S_b = 0\right] &= \frac{\mathbb{P}_t(\langle 2,0,0\rangle) \cdot \sum_{\langle 2,0,0,C_t,C_r\rangle \in \mathbb{Y}_{\langle 2,0,0\rangle}} \langle C_t, C_r\rangle \cdot \mathbb{P}_t(\langle C_t, C_r\rangle \mid \langle 2,0,0\rangle)}{\mathbb{P}_t(S_b = 0)} \\
&+ \frac{\mathbb{P}_t(\langle 1,1,0\rangle) \cdot \sum_{\langle 1,1,0,C_t,C_r\rangle \in \mathbb{Y}_{\langle 1,1,0\rangle}} \langle C_t, C_r\rangle \cdot \mathbb{P}_t(\langle C_t, C_r\rangle \mid \langle 1,1,0\rangle)}{\mathbb{P}_t(S_b = 0)} \\
&+ \frac{\mathbb{P}_t(\langle 0,2,0\rangle) \cdot \sum_{\langle 0,2,0,C_t,C_r\rangle \in \mathbb{Y}_{\langle 0,2,0\rangle}} \langle C_t, C_r\rangle \cdot \mathbb{P}_t(\langle C_t, C_r\rangle \mid \langle 0,2,0\rangle)}{\mathbb{P}_t(S_b = 0)}
\end{aligned} \tag{7.10}$$

In our final transformation, we use Def 28 to form the conditional expectation terms. According to this definition, for any $\boldsymbol{\beta} \in D_{CS}^{agg}$:

$$\sum_{\substack{\langle S_i,S_l,S_b,C_t,C_r\rangle \in \mathbb{Y}_{\boldsymbol{\beta}} \\ \langle S_i,S_l,S_b\rangle = \boldsymbol{\beta}}} \langle C_t, C_r\rangle \cdot \mathbb{P}_t(\langle C_t, C_r\rangle \mid \boldsymbol{\beta}) = \mathbb{E}_t\left[\langle C_t, C_r\rangle \mid \boldsymbol{\beta}\right] \tag{7.11}$$

Therefore, Eq.(7.10) is transformed into:

$$\begin{aligned}
\mathbb{E}_t\left[\langle C_t, C_r\rangle \mid S_b = 0\right] &= \frac{\mathbb{P}_t(\langle 2,0,0\rangle) \cdot \mathbb{E}_t\left[\langle C_t, C_r\rangle \mid \langle 2,0,0\rangle\right]}{\mathbb{P}_t(S_b = 0)} \\
&+ \frac{\mathbb{P}_t(\langle 1,1,0\rangle) \cdot \mathbb{E}_t\left[\langle C_t, C_r\rangle \mid \langle 1,1,0\rangle\right]}{\mathbb{P}_t(S_b = 0)} + \frac{\mathbb{P}_t(\langle 0,2,0\rangle) \cdot \mathbb{E}_t\left[\langle C_t, C_r\rangle \mid \langle 0,2,0\rangle\right]}{\mathbb{P}_t(S_b = 0)}
\end{aligned} \tag{7.12}$$

This equation is closed in terms of the probability distribution over D_{CS}^{agg} and the MCM conditional expectations. By following a similar process for other operational modes

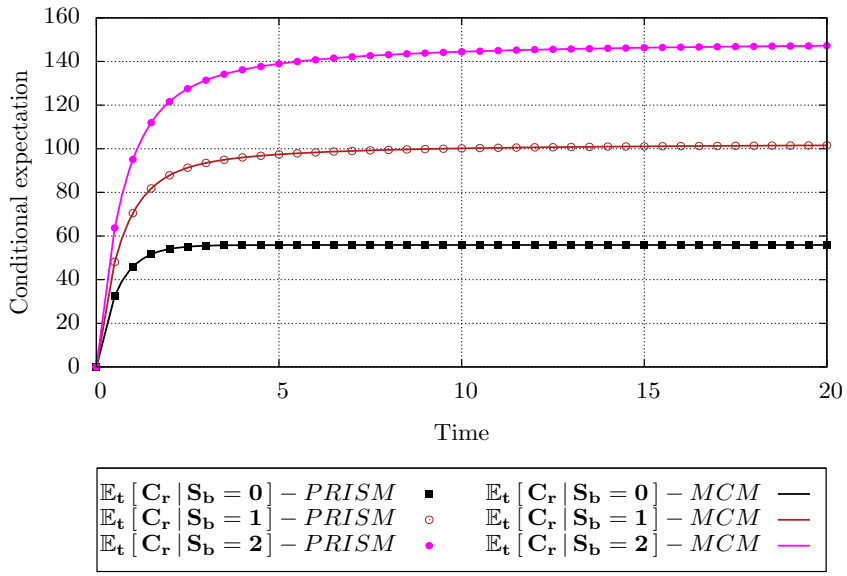


Figure 7.7: Conditional expectations of C_r across different values of S_b, as derived from PRISM and MCM.

($S_b = 1$ and $S_b = 2$), equations for $\mathbb{E}[\langle C_t, C_r\rangle \mid S_b = 1]$, $\mathbb{E}[\langle C_t, C_r\rangle \mid S_b = 2]$ are derived:

$$\begin{aligned}\mathbb{E}_t[\langle C_t, C_r\rangle \mid S_b = 1] &= \frac{\mathbb{P}_t(\langle 1,0,1\rangle)\cdot \mathbb{E}_t[\langle C_t, C_r\rangle \mid \langle 1,0,1\rangle]}{\mathbb{P}_t(S_b = 1)} \\ &+ \frac{\mathbb{P}_t(\langle 0,1,1\rangle)\cdot \mathbb{E}_t[\langle C_t, C_r\rangle \mid \langle 0,1,1\rangle]}{\mathbb{P}_t(S_b = 1)} \end{aligned} \tag{7.13}$$

$$\begin{aligned}\mathbb{E}_t[\langle C_t, C_r\rangle \mid S_b = 2] &= \frac{\mathbb{P}_t(\langle 0,0,2\rangle)\cdot \mathbb{E}_t[\langle C_t, C_r\rangle \mid \langle 0,0,2\rangle]}{\mathbb{P}_t(S_b = 2)} \\ &= \frac{\mathbb{P}_t(\langle 0,0,2\rangle)\cdot \mathbb{E}_t[\langle C_t, C_r\rangle \mid \langle 0,0,2\rangle]}{\mathbb{P}_t(\langle 0,0,2\rangle)} \end{aligned} \tag{7.14}$$

The solution of the equations (7.12), (7.13) and (7.14), projected on the state variable C_r is shown in Fig.7.7. Similarly to the previous case, to check the solution's accuracy a comparison was made against the same result obtained from PRISM. The measurements of the error are shown in Table 7.8. This comparison shows that $\mathbb{E}[C_r \mid S_b = n], n = 0,1,2$ derived from the MCM solution is highly accurate.

The conditional expectations $\mathbb{E}[C_r \mid S_b = \cdot]$ shed light on important qualitative and quantitative aspects of the client-server system's behaviour. Since the client's behaviour across the servers' different modes of operation is obtained, we can qualitatively verify the sensitivity of the client's behaviour to the servers' breakdown and recovery. Note that, such a sensitivity cannot be detected by methods such as fluid flow

	Conditional expectation	MCM	PRISM	Error (%)
$S_b = 0$	$\mathbb{E}[C_t \mid \langle 1,1,0\rangle]$	94.114	94.35	0.25
	$\mathbb{E}[C_r \mid \langle 1,1,0\rangle]$	55.886	56.026	0.24
$S_b = 1$	$\mathbb{E}[C_t \mid \langle 1,0,1\rangle]$	47.084	46.892	0.40
	$\mathbb{E}[C_r \mid \langle 1,0,1\rangle]$	102.92	102.5	0.40
$S_b = 2$	$\mathbb{E}[C_t \mid \langle 0,0,2\rangle]$	0.055088	0.055045	0.07
	$\mathbb{E}[C_r \mid \langle 0,0,2\rangle]$	149.92	149.94	0.01

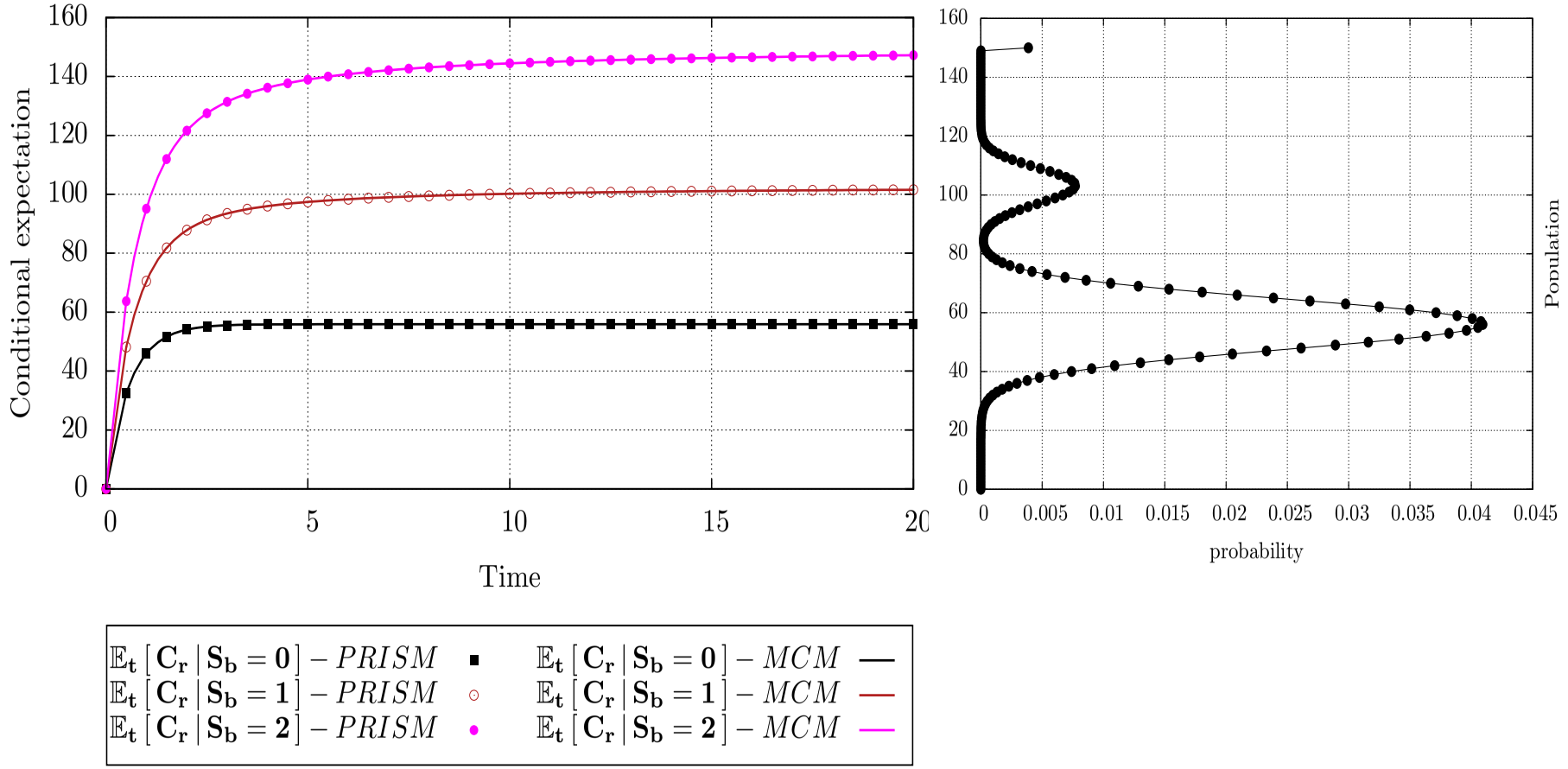
Table 7.8: The error associated with MCM $\mathbb{E}_t[C_r \mid S_b = n]$, n=0,1,2 obtained by comparing the result with PRISM. The model has two servers and 150 clients.

analysis, which find the expectation of the system's behaviour whilst abstracting away the internal stochastic dynamics of the servers.

Moreover, the conditional expectations can quantitatively state how exactly the behaviour of the clients changes as a result of the servers' noisy behaviour. In Fig.7.8 we compare the MCM steady state conditional expectations $\mathbb{E}_t[C_r \mid S_b = \cdot]$ with the steady state distribution of C_r obtained through the analysis of the model's complete and large state space. The comparison shows that the conditional expectations are capable of specifying all the locations in the domain of C_r where significant probability masses are located and capture C_r's multi-modality. In the next sections, we show that the inclusion of higher-order conditional moments (such as conditional variance) in the analysis allows us to also capture the variability around the conditional expectations and therefore, gain a richer representation of this multi-modaility.

7.6 Efficiency Gain

The analyses through both the MCM and PRISM allowed us to study the client's behaviour across the different modes of operation exhibited by the servers. The MCM relies on deriving the numerical solution of a system of DAEs. The size of this system depends on the number of states in the aggregated state space and the number of state variables related to the large groups. This size matters, as it determines the time complexity of running the MCM. On the other hand, in PRISM, a transient analysis is performed over the model's complete state space. In this approach, the key factor in determining the computational cost is the size of this state space. As a final step in our



(A) Conditional expectations of C_r across $S_b = 0$, $S_b = 1$, $S_b = 2$.

(B) The distribution of C_r at the equilibrium.

Figure 7.8: Comparison between the MCM $\mathbb{E}[C_r \mid S_b = n]$, $n = 0, 1, 2$ (when converged to the stationary phase) and the steady state distribution of C_r obtained by the computationally expensive analysis of the complete state space.

	Population of clients			
	150	200	300	400
State space size	906	1206	1806	2406
Runtime-PRISM	8.9	1447	1936	2653
Runtime-MCM	9.0	9.1	9.4	9.15

Table 7.9: Comparison between the MCM and PRISM when the model is initialised with increasing values for n_c. Here, the number of equations in the models' underlying DAEs is constant. The times are expressed in seconds.

example, we compare the MCM and PRISM in terms of their runtime when used for the analysis of our client-server model.

In our first comparison, we constructed four versions of the client-server model with increasing values for the parameter n_c (the total number of clients in the model), namely $n_c = 150, 200, 300, 400$. This results in increasing sizes for the complete state space, whilst the size of the aggregated state space remains constant. The performance measures of interest were the conditional expectations related to the clients with respect to the time interval $0 \leq t \leq 500$. The run times of using the MCM and PRISM were collected and are presented in Table 7.9. Our comparison shows that the MCM is significantly more efficient than PRISM, and as n_c increases its relative efficiency is more pronounced. We also observe that although increasing n_c causes higher computational costs for PRISM, the run time of the MCM is not worsened (the small variations related to the MCM runtime are due to the internal decisions made by the DAE solver). Since n_s does not change, the aggregated state space remains the same and the MCM solves the same equations but with different parameters. This observation underlines an important advantage of the MCM; the method is particularly useful when one intends to investigate the model's behaviour under scenarios with different populations assigned to the large groups.

In another experiment, we checked the impact of changing the number of servers on the time required to capture the model's multi-modality through the MCM and PRISM. Here, we kept $n_c = 500$ as a constant and constructed two versions of the model with varying number of servers, namely $n_s = 2$ and $n_s = 3$. The result is shown in Table 7.10. We observe that increasing n_s enlarges the model's aggregated state space and therefore, when $n_s = 3$, the model's system of DAEs contains more equations. This causes the MCM to have a worse run time than when $n_c = 2$. Nevertheless, we also

	No. of equations	Runtime-MCM	Size of state space	Runtime-PRISM
$n_s = 2$	12	9	3006	3531
$n_s = 3$	20	15	5010	4881

Table 7.10: Comparison between the run times of the MCM and PRISM when the model is initialised with different numbers of servers. The MCM's performance degrades when n_s increases. Yet, an even more severe degradation occurs with respect to PRISM. Times are expressed in seconds.

observe that the degradation observed for the MCM is significantly lower than PRISM. The increase in the population of the small groups increases the size of the aggregated state space, but still, given its inherent size, the impact is limited. On the other hand, such an increase causes a growth of a large magnitude with respect to the model's complete state space. Therefore, whilst the increase in n_s slightly degrades the run time of the MCM, the degradation is more severe with respect to PRISM. This shows that, using the MCM is still favoured in the experiments where the populations of the small groups are scaled up.

Chapter 8

Analysis of LSRB Models Using Higher-Order Conditional Moments

8.1 Introduction

In Chapter 6, we presented the method of conditional expectations as an efficient technique that is suitable for the analysis of LSRB models. These models allow us to capture the dynamics of resource-bound systems. In the analysis based on the conditional expectations we distinguish between the different configurations that resources exhibit and study the behaviour of resource users across those different configurations in terms of conditional expectations. We showed that this method is particularly useful for analysing the impact of the random behaviour of resources on the dynamics of resource users.

The expectations we obtain can be regarded as first moment approximations to the stochastic behaviour of the users. In this chapter, we enrich the analysis of the conditional expectations to include *higher-order conditional moments*.

Whilst it is important to measure the conditional expectations related to the different configurations of the resources, such expectations are not sufficient to capture further important measures, such as, how the actual stochastic evolution of the users deviates from their mean behaviour. The knowledge of the potential distance from the conditional expectations is critical, for instance in tasks such as capacity planning, when in addition to the usual and optimal modes of operation, the system's behaviour is investigated under sub-optimal case scenarios. The higher-order conditional mo-

ments provide us with a richer representation of the system's stochastic behaviour and enable us to form more informed judgements about the performance of the system.

Similarly to the previous chapter, the analysis based on higher-order conditional moments constructs a system of DAEs from the model. In Section 8.2, we provide some definitions. In Section 8.3, we describe the derivation of DAEs for higher-order conditional moments. In Section 8.4, we describe an approach for deriving the initial values required to solve the DAEs.

In Chapter 9, the analysis of higher-order conditional moments is applied to our simple model of the client-server system.

8.2 Definitions

In our derivations, we use vectors extensively. First, we describe some notation and a number of operations related to the vectors [53].

8.2.1 Vector Arithmetic

Definition 31. (Unit vector) *A unit vector* $\mathbf{e} = \langle e_i \rangle, i \in \mathbb{N}$*, is a vector of non-negative integers where all elements are zero, except the element indexed by i which is one. We assume that when analysing a LSRB model* $\mathbb{M}$*, the vector* e_i *has the size* $n(\mathbb{M},l)$*, where* $n(\mathbb{M},l)$ *denotes the number of state variables in vector* $\boldsymbol{\xi}^l$*.*

Consider two vectors $\mathbf{A} = \langle a_i \rangle_{i=1\cdots n}$ and $\mathbf{B} = \langle b_j \rangle_{j=1\cdots n}$ of size n. The following binary operators are defined for vectors such as A and B which have the same size.

Definition 32. (Vectorial exponentiation) *A vector* $\mathbf{A}$ *to the power of a vector* $\mathbf{B}$ *is represented by* $\mathbf{A}^{\mathbf{B}}$ *and is defined as:*

$$\mathbf{A}^{\mathbf{B}} = \prod_{i=1}^{n} (a_i)^{b_i} \tag{8.1}$$

Definition 33. (Vectorial combination) *For vectors* $\mathbf{A}$ *and* $\mathbf{B}$*, the vectorial combination* $\binom{\mathbf{A}}{\mathbf{B}}$ *is defined as:*

$$\binom{\mathbf{A}}{\mathbf{B}} = \prod_{i=1}^{n} \binom{a_i}{b_i} \tag{8.2}$$

where $\binom{a_i}{b_i}$ *is the combinatorial operator defined on integers as* $\binom{a}{b} = \frac{a!}{b!\,(a-b)!}$

Definition 34. (Vectorial inequality) *For two vectors* $\mathbf{A}$ *and* $\mathbf{B}$*, we have* $\mathbf{A} \leq \mathbf{B}$ *when* $\forall\, i = 1, \cdots, n : a_i \leq b_i$*. Similarly, we have* $\mathbf{A} < \mathbf{B}$ *when* $\forall\, i = 1, \cdots, n : a_i < b_i$*.*

The binomial expansion theorem is originally defined for natural numbers. In the following lemma, the theorem is extended to account for vectors.

Lemma 8.2.1. (Binomial expansion for vectors) *For three vectors* $\mathbf{A}$*,* $\mathbf{B}$ *and* $\mathbf{X}$*, the exponentiation* $(\mathbf{A}+\mathbf{B})^{\mathbf{X}}$ *is expanded as:*

$$(\mathbf{A}+\mathbf{B})^{\mathbf{X}} = \sum_{\langle 0\rangle \leq \mathbf{W} \leq \mathbf{X}} \binom{\mathbf{X}}{\mathbf{W}} (\mathbf{A})^{\mathbf{W}} \cdot (\mathbf{B})^{\mathbf{X}-\mathbf{W}} \tag{8.3}$$

where the vectorial combination $\binom{\mathbf{X}}{\mathbf{W}}$ *is derived using Eq.(8.2) and* $\langle 0\rangle$ *is the vector of size* $n(\mathbb{M}, l)$ *with all elements as zero.*

Proof.

$$(\mathbf{A}+\mathbf{B})^{\mathbf{X}} = \prod_{i=1}^{n} (a_i + b_i)^{x_i} = \prod_{i=1}^{n} \sum_{k_i=0}^{x_i} \binom{x_i}{k_i} (a_i)^{k_i} \cdot (b_i)^{x_i - k_i}$$

$$= \left(\sum_{k_1=0}^{x_1} \binom{x_1}{k_1} (a_1)^{k_1} (b_1)^{x_1-k_1} \right) \left(\sum_{k_2=0}^{x_2} \binom{x_2}{k_2} (a_2)^{k_2} (b_2)^{x_2-k_2} \right) \cdots \left(\sum_{k_n=0}^{x_n} \binom{x_n}{k_n} (a_n)^{k_n} (b_n)^{x_n-k_n} \right)$$

$$= \sum_{k_1=0}^{x_1} \sum_{k_2=0}^{x_2} \cdots \sum_{k_n=0}^{x_n} \binom{x_1}{k_1} \binom{x_2}{k_2} \cdots \binom{x_n}{k_n} (a_1)^{k_1} (a_2)^{k_2} \cdots (a_n)^{k_n} (b_1)^{x_1-k_1} (b_2)^{x_2-k_2} \cdots (b_n)^{x_n-k_n}$$

$$= \sum_{\mathbf{K}=\langle k_1,\cdots,k_n\rangle=\langle 0\rangle}^{\mathbf{X}=\langle x_1,\cdots,x_n\rangle} \binom{\langle x_1, \cdots, x_n\rangle}{\langle k_1, \cdots, k_n\rangle} \langle a_1, \cdots, a_n\rangle^{\langle k_1,\cdots,k_n\rangle} \langle b_1, \cdots, b_n\rangle^{\langle x_1-k_1,\cdots,x_n-k_n\rangle}$$

$$= \sum_{\mathbf{K}=\langle k_1,\cdots,k_n\rangle=\langle 0\rangle}^{\mathbf{X}=\langle x_1,\cdots,x_n\rangle} \binom{\mathbf{X}}{\mathbf{K}} (\mathbf{A})^{\mathbf{K}} \cdot (\mathbf{B})^{\mathbf{X}-\mathbf{K}}$$

□

8.2.2 Higher-Order Conditional Moments

First, we briefly review a number of definitions related to LSRB models.

In a LSRB model, the groups are categorised based on their size. The partition $\Delta_{\mathcal{G}} = \{\mathcal{G}_s(\mathbb{M}), \mathcal{G}_l(\mathbb{M})\}$ is defined over the set of the groups $\mathcal{G}(\mathbb{M})$ in the model where

$\mathcal{G}_s(\mathbb{M})$ is the set of small groups and $\mathcal{G}_l(\mathbb{M})$ is the set of large ones. The state vector defined for capturing the state of the system is $\langle \boldsymbol{\xi}^s, \boldsymbol{\xi}^l \rangle$ where $\boldsymbol{\xi}^s$ captures the state of the small groups and $\boldsymbol{\xi}^l$ captures the state of the large ones. The complete state space of the model is denoted by D. When the model satisfies the conditions of the aggregated method that we presented in Chapter 4, its state space can be divided into a number of sub-chains and an aggregation model with the state space D^{agg} is constructed. In the aggregated state space each sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$ in D is represented by a single state $\boldsymbol{\gamma}$. The aggregated state space captures the configurations that the small groups exhibit, whilst abstracting from the dynamics of the large ones.

The behaviour of the large groups is studied in terms of the moments of the conditional distributions $\mathbb{P}_t(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma})$, $\forall\ \boldsymbol{\gamma} \in D^{agg}$. In Chapter 6 we derived a system of DAEs which captures the evolution of the conditional means: $\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right], \forall : \boldsymbol{\gamma} \in D^{agg}$. These moments are regarded as the *first-order* conditional moments of the conditional distributions $\mathbb{P}_t(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma})$. The other important measures related to these conditional distributions are the *higher-order* conditional moments, which describe distances (of different order) between the stochastic behaviour of $\boldsymbol{\xi}^l$ and the calculated expectations. As an example, the conditional moments of the second order, including the conditional variances, are useful for capturing the width of the conditional distributions $\mathbb{P}_t(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma})$.

Each higher-order conditional moment is expressed using a *moment vector* $\boldsymbol{I} = \langle I_i \rangle$, $i = 1 \cdots n(\mathbb{M}, l)$, $I_i \in \mathbb{Z}_{\geq 0}$. The $\boldsymbol{I}$-th conditional moment of $\boldsymbol{\xi}^l$ is defined as follows.

Definition 35. (The **I**-th conditional moment of $\boldsymbol{\xi}^l$ in a sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$). *Consider a sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}} \in D$ represented by the state $\boldsymbol{\gamma} \in D^{agg}$. At any point of time t, the conditional expectation of $\boldsymbol{\xi}^l$ given that at t, $\boldsymbol{\xi}^s_i = \boldsymbol{\gamma}$, is denoted by $\mu(\boldsymbol{\gamma}, t)$. For a moment vector $\boldsymbol{I} = \langle I_i \rangle$, at any time t the $\boldsymbol{I}$-th conditional moment of $\boldsymbol{\xi}^l$ given that at t the small groups are in the configuration $\boldsymbol{\gamma}$, is denoted by $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t)$ and is defined as:*

$$\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t) \;=\; \mathbb{E}_t\left[\left(\boldsymbol{\xi}^l - \mu(\boldsymbol{\gamma}, t)\right)^{\mathbf{I}} \mid \boldsymbol{\gamma}\right] \;=\; \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}^l_i \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \left(\boldsymbol{\xi}^l_i - \mu(\boldsymbol{\gamma}, t)\right)^{\mathbf{I}} \cdot \mathbb{P}_t(\boldsymbol{\xi}^l_i \mid \boldsymbol{\gamma}) \quad (8.4)$$

where $\left(\boldsymbol{\xi}^l_i - \mu(\boldsymbol{\gamma}, t)\right)^{\mathbf{I}}$ is calculated using the vector exponentiation shown in Def. 32.

One important property of a moment vector is its order:

Definition 36. (Order of a moment vector) *For a moment vector $\boldsymbol{I}$, its order is defined to be the sum of its elements:*

$$Ord(\boldsymbol{I}) = \sum_{I_i \in \boldsymbol{I}} (I_i) \quad (8.5)$$

By Def. 36, it is trivial that for any moment vector $\boldsymbol{I}$ with order $Ord(\boldsymbol{I}) = 1$, we have one entry of 1 and all other entries zero, and that $\boldsymbol{M}_{\mathbf{I}}(\cdot,t) = 0$. To avoid dealing with such trivial cases, in this thesis we restrict our moment vectors of order two and above.

8.3 Higher-Order Conditional Moments - Derivation of Equations

Here, we derive a system of equations which, for a given model, captures the evolution of its underlying higher-order conditional moments. Our derivations will be based on the following key lemma. The lemma is originally presented in [38] and [53] for modelling biological systems and here we adapt it to be used for LSRB models.

Proposition 8.3.1. (Higher-order conditional moments of a reward function) *Let us assume that* $T(\boldsymbol{\xi}^l,t) : \mathbb{N}^{n(\mathbb{M},l)} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *is a polynomial function defined over the Markov Chain of model* $\mathbb{M}$*. This function represents a time dependent instantaneous reward; at any point of time* t*, the reward* $T(\boldsymbol{\xi}_i^l,t)$ *is assigned to any state* $\langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l \rangle \in D$*. Assuming that* T *is differentiable across its domain, we have:*

$$\forall\ \mathbb{Y}_{\boldsymbol{\gamma}} \in D\ :\ \frac{d}{d\,t}\Big(\ \mathbb{E}_t\left[T(\boldsymbol{\xi}^l,t)\mid \boldsymbol{\gamma}\right]\cdot \mathbb{P}(\boldsymbol{\gamma})\ \Big) =$$

$$\underbrace{\sum_{S_i=\langle \boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} T(\boldsymbol{\xi}_i^l,t)\ \cdot\ \frac{d\ \mathbb{P}(S_i)}{d\,t}}_{\textit{first term}}\ +\ \underbrace{\mathbb{E}_t\left[\frac{d\,T(\boldsymbol{\xi}^l,t)}{d\,t}\mid \boldsymbol{\gamma}\right]\cdot \mathbb{P}(\boldsymbol{\gamma})}_{\textit{second term}} \tag{8.6}$$

where the conditional expectations $\mathbb{E}_t\left[T(\boldsymbol{\xi}_{,}^l t)\mid \boldsymbol{\gamma}\right]$ *and* $\mathbb{E}_t\left[\frac{d\,T(\boldsymbol{\xi}^l,t)}{d\,t}\mid \boldsymbol{\gamma}\right]$ *are derived using Def. 29 (recall that this definition applies the notion of conditional expectations to the reward functions such as* T*).*

Proof. The proof, adjusted to our notation, is illustrated in Appendix D.1. □

Using Prop. 8.3.1, the following is the outline of our derivations. Given a moment vector $\boldsymbol{I}$, for each sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$ we construct an instance of Eq.(8.6), with the substitution $T(\boldsymbol{\xi}_i^l,t) = \left(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}}$. This means that the reward $\left(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}}$ is assigned to each state $\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l\rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}$. Then, in the equation of each sub-chain the left hand side

and the first and second terms on the right hand side (as shown in Eq.(8.6)) are expanded using the symbolic representation of the model's underlying C-K equations. As a result, we obtain an equation closed in terms of the higher conditional moments $\mathbf{I}'$ of order $Ord(\mathbf{I}') \leq Ord(\mathbf{I})$. By recursively applying the proposition with respect to such $\boldsymbol{I}'$, a closed system of DAEs is ultimately obtained. In the rest of this section, the details of this process will be presented.

Let us consider a sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$ and the states $\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle$ within it. For this sub-chain $\mu(\boldsymbol{\gamma},t)$ denotes the conditional expectation of $\boldsymbol{\xi}^l$ at t, given the configuration $\boldsymbol{\gamma}$ for $\boldsymbol{\xi}^s$. We focus on this sub-chain and for the moment vector $\boldsymbol{I}$ of interest, construct an instance of Eq.(8.6) with the substitution $T(\boldsymbol{\xi}_i^l,t) = \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}$:

$$\frac{d}{d\,t}\Big(\ \mathbb{E}_t\left[\Big(\boldsymbol{\xi}^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}} \mid \boldsymbol{\gamma}\right]\cdot\mathbb{P}(\boldsymbol{\gamma})\ \Big) =$$

$$\underbrace{\sum_{S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle}\Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\ \cdot\ \frac{d\,\mathbb{P}(S_i)}{d\,t}}_{\text{first term}}\ +\ \underbrace{\mathbb{E}_t\left[\frac{d\,\Big(\boldsymbol{\xi}^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}}{dt}\mid\boldsymbol{\gamma}\right]\cdot\mathbb{P}(\boldsymbol{\gamma})}_{\text{second term}} \quad (8.7)$$

In the above equation, we apply the product rule for differentiation to the left hand side to derive:

$$\overbrace{\mathbb{E}_t\left[\Big(\boldsymbol{\xi}^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\mid\boldsymbol{\gamma}\right]}^{\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}\cdot\frac{d\,\mathbb{P}_t(\boldsymbol{\gamma})}{d\,t}\ +\ \overbrace{\left(\frac{d}{d\,t}\mathbb{E}_t\left[\Big(\boldsymbol{\xi}^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\mid\boldsymbol{\gamma}\right]\right)}^{\frac{d\,\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\,t}}\cdot\mathbb{P}_t(\boldsymbol{\gamma})\ =$$

$$\underbrace{\sum_{S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle}\Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\ \cdot\ \frac{d\,\mathbb{P}(S_i)}{d\,t}}_{\text{first term}}\ +\ \underbrace{\mathbb{E}_t\left[\frac{d\,\Big(\boldsymbol{\xi}^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}}{d\,t}\mid\boldsymbol{\gamma}\right]\cdot\mathbb{P}(\boldsymbol{\gamma})}_{\text{second term}} \quad (8.8)$$

In the left hand side of Eq.(8.8) and as indicated above, using Def. 35 we derive: $\mathbb{E}\left[\Big(\boldsymbol{\xi}^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\mid\boldsymbol{\gamma}\right] = \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)$ and $\frac{d}{d\,t}\mathbb{E}\left[\Big(\boldsymbol{\xi}^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\mid\boldsymbol{\gamma}\right] = \frac{d\,\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\,t}$. The reformulation means that the equation can capture the evolution of our conditional moments. Next, we focus on the first and second terms on the right hand side and by utilising the model's C-K equations, we transform them so that we derive an equation closed in terms of the conditional moments. Our derivations are lengthy, and in order to guide the reader a map of them is presented in Fig. 8.1.

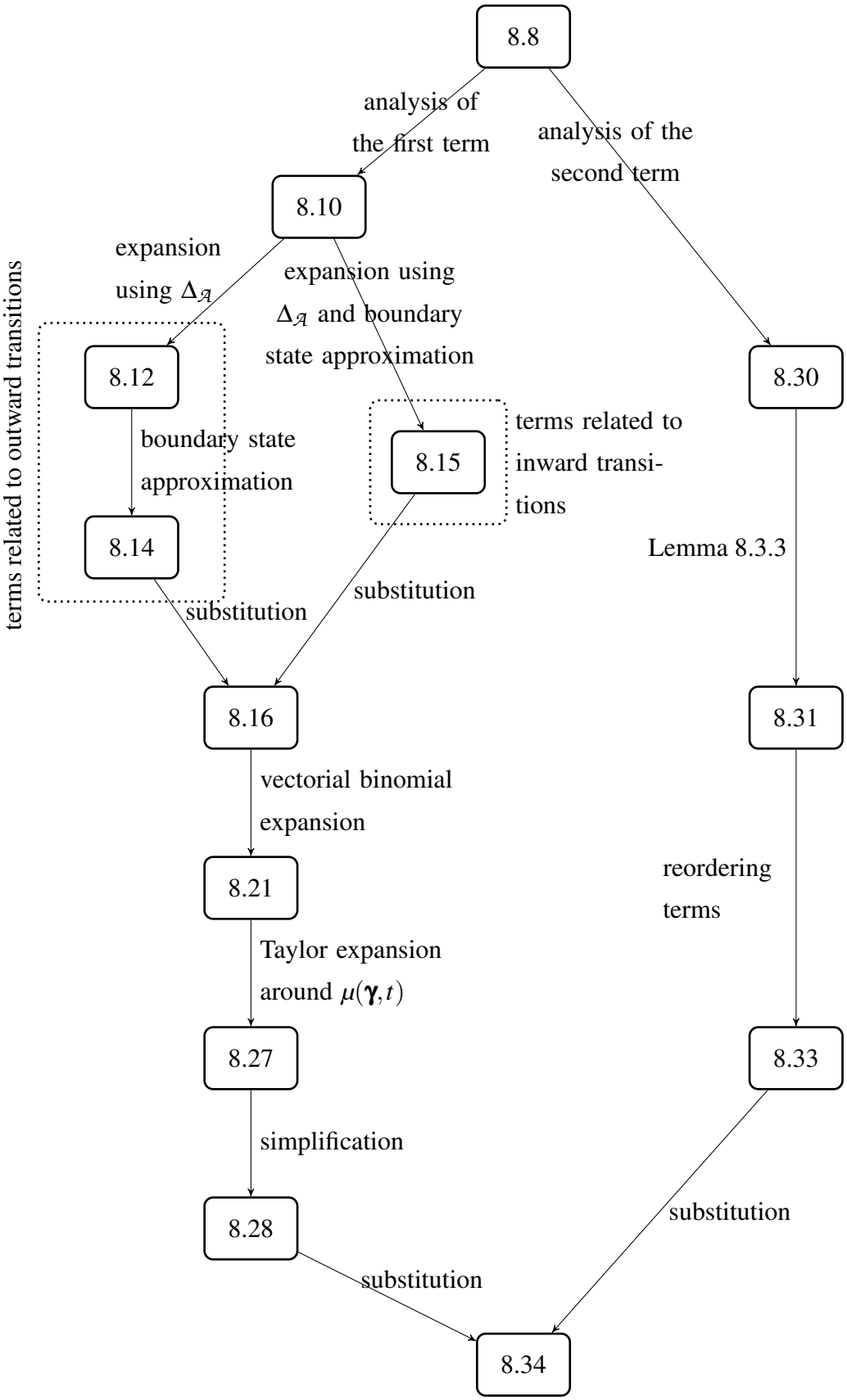


Figure 8.1: Map of the transformations that we apply to the first and second terms on the right hand side of Eq.(8.8) in order to reformulate them in terms of the conditional moments. Recall that the boundary state approximation was introduced on Page 113 in Eq.(6.14).

8.3.1 Closing the Equations - First Term

We use the model's C-K equations and construct an approximation for the first term in Eq.(8.8). Consider Eq.(8.9), the symbolic representation of a C-K equation for a state $S_i \in D$:

$$\forall\ S_i \in D\ :\ \frac{d\ \mathbb{P}_t(S_i)}{d\ t} = \underbrace{-\sum_{S_i \xrightarrow{(\alpha, r_\alpha(S_i))} S_j} r_\alpha(S_i)\cdot \mathbb{P}_t(S_i)}_{\text{outward transitions from } S_i} + \underbrace{\sum_{S_k \xrightarrow{(\alpha, r_\alpha(S_k))} S_i} r_\alpha(S_k)\cdot \mathbb{P}_t(S_k)}_{\text{inward transitions into } S_i} \quad (8.9)$$

Focusing on sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$ and given the moment vector $\boldsymbol{I}$, for each state $\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}$, we multiply both the left and right hand side by $\left(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t)\right)^{\mathbf{I}}$, and then sum over the states in the sub-chain to derive:

$$\forall\ \boldsymbol{\gamma} \in D^{agg}\ :\ \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \left(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t)\right)^{\mathbf{I}} \cdot \frac{d\ \mathbb{P}_t(S_i)}{d\ t} =$$

$$\underbrace{-\sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \left(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t)\right)^{\mathbf{I}} \cdot \sum_{S_i \xrightarrow{(\alpha, r_\alpha(S_i))} S_j} r_\alpha(S_i)\cdot \mathbb{P}_t(S_i)}_{\text{outward transitions from } S_i}$$

$$\underbrace{+\sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \left(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t)\right)^{\mathbf{I}} \cdot \sum_{S_k \xrightarrow{(\alpha, r_\alpha(S_k))} S_i} r_\alpha(S_k)\cdot \mathbb{P}_t(S_k)}_{\text{inward transitions into } S_i} \quad (8.10)$$

In Eq.(8.10), consider the term related to the outward transitions. This term consists of two summations, the outer over the states $S_i \in \mathbb{Y}_{\boldsymbol{\gamma}}$ and the inner over the transitions out of S_i. We can swap the ordering of these summations (similarly to the transformation applied to Eq.(6.11)). When swapping, for each state we sum over only the action types that the state enables[1]. Furthermore, a similar transformation is applied to the term related to the inward transitions. Here, for each action type α, we are restricted to sum only over the states S_k which enable a transition of type α leaving S_k and entering

[1] A state S_i enables an α if $S_i \geq \mathcal{V}_\alpha^-$.

S_i. Eq. (8.10) is turned into:

$$\sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^I \cdot \frac{d}{dt} \mathbb{P}_t(S_i) \;\; =$$

$$\underbrace{- \sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, S_i \geq \mathcal{V}_\alpha^-} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^I \cdot r_\alpha(S_i) \cdot \mathbb{P}_t(S_i)}_{\text{outward transitions}}$$

$$\underbrace{+ \sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \sum_{S_k \geq \mathcal{V}_\alpha^-, S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle = S_k + \mathcal{V}_\alpha} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^I \cdot r_\alpha(S_k) \cdot \mathbb{P}_t(S_k)}_{\text{inward transitions}} \quad (8.11)$$

Note that the first term on the right hand side of Eq.(8.8) is now formed in Eq.(8.11). However, the right hand side of Eq.(8.11) is not yet closed in terms of the conditional moments, and further transformations are required. Some of these transformations are similar to those applied in Section 6.2 which were related to the conditional expectations. Due to the similarity, when possible, we combine a series of them in a single step without presenting full detail and assume it suffices to only give references to explanations in Section 6.2.

The transformations on Eq.(8.11) are applied in three consecutive stages. First, we separately focus on the terms related to the inward and outward transitions and expand them using the partition $\Delta_{\mathcal{A}}$. The resulting equations are then transformed using vectorial binomial expansion as shown in Prop. 8.2.1. In the final stage, we apply a transformation based on Taylor expansion of the apparent rate functions. The descriptions of these stages follow.

8.3.1.1 First Term - Outward Transitions

Recall that the set of action types defined in the model is denoted by $\vec{\mathcal{A}}^*(\mathbb{M})$ and the partition $\Delta_{\mathcal{A}} = \{\vec{\mathcal{A}}_s^*(\mathbb{M}), \vec{\mathcal{A}}_{sl}^*(\mathbb{M}), \vec{\mathcal{A}}_l^*(\mathbb{M})\}$ is defined over $\vec{\mathcal{A}}^*(\mathbb{M})$. In the term related to the outward transitions, the outer summation sums over the action types. We expand this so that the action types in each category are summed over separately. We also use Bayes' Law to transform the probability terms into the conditional form as: $\mathbb{P}_t(\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle) = \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \cdot \mathbb{P}_t(\boldsymbol{\gamma})$. The term related to the outward transitions is

transformed into:

$$
\begin{aligned}
&- \sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\boldsymbol{I}} \cdot r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) \quad = \\
&\quad - \sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,-}} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\boldsymbol{I}} \cdot r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \\
&\quad - \sum_{\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} \sum_{\substack{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle, \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,-} \\ \boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}}} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\boldsymbol{I}} \cdot r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \qquad \star \\
&\quad - \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle, \boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\boldsymbol{I}} \cdot r_\alpha(\boldsymbol{\xi}_i^l) \cdot \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \qquad (8.12)
\end{aligned}
$$

In our next transformation, we exploit the fact that the probability of being in boundary states is close to zero. The condition is formally expressed in Eq.(6.14). By following the same logic as presented for Eq.(6.15), we conclude that in the term related to $\vec{\mathcal{A}}_{sl}^*(\mathbb{M})$ actions (marked by a star above), the condition $\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}$ imposed on the inner summation can be relaxed. This helps us to transform Eq.(8.12) into:

$$
\begin{aligned}
&- \sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\mathbf{I}} \cdot r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) \quad \approx \\
&\quad - \sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,-}} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\mathbf{I}} \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \\
&\quad - \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\mathbf{I}} \cdot r_\alpha(\boldsymbol{\xi}_i^l) \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \qquad (8.13)
\end{aligned}
$$

Note that in any state $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle$ and for any action $\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})$, if $\boldsymbol{\gamma} \not\geq \mathcal{V}_\alpha^{s,-}$, then the action is not enabled in S_i and $r_\alpha(\boldsymbol{\gamma}) = 0$. On the other hand, if $\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})$ and $\boldsymbol{\xi}_i^l \not\geq \mathcal{V}_\alpha^{l,-}$, again the action is not enabled and $r_\alpha(\boldsymbol{\xi}_i^l) = 0$. In each line of Eq.(8.13), we have a summation over only the states that enable the action types. Since the states that do not enable the actions can only contribute zero to the expression, we can relax the conditions $\boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,-}$ and $\boldsymbol{\xi}_i^l \geq \mathcal{V}_\alpha^{l,-}$ and thus, allow the summations to include all

states in the sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$:

$$
\begin{aligned}
&- \sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\mathbf{I}} \cdot r_\alpha(S_i) \cdot \mathbb{P}_t(S_i) \quad \approx \\
&\quad - \sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \overbrace{\sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\mathbf{I}} \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma})}^{\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t)} \\
&\quad - \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \sum_{S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\mathbf{I}} \cdot r_\alpha(\boldsymbol{\xi}_i^l) \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma})
\end{aligned} \tag{8.14}
$$

By using Def. 35, the first term on the right hand side of Eq.(8.14) can be expressed by a higher-order conditional moment. This is highlighted in the equation. However, for the second term, further transformations are required, which will be presented later in the second stage. We substitute Eq.(8.14) back into Eq.(8.10) as its term related to the outward transitions.

8.3.1.2 First Term - Inward Transitions

Now we focus on the term in Eq.(8.10) related to the inward transitions. Similarly to the previous case, the first step in the transformation is to use the partition $\Delta_{\mathcal{A}}$ to expand the summation over the action types. Then, for each incoming transition $S_k \xrightarrow{(\alpha,\cdot)} S_i$ with $S_k = \langle \boldsymbol{\xi}_k^s, \boldsymbol{\xi}_k^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma} - \mathcal{V}_\alpha^s}$ into each state $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}$ we apply the substitution $\boldsymbol{\xi}_k^l = \boldsymbol{\xi}_i^l - \mathcal{V}_\alpha^l$. Finally, the approximation with respect to the boundary states (introduced in Eq.(6.15)) is applied to the term related to $\vec{\mathcal{A}}_{sl}^*(\mathbb{M})$ action types. Then, we derive:

$$
\begin{aligned}
&\sum_{\alpha \in \vec{\mathcal{A}}^*(\mathbb{M})} \sum_{S_k \geq \mathcal{V}_\alpha^-, S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle = S_k + \mathcal{V}_\alpha} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\boldsymbol{I}} \cdot r_\alpha(S_k) \cdot \mathbb{P}_t(S_k) \quad \approx \\
&\qquad \sum_{\substack{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+}}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \sum_{S_k = \langle \boldsymbol{\gamma} - \mathcal{V}_\alpha^s, \boldsymbol{\xi}_k^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma} - \mathcal{V}_\alpha^s}} \Big(\boldsymbol{\xi}_k^l + \mathcal{V}_\alpha^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\boldsymbol{I}} \cdot \mathbb{P}_t(\boldsymbol{\xi}_k^l \mid \boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \\
&+ \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \sum_{S_k = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_k^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_k^l + \mathcal{V}_\alpha^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\boldsymbol{I}} \cdot r_\alpha(\boldsymbol{\xi}_k^l) \cdot \mathbb{P}_t(\boldsymbol{\xi}_k^l \mid \boldsymbol{\gamma})
\end{aligned} \tag{8.15}
$$

In Eq.(8.15) and the right hand side, no term can be directly formulated as a higher-order conditional moment. This will be resolved by the transformations that follow. By substituting the results of our transformation thus far in Eq.(8.11), we have:

$$\sum_{S_i=\langle \boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_i^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\cdot\frac{d}{dt}\,\mathbb{P}_t(S_i) \;\approx$$

$$-\sum_{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma})\cdot\mathbb{P}_t(\boldsymbol{\gamma})\cdot\boldsymbol{M}_I(\boldsymbol{\gamma},t)$$

$$-\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})}\mathbb{P}_t(\boldsymbol{\gamma})\cdot\sum_{S_i=\langle \boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}}\Big(\boldsymbol{\xi}_i^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\cdot r_\alpha(\boldsymbol{\xi}_i^l)\cdot\mathbb{P}_t(\boldsymbol{\xi}_i^l\mid\boldsymbol{\gamma})$$

$$+\sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\sum_{S_k=\langle \boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}}\Big(\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\quad\text{(I)}$$

$$+\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})}\mathbb{P}_t(\boldsymbol{\gamma})\sum_{S_k=\langle \boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}}\Big(\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\cdot r_\alpha(\boldsymbol{\xi}_k^l)\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma})\quad\text{(II)}\quad(8.16)$$

8.3.1.3 First Term - Expanding by Vectorial Binomial

In the second stage of our transformations, we expand the terms marked as (I), (II) in Eq.(8.16) using the vectorial binomial expansion presented in Lemma 8.2.1.

First, we consider (I). In the inner summation of this term, we sum over the states $S_k=\langle\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}$ and calculate the distances $(\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t))^{\boldsymbol{I}}$ with respect the conditional expectation of the sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$. In order to form expressions closed in terms of the conditional moments, we add and subtract $\mu(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)$ to have:

$$\Big(\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)\Big)^{\boldsymbol{I}}=\left(\underbrace{\boldsymbol{\xi}_k^l-\mu(\boldsymbol{\gamma}-\mathcal{V}_{\boldsymbol{\gamma}}^s,t)}+\underbrace{\mu(\boldsymbol{\gamma}-\mathcal{V}_{\boldsymbol{\gamma}}^s,t)+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)}\right)^{\mathbf{I}}\quad(8.17)$$

In Eq. 8.17, we regroup the terms as indicated and apply the binomial expansion to obtain:

$$\Big(\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)\Big)^{\boldsymbol{I}}=\sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}}\binom{\mathbf{I}}{\mathbf{X}}\Big(\boldsymbol{\xi}_k^l-\mu(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)\Big)^{\mathbf{X}}\Big(\mu(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}-\mathbf{X}}$$

(8.18)

Then, we substitute Eq.(8.18) in (I) to transform the term into:

$$\sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\sum_{S_k=\langle\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}}\Big(\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\;=$$

$$\sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\sum_{S_k=\langle\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}}\Bigg(\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot$$
$$\sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}}\binom{\mathbf{I}}{\mathbf{X}}\Big(\boldsymbol{\xi}_k^l-\mu(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)\Big)^{\mathbf{X}}\Big(\mu(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}-\mathbf{X}}\Bigg)$$

The ordering of the second and the third summations above can be swapped. By doing so, we derive:

$$\sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\sum_{S_k=\langle\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}}\Big(\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\;=$$

$$\sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}}\binom{\mathbf{I}}{\mathbf{X}}\Big(\mu(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}-\mathbf{X}}$$
$$\underbrace{\Bigg(\sum_{S_k=\langle\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}-\mathcal{V}_\alpha^s}}\Big(\boldsymbol{\xi}_k^l-\mu(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)\Big)^{\mathbf{X}}\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\Bigg)}_{\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)} \tag{8.19}$$

As highlighted above, the right hand side of Eq.(8.19) is now expressed in terms of the conditional moments $\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma},t)$, $\mathbf{X}\leq\mathbf{I}$. We substitute Eq.(8.19) into Eq.(8.16).

Now we focus on the term in Eq.(8.16) marked by (II). Unlike the previous case, no addition and subtraction is required and by applying the binomial expansion directly

and swapping the ordering of the summations we derive:

$$\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})}\mathbb{P}_t(\boldsymbol{\gamma})\sum_{S_k=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}}\Big(\boldsymbol{\xi}_k^l+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)\Big)^{\boldsymbol{I}}\cdot r_\alpha(\boldsymbol{\xi}_k^l)\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma})$$

$$=\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})}\mathbb{P}_t(\boldsymbol{\gamma})\sum_{S_k=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}}\left(\sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}}\binom{\mathbf{I}}{\mathbf{X}}\Big(\boldsymbol{\xi}_k^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{X}}\cdot\Big(\mathcal{V}_\alpha^l\Big)^{\mathbf{I}-\mathbf{X}}\right)\cdot r_\alpha(\boldsymbol{\xi}_k^l)\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma})$$

$$=\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})}\mathbb{P}_t(\boldsymbol{\gamma})\sum_{\mathbf{X}=\langle 0\rangle}^{I}\binom{\mathbf{I}}{\mathbf{X}}\Big(\mathcal{V}_\alpha^l\Big)^{\mathbf{I}-\mathbf{X}}\sum_{S_k=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}}\Big(\boldsymbol{\xi}_k^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{X}}\cdot r_\alpha(\boldsymbol{\xi}_k^l)\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma}) \tag{8.20}$$

By substituting Eq.(8.20) in place of (II), Eq.(8.16) is transformed into:

$$\begin{aligned}
&\sum_{S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}}\Big(\boldsymbol{\xi}_i^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\cdot\frac{d}{dt}\,\mathbb{P}_t(S_i)\;\approx\\
&-\sum_{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})}r_\alpha(\boldsymbol{\gamma})\cdot\mathbb{P}_t(\boldsymbol{\gamma})\cdot\boldsymbol{M}_I(\boldsymbol{\gamma},t)\\
&-\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})}\mathbb{P}_t(\boldsymbol{\gamma})\cdot\sum_{S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}}\Big(\boldsymbol{\xi}_i^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}\cdot r_\alpha(\boldsymbol{\xi}_i^l)\cdot\mathbb{P}_t(\boldsymbol{\xi}_i^l\mid\boldsymbol{\gamma})\qquad\text{(I)}\\
&+\sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}}r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\\
&\qquad\left(\sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}}\binom{\mathbf{I}}{\mathbf{X}}\Big(\mu(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}-\mathbf{X}}\cdot\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)\right)\\
&+\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})}\mathbb{P}_t(\boldsymbol{\gamma})\sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}}\binom{\mathbf{I}}{\mathbf{X}}\Big(\mathcal{V}_\alpha^l\Big)^{\mathbf{I}-\mathbf{X}}\sum_{S_k=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}}\Big(\boldsymbol{\xi}_k^l-\mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{X}}\cdot r_\alpha(\boldsymbol{\xi}_k^l)\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma})\qquad\text{(II)}
\end{aligned}\tag{8.21}$$

In the right hand side of this equation, the terms which are not formulated in terms of the conditional moments are marked by (I) and (II). In the next stage, we use an approximation based on Taylor expansion [99] to find closed form expressions for these terms.

8.3.1.4 First Term - Transformation by Taylor Expansion

The transformations in this stage rely on the following two lemmas.

Lemma 8.3.1. (Taylor expansion of apparent rate function) *Consider a sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$ and its corresponding conditional expectation $\mu(\boldsymbol{\gamma},t)$. Let α be an action type in $\vec{\mathcal{A}}_l^*(\mathbb{M})$ with the apparent rate function $r_\alpha(\boldsymbol{\xi}^l)$. The rate function $r_\alpha(\boldsymbol{\xi}^l)$ evaluated at any state $S_i = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}$ can be approximated by taking the first two terms of the Taylor expansion around $\mu(\boldsymbol{\gamma},t)$ as:*

$$r_\alpha(\boldsymbol{\xi}_i^l) \;=\; r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,) \;+\; \sum_{\xi(H,C)\in\xi^l} \frac{\partial\, r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,)}{\partial\, \xi(H,C)} \Big(\, \xi_i(H,C) - \mu_{(H,C)}(\boldsymbol{\gamma},t) \,\Big) \quad (8.22)$$

where $\xi(H,C)$ is the element (state variable) in vector $\boldsymbol{\xi}^l$ that captures the number of instances in group $H \in \mathcal{G}_l(\mathbb{M})$ who are in state $C \in ds^*(H)$, $\xi_i(H,C)$ is the number of such instances in the state S_i and $\mu_{(H,C)}(\boldsymbol{\gamma},t)$ is the element in vector $\mu(\boldsymbol{\gamma},t)$ related to the state variable $\xi(H,C)$.

Proof. A brief description of the proof follows. The LSRB models we consider satisfy the split-free condition. This means that the apparent rate functions are piece-wise linear and for any $\mu(\boldsymbol{\gamma},t)$ one of the elements (pieces) is the outputs. Given $\mu(\boldsymbol{\gamma},t)$ from the solution of conditional expectations, we evaluate the function and then apply the Taylor expansion using the form appropriate for multivariate functions. This results in Eq.(8.22). □

Lemma 8.3.2. *Consider a sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$ with the associated conditional expectation $\mu(\boldsymbol{\gamma},t)$. Let $\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})$ be an action type with the apparent rate function $r_\alpha(\boldsymbol{\xi}^l)$. For a state $S_k = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_k^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}$ and a moment vector $\boldsymbol{I}$, the product $\Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}} \cdot r_\alpha(\boldsymbol{\xi}^l)$ can be expressed by the Taylor expansion as:*

$$\Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}} \cdot r_\alpha(\boldsymbol{\xi}_k^l) \;\approx\; \Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}} r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,) \;+$$

$$\sum_{\xi(H,C)\in\xi^l} \frac{\partial\, r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,)}{\partial\, \xi(H,C)} \Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}+\mathbf{e}_{(H,C)}} \quad (8.23)$$

where $\mathbf{e}_{(H,C)}$ is a unit vector with the element related to the state variable $\xi(H,C)$ as one, and zero elsewhere.

Proof. We start by considering the left hand side of Eq.(8.23) and substitute $\Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}}$

with its Taylor expansion derived using Lemma 8.3.1. The right hand side follows immediately. □

Lemma 8.3.2 enables us to reformulate the terms marked by (I) and (II) in Eq.(8.21) in terms of the conditional moments. Focusing on (I), we have:

$$\begin{aligned}
&\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \sum_{S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}} \cdot r_\alpha(\boldsymbol{\xi}_i^l) \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \\
&\approx \sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,) \cdot \sum_{S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}} \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \\
&+ \sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \sum_{\substack{S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle \\ S_i\in\mathbb{Y}_{\boldsymbol{\gamma}}}} \sum_{\xi(H,C)\in\xi^l} \frac{\partial\, r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,)}{\partial\,\xi(H,C)} \Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}+\mathbf{e}_{(H,C)}} \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma})
\end{aligned} \tag{8.24}$$

Next, we change the ordering of the second and third summations in the last line above, to get:

$$\begin{aligned}
&\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \sum_{S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}} \cdot r_\alpha(\boldsymbol{\xi}_i^l) \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \\
&\approx \sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,) \cdot \sum_{S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}} \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \\
&+ \sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \sum_{\xi(H,C)\in\boldsymbol{\xi}^l} \frac{\partial\, r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,)}{\partial\,\xi(H,C)} \sum_{\substack{S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle \\ S_i\in\mathbb{Y}_{\boldsymbol{\gamma}}}} \Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma},t)\Big)^{\mathbf{I}+\mathbf{e}_{(H,C)}} \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\gamma}) \\
&= \sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \left(r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,) \cdot \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t) \;+ \sum_{\xi(H,C)\in\xi^l} \frac{\partial\, r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,)}{\partial\,\xi(H,C)} \cdot \boldsymbol{M}_{\boldsymbol{I}+\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t) \right)
\end{aligned} \tag{8.25}$$

In Eq.(8.25), the right hand side is now closed in terms of the conditional moments. This is substituted as the term marked by (I) in Eq.(8.21).

For the term marked by (II) in Eq.(8.21) a similar transformation is applied (Ap-

pendix D.3). We omit the details and show the final result in the following equation.

$$\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \sum_{\langle 0\rangle\le\mathbf{X}<\mathbf{I}} \binom{\mathbf{I}}{\mathbf{X}} \left(\mathcal{V}_\alpha^l\right)^{\mathbf{I}-\mathbf{X}} \sum_{S_k=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}} \underline{\left(\boldsymbol{\xi}_k^l-\mu(\boldsymbol{\gamma},t)\right)^{\mathbf{X}} r_\alpha(\boldsymbol{\xi}_k^l)\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma})} \;\approx$$

$$\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}} \binom{\mathbf{I}}{\mathbf{X}} \left(\mathcal{V}_\alpha^l\right)^{\mathbf{I}-\mathbf{X}} \left(r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,)\cdot\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma},t) + \sum_{\xi(H,C)\in\boldsymbol{\xi}^l} \frac{\partial\, r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,)}{\partial\,\xi(H,C)}\cdot\boldsymbol{M}_{\mathbf{X}+\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t)\right) \quad (8.26)$$

This is also substituted into Eq.(8.21) in place of the term marked by (II). Consequently, Eq. (8.21) is transformed into:

$$\sum_{S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}} \left(\boldsymbol{\xi}_i^l-\mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}} \cdot \frac{d}{dt}\,\mathbb{P}_t(S_i) \;\approx$$

$$- \sum_{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma})\cdot\mathbb{P}_t(\boldsymbol{\gamma})\cdot\boldsymbol{M}_I(\boldsymbol{\gamma},t)$$

$$- \sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma})\cdot\left(r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,)\cdot\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t) + \sum_{\xi(H,C)\in\xi^l} \frac{\partial\, r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,)}{\partial\,\xi(H,C)}\cdot\boldsymbol{M}_{\mathbf{I}+\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t)\right) \;\star$$

$$+ \sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma}\ge\mathcal{V}_\alpha^{s,+}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot \left(\sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}} \binom{\mathbf{I}}{\mathbf{X}} \left(\mu(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}-\mathbf{X}} \cdot \boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t) \right)$$

$$+ \underline{\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}} \binom{\mathbf{I}}{\mathbf{X}} \left(\mathcal{V}_\alpha^l\right)^{\mathbf{I}-\mathbf{X}} \left(r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,)\cdot\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma},t) + \sum_{(H,C)\in\xi^l} \frac{\partial\, r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,)}{\partial\,\xi(H,C)}\cdot\boldsymbol{M}_{\mathbf{X}+\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t)\right)} \quad (8.27)$$

As a final transformation, we apply the following important simplification. The underlined term in Eq.(8.27) includes a summation over the moment vectors $\langle 0\rangle \le \mathbf{X} \le \mathbf{I}$. We expand this summation by separating the term associated with the largest value $\mathbf{X}=\mathbf{I}$ from the rest $\langle 0\rangle \le \mathbf{X} < \mathbf{I}$. As a result, the terms associated with $\mathbf{X}=\mathbf{I}$ get

cancelled by the term marked by a star above. This simplification turns Eq.(8.27) into the following:

$$
\begin{aligned}
&\sum_{S_i=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_i^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_i^l - \mu(\boldsymbol{\gamma},t) \Big)^{\mathbf{I}} \cdot \frac{d}{dt}\, \mathbb{P}_t(S_i) \;\approx \\
&\quad - \sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t) \\
&\quad + \sum_{\substack{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+}}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \\
&\qquad\qquad \left(\sum_{\mathbf{X}=\langle 0 \rangle}^{\mathbf{I}} \binom{\mathbf{I}}{\mathbf{X}} \Big(\mu(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s, t) + \mathcal{V}_\alpha^l - \mu(\boldsymbol{\gamma},t) \Big)^{\mathbf{I}-\mathbf{X}} \cdot \boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s, t) \right) \\
&\quad + \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \sum_{\mathbf{X}=\langle 0 \rangle}^{\mathbf{X}<\mathbf{I}} \binom{\mathbf{I}}{\mathbf{X}} \Big(\mathcal{V}_\alpha^l \Big)^{\mathbf{I}-\mathbf{X}} \\
&\qquad\qquad \left(r_\alpha(\, \mu(\boldsymbol{\gamma},t)\,) \cdot \boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma},t) + \sum_{\xi(H,C) \in \xi^l} \frac{\partial\, r_\alpha(\, \mu(\boldsymbol{\gamma},t)\,)}{\partial\, \xi(H,C)} \cdot \boldsymbol{M}_{\mathbf{X}+\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t) \right)
\end{aligned}
\tag{8.28}
$$

For a moment vector $\boldsymbol{I}$ with the order $Ord(\boldsymbol{I})$ the simplification enables us to prove an important property about the evolution of the conditional moments $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)$, $\boldsymbol{\gamma} \in D^{agg}$. Consider Eq.(8.27) again and in particular the last term. Here, when $\mathbf{X} = \mathbf{I}$, the conditional moment $\boldsymbol{M}_{\mathbf{I}+\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t)$ with order $Ord(\mathbf{I})+1$ is added to the equation. This might appear to suggest that the evolution of $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)$ depends on some conditional moments which have orders greater than $Ord(\boldsymbol{I})$. Nevertheless, our simplification proves that this is not the case. Since the term related to $\mathbf{X} = \mathbf{I}$ is cancelled, the summation eventually sums over the moment vectors $\mathbf{X}$ that are *strictly less than* $\mathbf{I}$: $\langle 0 \rangle \leq \mathbf{X} < \mathbf{I}$ and therefore, it will only add moments $\boldsymbol{M}_{\mathbf{I}'}(\boldsymbol{\gamma},t)$ with $Ord(\boldsymbol{I}') \leq Ord(\boldsymbol{I})$. As no other term in the equation introduces moments of higher order, this shows that when analysing LSRB models for a conditional moment associated with a moment vector $\mathbf{I}$, we do not need moments with orders higher than $Ord(I)$.

The simplification step above concludes our transformations related to the first term in the right hand side of Eq.(8.8). In the next section, we focus on the second term in this equation.

8.3.2 Closing the Equations - Second Term

The transformations related to this term rely on the following lemma.

Lemma 8.3.3. *Consider a sub-chain $\mathbb{Y}_{\boldsymbol{\gamma}}$ and the associated* time-dependent *conditional expectation $\mu(\boldsymbol{\gamma},t)$. For any moment vector $\boldsymbol{I}$ and any state $S_k = \langle \boldsymbol{\gamma}, \boldsymbol{\xi}_k^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}$, we have:*

$$\frac{d\,\left(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}}}{d\,t} = -\sum_{\xi(H,C)} I_{(H,C)}\,\frac{d\,\mu_{(H,C)}(\boldsymbol{\gamma},t)}{d\,t}\left(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}-\mathbf{e}_{(H,C)}} \tag{8.29}$$

Proof. We start from the left hand side and first expand it using Def. 32. Then, the product rule for differentiation is applied. The right hand side is derived by another application of Def. 32. These steps are shown in detail in Appendix D.2. □

In the first step of our transformations, the second term on the right hand side of Eq.(8.8) is expanded using Lemma 8.3.3. We obtain:

$$\begin{aligned}\mathbb{P}(\boldsymbol{\gamma})\cdot\mathbb{E}_t\left[\frac{d\,\left(\boldsymbol{\xi}^l - \mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}}}{dt}\mid\boldsymbol{\gamma}\right] \;&=\\ -\;\mathbb{P}_t(\boldsymbol{\gamma})\cdot\sum_{S_k=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle}&\left(\sum_{\xi(H,C)\in\boldsymbol{\xi}^l} I_{(H,C)}\cdot\frac{d\,\mu_{(H,C)}(\boldsymbol{\gamma},t)}{d\,t}\cdot\left(\boldsymbol{\xi}_k^l-\mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}-\mathbf{e}_{(H,C)}}\right)\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma})\end{aligned} \tag{8.30}$$

In Eq.(8.30), the ordering of the summations on the right hand side can be swapped. By doing so, we derive:

$$\begin{aligned}\mathbb{P}(\boldsymbol{\gamma})\cdot\mathbb{E}_t\left[\frac{d\,\left(\boldsymbol{\xi}^l - \mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}}}{dt}\mid\boldsymbol{\gamma}\right] \;&=\\ -\;\mathbb{P}_t(\boldsymbol{\gamma})\cdot\sum_{\xi(H,C)\in\boldsymbol{\xi}^l} I_{(H,C)}\cdot\frac{d\,\mu_{(H,C)}(\boldsymbol{\gamma},t)}{d\,t}\cdot&\underline{\left(\sum_{S_k=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle}\left(\boldsymbol{\xi}_k^l-\mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}-\mathbf{e}_{(H,C)}}\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma})\right)}\end{aligned} \tag{8.31}$$

Using Def. 8.4, for the part underlined above we have:

$$\left(\sum_{S_k=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_k^l\rangle}\left(\boldsymbol{\xi}_k^l-\mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}-\mathbf{e}_{(H,C)}}\cdot\mathbb{P}_t(\boldsymbol{\xi}_k^l\mid\boldsymbol{\gamma})\right) = \boldsymbol{M}_{\mathbf{I}-\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t) \tag{8.32}$$

By substituting Eq.(8.32) in Eq.(8.31), we obtain:

$$\mathbb{P}(\boldsymbol{\gamma})\cdot\mathbb{E}_t\left[\frac{d\,\left(\boldsymbol{\xi}^l-\mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}}}{d\,t}\mid\boldsymbol{\gamma}\right] \;=\; -\,\mathbb{P}_t(\boldsymbol{\gamma})\;\cdot\sum_{\xi(H,C)\in\boldsymbol{\xi}^l} I_{(H,C)}\cdot\frac{\partial\,\mu_{(H,C)}(\boldsymbol{\gamma},t)}{\partial\,t}\cdot\boldsymbol{M}_{\mathbf{I}-\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t) \quad (8.33)$$

In Eq.(8.33), the right hand side is now closed in terms of the conditional moments. This is substituted on the right hand side of Eq.(8.8) in place of (II).

8.3.3 Transformation's Result

The transformations presented allow us to reformulate Eq.(8.8) as:

$$\begin{aligned}
\forall\;\boldsymbol{\gamma}\in D^{agg}\;\;:\;\;&\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)\;\cdot\;\frac{d\,\mathbb{P}_t(\boldsymbol{\gamma})}{d\,t}\;+\;\left(\frac{d\,\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\,t}\right)\cdot\;\mathbb{P}_t(\boldsymbol{\gamma})\;\;\approx\\
&-\sum_{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma})\cdot\mathbb{P}_t(\boldsymbol{\gamma})\cdot\boldsymbol{M}_{\boldsymbol{I}}(\boldsymbol{\gamma},t)\\
&+\sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\\
&\qquad\left(\;\sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}}\binom{\mathbf{I}}{\mathbf{X}}\left(\mu(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}-\mathbf{X}}\cdot\;\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)\;\right) \qquad \text{I}\\
&+\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})}\mathbb{P}_t(\boldsymbol{\gamma})\sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{X}<\mathbf{I}}\binom{\mathbf{I}}{\mathbf{X}}\left(\mathcal{V}_\alpha^l\right)^{\mathbf{I}-\mathbf{X}}\\
&\qquad\left(r_\alpha(\;\mu(\boldsymbol{\gamma},t)\;)\cdot\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma},t)\;+\sum_{\boldsymbol{\xi}(H,C)\in\xi^l}\frac{\partial\,r_\alpha(\;\mu(\boldsymbol{\gamma},t)\;)}{\partial\,\xi(H,C)}\cdot\boldsymbol{M}_{\mathbf{X}+\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t)\right) \qquad \text{II}\\
&-\;\mathbb{P}_t(\boldsymbol{\gamma})\;\cdot\sum_{\xi(H,C)\in\boldsymbol{\xi}^l} I_{(H,C)}\cdot\frac{d\,\mu_{(H,C)}(\boldsymbol{\gamma},t)}{d\,t}\cdot\boldsymbol{M}_{\mathbf{I}-\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t) \qquad (8.34)
\end{aligned}$$

In Eq.(8.34) both the left and right hand sides are written in terms of the conditional expectations and the conditional higher-order moments. This equation possesses the desirable form and is used for studying the conditional moments $\boldsymbol{M}_I(\boldsymbol{\gamma},t)$, $\boldsymbol{\gamma}\in D^{agg}$.

Each instance of this equation can be systematically constructed by considering the transitions of type $\overrightarrow{\mathcal{A}}^*_s(\mathbb{M}) \cup \overrightarrow{\mathcal{A}}^*_{sl}(\mathbb{M})$ and the ones with type $\overrightarrow{\mathcal{A}}^*_l(\mathbb{M})$. The pseudo code for this construction is shown in Alg. 4, which is the basis for deriving the equations by a software tool.

8.3.4 Size of the System of DAEs

One important aspect of applying the MCM is its scalability with respect to the number of equations it constructs in the model's underlying system of DAEs. For a moment vector $\boldsymbol{I}$, in order to obtain a higher-order moment $\boldsymbol{M}_I(\boldsymbol{\gamma},t)$, $\boldsymbol{\gamma} \in D^{agg}$ we are required to construct one instance of Eq.(8.34). However, the structure of the equation shows that the evolution of $\boldsymbol{M}_{\boldsymbol{I}}(\boldsymbol{\gamma},t)$ depends on other conditional moments $\boldsymbol{M}_{\boldsymbol{I'}}(\boldsymbol{\gamma'},t)$ with orders $Ord(\boldsymbol{I'}) \leq Ord(\boldsymbol{I})$ and $\boldsymbol{\gamma'} \in D^{agg}$. To account for this dependency and construct a closed system of equations, the system needs to be augmented with new instances of Eq.(8.34) which capture the evolution of the additional moments. This enlarges the model's underlying system of DAEs and increases the computational cost of finding the numerical solution. Therefore, in practice, when analysing for a moment vector $\boldsymbol{I}$, attention must be especially paid to the number of equations that will be constructed and the capabilities of the existing solvers.

For a model $\mathbb{M}$ and a moment vector $\mathbf{I}$, the size of the underlying system of DAEs can be obtained by exploiting the structure of Eq.(8.34). In the following, we explain the derivation of this size. First, let us introduce two lemmas which will be used in our derivation.

Lemma 8.3.4. *[40, Chap. 3] Consider the equation $a_1 + a_2 + \cdots + a_k = B$ where B is a non-negative integer constant and variables a_k, $i \in \{1,2,\cdots,k\}$ are bound to be non-negative integers. The number of solutions that the equation admits is $\binom{B+(k-1)}{(k-1)}$.*

Lemma 8.3.5. *The number of moment vectors with a given order B is $\binom{B+(n(M,l)-1)}{n(M,l)-1}$.*

Proof. By Def. 36, a moment vector $\mathbf{I}$ has order $Ord(\mathbf{I}) = B$ when its elements I_i (which are assumed to be non-negative integers) sum up to B. Thus, every solution to the equation $I_1 + I_2 + \cdots + I_{n(M,l)} = B$ denotes one moment vector with order $Ord(\mathbf{I}) = B$. Using Lemma 8.3.4, we have that the number of the solutions and consequently, the number of the possible moment vectors is $\binom{B+(n(M,l)-1)}{n(M,l)-1}$. □

Algorithm 4 Given an aggregated state $\boldsymbol{\gamma} \in D^{agg}$ and a moment vector $\mathbf{I} > \langle 0 \rangle$, return the equation capturing $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t)$

function HIGHER_ORDER_MOMENT_EQ(aggregated state $\boldsymbol{\gamma}$, moment vector $\mathbf{I}$)
  initialise an equation as *eq*;
  put $\left[\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t) \cdot \frac{d\ \mathbb{P}_t(\boldsymbol{\gamma})}{d\ t} + \left(\frac{d\ \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\ t}\right) \mathbb{P}_t(\boldsymbol{\gamma})\right]$ as the left hand side in *eq*;
  $out\_trans \leftarrow \{\alpha \mid \alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \wedge \boldsymbol{\gamma}$ enables a transition of type $\alpha\}$
  for (action type: $\alpha \in out\_trans$) **do** // outward transitions
    add $(-r_\alpha \cdot \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \boldsymbol{M}_{\boldsymbol{I}}(\boldsymbol{\gamma},t))$ to *eq* ;
  end for
  $lower\_equal\_order\_moments \leftarrow \{\mathbf{X} \mid \langle 0 \rangle \leq \mathbf{X} \leq \mathbf{I}\}$;
  $in_trans \leftarrow \{(\boldsymbol{\gamma}', \alpha) \mid \boldsymbol{\gamma}'$enables a transition of type $\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})$ into $\boldsymbol{\gamma}\}$;
  for (the pair: $(\boldsymbol{\gamma}', \alpha) \in in\_trans$) **do** // inward transitions
    for (moment vector: $\boldsymbol{X} \in lower\_equal\_order\_moments$) **do**
      add $r_\alpha(\boldsymbol{\gamma}') \cdot \mathbb{P}_t(\boldsymbol{\gamma}') \cdot \binom{\mathbf{I}}{\mathbf{X}} \left(\mu(\boldsymbol{\gamma}',t) + \mathcal{V}_\alpha^l - \mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}-\mathbf{X}} \cdot \boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}',t)$ to *eq* ;
    end for
  end for
  $lower\_order\_moments \leftarrow \{\mathbf{X} \mid \langle 0 \rangle < \mathbf{X} \leq \mathbf{I}\}$;
  for (action type: $\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})$) **do** // actions related to large groups only
    for (moment vector: $\boldsymbol{X} \in lower\_order\_moments$) **do**
      expr ="";
      for (state variable: $\xi(H,C) \in \boldsymbol{\xi}^l$) **do**
        $local\_rate = r_\alpha(H,C)$;
        $expr = expr + (local\_rate) \cdot \boldsymbol{M}_{\mathbf{X}+\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t)$
      end for
      $expr = expr + r_\alpha(\mu(\boldsymbol{\gamma},t)) \cdot \boldsymbol{M}_{\boldsymbol{X}}(\boldsymbol{\gamma},t)$;
      $expr = expr \cdot \binom{\mathbf{I}}{\mathbf{X}} \cdot \mathcal{V}_\alpha^l \cdot \mathbb{P}_t(\boldsymbol{\gamma})$;
      add *expr* to *eq* ;
    end for
  end for

28: **for** (state variable: $\xi(H,C) \in \boldsymbol{\xi}^l$) **do**
29: add $\left[-\mathbb{P}_l(\boldsymbol{\gamma}) \cdot I_{(H,C)} \cdot \left(\frac{d\,\mu_{(H,C)}(\boldsymbol{\gamma},t)}{d\,t}\right) \cdot \boldsymbol{M}_{\mathbf{I}-\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t) \right]$ to eq ;
30: **end for**
31: **end function**

The following observations regarding Eq.(8.34) are key in our counting of the number of equations constructed when analysing $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)$.

First, in Eq.(8.34) let us consider the term marked by (II). In this term, we have a summation over the moment vectors $\mathbf{X}$: $\langle 0 \rangle \leq \mathbf{X} < \mathbf{I}$. For any aggregated state $\boldsymbol{\gamma}$, this shows that $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)$ depends both on moments $\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma},t)$ (which have orders strictly less than $Ord(\mathbf{I})$) and *one-step-different* conditional moments $\boldsymbol{M}_{\mathbf{X}+\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t)$ (which, for maximum values of $\mathbf{X}$ have the exact order $Ord(\mathbf{I})$). By taking into account the equations of those moments recursively, we conclude that in order to obtain $\boldsymbol{M}_I(\boldsymbol{\gamma},t)$ the equations for all moment vectors $\mathbf{I}'$ with orders $Ord(\mathbf{I}') \leq Ord(\mathbf{I})$ need to be added to the system of DAEs. Using Lemma 8.3.5, the number of moment vectors with any order k is $\binom{k+(n(M,l)-1)}{n(M,l)-1}$. Therefore, the number of equations constructed for the moment vectors related to the state $\boldsymbol{\gamma}$ and with orders $k = 0, \cdots, Ord(\mathbf{I})$ is $\sum_{k=0}^{Ord(\mathbf{I})} \binom{k+(n(M,l)-1)}{n(M,l)-1}$.

In our second observation we focus on the term marked by (I), which captures the dependency between the conditional moments but with respect to different aggregated states. This term consists of a summation over the moment vectors $\mathbf{X}$: $\langle 0 \rangle \leq \mathbf{X} \leq \mathbf{I}$ and the associated moments $\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s, t)$. These moments are related to the states $(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \in D^{agg}$ that enable inward transitions into $\boldsymbol{\gamma}$. The summation shows that in order to obtain $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)$, the equations related to the moments vectors $\mathbf{X}$ ($\langle 0 \rangle \leq \mathbf{X} \leq \mathbf{I}$) and of the neighbouring states $(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)$ need to be added to the system of DAEs. By considering the equations associated with such $\mathbf{X}$ and the states $(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \in D^{agg}$ recursively, it follows that obtaining $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)$ requires the equations, related to all moments $\mathbf{X}$: $\langle 0 \rangle \leq \mathbf{X} \leq \mathbf{I}$ and to all other states in the aggregated state space, to be also added to the DAEs.

The combination of the observations above show that when analysing for $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)$, we need the equations related to all moment vectors $\langle 0 \rangle \leq \mathbf{X} \leq \mathbf{I}$ with $Ord(\mathbf{X}) \leq Ord(\mathbf{I})$ and for all states in D^{agg} to be present in the system of DAEs. Assuming that the aggregated state space D^{agg} has N_{agg} states and given that for each $\boldsymbol{\gamma} \in D^{agg}$, there exists $\sum_{k=0}^{Ord(\mathbf{I})} \binom{k+(n(M,l)-1)}{n(M,l)-1}$ equations, the total number of equations in the system

is[2]:

$$\sum_{\boldsymbol{\gamma}\in D^{agg}} \sum_{k=0}^{Ord(\mathbf{I})} \left(\begin{array}{c} k+(n(M,l)-1) \\ n(M,l)-1 \end{array} \right) = N_{agg} \times \sum_{k=0}^{Ord(\mathbf{I})} \left(\begin{array}{c} k+(n(M,l)-1) \\ n(M,l)-1 \end{array} \right)$$

$$= N_{agg} \times \sum_{k=0}^{Ord(\mathbf{I})} \left(\begin{array}{c} k+(n(M,l)-1) \\ k \end{array} \right) \quad (8.35)$$

As indicated by Eq.(8.35), when analysing a model for $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)$, the three factors which determine the number of system of DAEs are the size of D^{agg}, the order of $\mathbf{I}$ and the number of state variables in $\boldsymbol{\xi}^l$: $n(M,l)$. The equation shows that for a constant $n(M,l)$ (i.e. a given size for the vector $\boldsymbol{\xi}^l$) the number of equations constructed has the complexity $O(Ord(\mathbf{I})^{n(M,l)})$; for a constant order $\mathbf{I}$, the complexity is $O((n(M,l))^{Ord(\mathbf{I})})$; and for varying $\mathbf{I}$ and $n(M,l)$ the number of equations is bounded by $x!$, $x = ((n(M,l)-1)+Ord(I))$, which by the Stirling approximation[3] [82], lies within the complexity class of $O(x^x)$. Consequently, the size of the DAEs quickly grows as we analyse for higher-order moments or larger models; for a given model with the state vector $\boldsymbol{\xi} = \langle \boldsymbol{\xi}^s, \boldsymbol{\xi}^l \rangle$, depending on the moment vector $\mathbf{I}$, the size can be so large that finding the numerical solution exceeds the capabilities of the existing solvers. In this case, applying the MCM, like the analysis by construction of the complete state space, becomes computationally infeasible. In theory, this underlines an important limitation with respect to the analysis of the higher-order conditional moments.

Nevertheless, the inherent limitation stated has little impact when the method is used in practice. Since the aggregated state space abstracts from the dynamics of the model's large groups, it is relatively small (orders of magnitude smaller than the complete state space). With respect to the moments, usually, the analysis is bounded to finding the conditional moments up to only the *third* order; since for the conditional distributions $\mathbb{P}_t(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma})$, the first three conditional moments, i.e. conditional expectations, conditional variance and conditional skewness capture the majority of information about the shape of the distributions and the moments of higher orders practically add little meaningful information whilst coming at a high cost. Thus, usually the equations up to the second or the third order are needed (an example of this will be given

[2]Note that $\binom{a}{b} = \binom{a}{a-b}$.

[3]According to Stirling approximation for any $n \geq 1$ we have: $\sqrt{2\pi} n^{\frac{n+1}{2}} e^{-n} \leq n! \leq e \cdot n^{\frac{n+1}{2}} e^{-n}$. Consequently, an algorithm with complexity $O(n!)$ lies within the complexity class $O(n^n)$.

in Chap. 10). Finally, the capacity of the existing solvers is also important. Having been improved over the last decade, modern DAE numerical solvers such as the one in Matlab's Suite or the SUNDIALS package are capable of handling a system with as many as ten thousand equations relatively efficiently. Considering the bound on the order of the moments, this limit is not exceeded. Consequently, in spite of the theoretical limitation, in practice the MCM is considered as an efficient analysis method for LSRB models.

8.4 Construction of the Initial Values

The system of equations that the analysis of higher-order moments constructs can be regarded as an initial value problem. For the analysis with respect to a moment vector $\mathbf{I}$, the set of initial values which need to be specified are the initial conditional moments $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t_0)$ and their derivatives $\frac{d\, \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\, t}\big|_{t_0}$, for all $\boldsymbol{\gamma} \in D^{agg}$. In this section, we describe a method for the derivation of these initial values.

In Section 6.4, we considered the derivation of the initial values $\mu(\boldsymbol{\gamma}, t_0)$ and $\frac{d\, \mu(\boldsymbol{\gamma},t)}{d\, t}\Big|_{t_0}$ for the system of DAEs constructed when the analysis of the conditional expectations is applied. The main input into that derivation is the initial probability distribution that the modeller specifies over the complete state space. From this distribution, first one derives an initial marginal probability distribution $\mathbb{P}_{t_0}(\boldsymbol{\gamma})$ over the states in D^{agg}. This distribution is then used to build the partition $\Delta_{\mathcal{Z}} = \{D_z^{agg}, D_{nz}^{agg}\}$ over D^{agg}, where D_z^{agg} consists of the states $\boldsymbol{\gamma}$ with $\mathbb{P}_{t_0}(\boldsymbol{\gamma}) = 0$ and D_{nz}^{agg} is the set of states $\boldsymbol{\gamma}$ for which $\mathbb{P}_{t_0}(\boldsymbol{\gamma}) \neq 0$. For each aggregated state $\boldsymbol{\gamma}$, if $\boldsymbol{\gamma} \in D_{nz}^{agg}$, then its initial values $\mu(\boldsymbol{\gamma}, t_0)$ and $\frac{d\, \mu(\boldsymbol{\gamma},t)}{d\, t}\Big|_{t_0}$ are straightforwardly derived using the procedure referred to as SIVE (which relies on deriving the initial conditional distribution $\mathbb{P}_{t_0}(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma})$ and using Bayes' Law) from the initial distribution. However, if $\boldsymbol{\gamma} \in D_z^{agg}$, the SIVE is not applicable. For a state $\boldsymbol{\gamma}$ of this case, first one calculates its initial distance $\nabla_{\boldsymbol{\gamma}}$ from the D_{nz}^{agg} states (see Def. 30). Then, as shown in Prop. 6.4.2, the initial value $\mu(\boldsymbol{\gamma}, t_0)$ is derived by differentiating the equation capturing the evolution of $\mu(\boldsymbol{\gamma}, t)$ up to the order $(\nabla_{\boldsymbol{\gamma}} - 1)$ and evaluating the result at t_0. Furthermore, as illustrated in Prop. 8.4.2, the initial value $\frac{d\, \mu(\boldsymbol{\gamma},t)}{d\, t}\Big|_{t_0}$ is obtained by finding the $\nabla_{\boldsymbol{\gamma}}$-th derivative of the same equation and again evaluating the result at t_0.

The initial values related to the higher-order conditional moments can be obtained

by following an analogous approach. Assuming that the partition Δ_Z is constructed, for each $\boldsymbol{\gamma} \in D^{agg}$, depending on whether $\boldsymbol{\gamma} \in D_z^{agg}$ or $\boldsymbol{\gamma} \in D_{nz}^{agg}$, different paths are taken. The derivations associated with each of the two cases are explained in the following. Note that due to the similarity of these derivations to those related to the initial conditional expectations, we skip the details and only provide the final results.

8.4.1 Initial Moments - States with Non-zero Initial Probability

For any moment vector $\mathbf{I}$ and aggregated state $\boldsymbol{\gamma} \in D_z^{agg}$, the initial values $\boldsymbol{M}_I(\boldsymbol{\gamma}, t_0)$ and $\frac{d\, \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\, t}\big|_{t_0}$ are derived using the following procedure, which we refer to as the *simple extraction of higher-order initial values* (SEHV).

1. Using the initial distribution, the initial marginal distribution $\mathbb{P}_{t_0}(\boldsymbol{\gamma})$ is derived over the aggregated state space.

2. Assuming that the initial conditional expectations $\mu(\boldsymbol{\gamma}, t_0)$ are known using the SIVE, we apply Def. 8.4 to obtain the initial value $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t_0)$.

3. We substitute $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t_0)$ in the instance of Eq.(8.34) which captures the evolution of $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t)$ and evaluate at t_0. Thus, $\frac{d\, \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\, t}\big|_{t_0}$ is obtained.

The structure of Eq.(8.4) shows that for any aggregated state $\boldsymbol{\gamma}$, the initial derivative $\frac{d\, \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\, t}\big|_{t_0}$ depends on the initial conditional moments $\boldsymbol{M}_{\mathbf{I}'}(\boldsymbol{\gamma}', t_0)$ related to other aggregated states and with orders $Ord(\mathbf{I}') \leq Ord(\mathbf{I})$. Therefore, for all aggregated states and when constructing the initial values, we have to start from the lower-order moments and progressively move toward the higher-order ones.

8.4.2 Initial Moments - States with Zero Initial Probability

For the states in D_z^{agg} the following propositions enable us to derive their initial conditional moments and the associated derivatives.

Proposition 8.4.1. *For any aggregated state $\boldsymbol{\gamma} \in D_z^{agg}$ with $\nabla_{\boldsymbol{\gamma}}$ as its initial distance, and any moment vector $\mathbf{I}$, the $(\nabla_{\boldsymbol{\gamma}} - 1)$-th derivative of its associated conditional moment equation is sufficient for calculating $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t_0)$. Moreover, by removing the terms*

that are zero in the differentiated equation, we derive:

$$\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t_0)\ \cdot\ \frac{d^{\nabla_{\boldsymbol{\gamma}}}\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\ t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}\quad\approx$$

$$\sum_{\substack{\alpha\in\vec{\mathcal{A}}^*_s(\mathbb{M})\cup\vec{\mathcal{A}}^*_{sl}(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}^{s,+}_\alpha,\nabla_{\boldsymbol{\gamma}-\mathcal{V}^s_\alpha}=\nabla_{\boldsymbol{\gamma}}-1}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha)\cdot\frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\ (\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha))}{d\ t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0}\cdot$$

$$\left(\ \sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}}\binom{\mathbf{I}}{\mathbf{X}}\Big(\mu(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha,t_0)+\mathcal{V}^l_\alpha-\mu(\boldsymbol{\gamma},t_0)\Big)^{\mathbf{I}-\mathbf{X}}\ \cdot\ \boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha,t_0)\ \right) \tag{8.36}$$

where $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t_0)$ can be obtained by reordering of the terms.

The structure of Eq.(8.36) is important. It shows that for any $\boldsymbol{\gamma}\in D_z^{agg}$ and moment vector $\mathbf{I}$, the initial value $\boldsymbol{M}_I(\boldsymbol{\gamma},t_0)$ depends on the higher-order initial conditional moments $\boldsymbol{M}_{\mathbf{I}'}(\boldsymbol{\gamma}',t_0)$ which are related to the states $\boldsymbol{\gamma}'$ with initial distances $\nabla_{\boldsymbol{\gamma}'}$ strictly less than $\nabla_{\boldsymbol{\gamma}}$ and have orders $Ord(\mathbf{I}')\leq Ord(\mathbf{I})$. The observation enables us to propose a recursive algorithm which iterates over all states in the aggregated states and all moments less than or equal to $\mathbf{I}$, and outputs the related initial values. The pseudo code of this algorithm is shown in Alg. 5.

As part of the set of initial values and in addition to the conditional moments $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t_0)$, the derivatives of these initial moments with respect to time, i.e. $\frac{d\ \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\ t}\big|_{t_0}$, need also to be specified. These are derived using the following proposition.

Proposition 8.4.2. *For any aggregated state $\boldsymbol{\gamma}\in D_z^{agg}$ with initial distance $\nabla_{\boldsymbol{\gamma}}$, the $\nabla_{\boldsymbol{\gamma}}$-th derivative of the equation that captures the evolution of $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)$ is sufficient to calculate $\frac{d\ \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\ t}\big|_{t_0}$. Moreover, by removing the terms that reduce to zero in the differentiated equation, we obtain Eq.(8.37) where $\frac{d\ \mathbb{E}_t\left[\boldsymbol{\xi}^I|\boldsymbol{\gamma}\right]}{d\ t}\Big|_{t_0}$ can be obtained by reordering of the terms.*

$$(\nabla_{\boldsymbol{\gamma}}+1)\cdot\frac{d^{\nabla_{\boldsymbol{\gamma}}}\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}\cdot\underline{\frac{d\,\boldsymbol{M_I}(\boldsymbol{\gamma},t)}{d\,t}\Big|_{t_0}}+\frac{d^{(\nabla_{\boldsymbol{\gamma}}+1)}\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{(\nabla_{\boldsymbol{\gamma}}+1)}}\Big|_{t_0}\cdot\boldsymbol{M_I}(\boldsymbol{\gamma},t_0)\;=$$

$$-\sum_{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma})\cdot\frac{d^{\nabla_{\boldsymbol{\gamma}}}\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}\cdot\boldsymbol{M_I}(\boldsymbol{\gamma},t_0) \qquad (I)$$

$$+\sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+},\nabla_{(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)}\leq\nabla_{\boldsymbol{\gamma}}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\frac{d^{\nabla_{\boldsymbol{\gamma}}}\,(\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s))}{d\,t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}\cdot$$
$$\left(\sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}}\binom{\mathbf{I}}{\mathbf{X}}\Big(\mu(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t_0)+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t_0)\Big)^{\mathbf{I}-\mathbf{X}}\cdot\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t_0)\right) \qquad (II)$$

$$+\sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+},\nabla_{(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)}<\nabla_{\boldsymbol{\gamma}}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\nabla_{\boldsymbol{\gamma}}\cdot\frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\,(\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s))}{d\,t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0}\cdot$$
$$\left(\sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}}\binom{\mathbf{I}}{\mathbf{X}}\Big(\mu(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t_0)+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t_0)\Big)^{\mathbf{I}-\mathbf{X}}\cdot\frac{d\,\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)}{d\,t}\Big|_{t_0}\right) \qquad (III)$$

$$+\sum_{\substack{\alpha\in\vec{\mathcal{A}}_s^*(\mathbb{M})\cup\vec{\mathcal{A}}_{sl}^*(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}_\alpha^{s,+},\nabla_{(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)}<\nabla_{\boldsymbol{\gamma}}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s)\cdot\nabla_{\boldsymbol{\gamma}}\cdot\frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\,(\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s))}{d\,t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0}\cdot\sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{I}}\binom{\mathbf{I}}{\mathbf{X}}\cdot\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t_0)$$
$$\left[\sum_{\xi(H,C)\in\boldsymbol{\xi}^l}(\,\mathbf{I}_{(H,C)}-\mathbf{X}_{(H,C)}\,)\cdot\left(\frac{d\,\mu_{(H,C)}(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t)}{d\,t}\Big|_{t_0}-\frac{d\,\mu_{(H,C)}(\boldsymbol{\gamma},t)}{d\,t}\Big|_{t_0}\right)\cdot\right.$$
$$\left.\Big(\mu(\boldsymbol{\gamma}-\mathcal{V}_\alpha^s,t_0)+\mathcal{V}_\alpha^l-\mu(\boldsymbol{\gamma},t_0)\Big)^{\boldsymbol{I}-\boldsymbol{X}-\boldsymbol{e}_{(H,C)}}\right] \qquad (IV)$$

$$+\sum_{\alpha\in\vec{\mathcal{A}}_l^*(\mathbb{M})}\left(\frac{d^{\nabla_{\boldsymbol{\gamma}}}\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}\right)\cdot\sum_{\mathbf{X}=\langle 0\rangle}^{\mathbf{X}<\mathbf{I}}\binom{\mathbf{I}}{\mathbf{X}}\Big(\mathcal{V}_\alpha^l\Big)^{\mathbf{I}-\mathbf{X}}$$
$$\left(r_\alpha(\,\mu(\boldsymbol{\gamma},t_0)\,)\cdot\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma},t_0)+\sum_{\boldsymbol{\xi}(H,C)\in\xi^l}\frac{\partial\,r_\alpha(\,\mu(\boldsymbol{\gamma},t)\,)}{\partial\,\xi(H,C)}\Big|_{t_0}\cdot\boldsymbol{M}_{\mathbf{X}+\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t_0)\right) \qquad (V)$$

$$-\left(\frac{d^{\nabla_{\boldsymbol{\gamma}}}\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}\right)\cdot\sum_{\xi(H,C)\in\boldsymbol{\xi}^l}I_{(H,C)}\cdot\frac{d\,\mu_{(H,C)}(\boldsymbol{\gamma},t)}{d\,t}\Big|_{t_0}\cdot\boldsymbol{M}_{\mathbf{I}-\mathbf{e}_{(H,C)}}(\boldsymbol{\gamma},t_0) \qquad (VI) \quad (8.37)$$

The structure of the Eq.8.37 shows that in order to obtain $\frac{d\,\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\,t}\big|_{t_0}$ that is related to the aggregated state $\boldsymbol{\gamma}$ and moment vector $\mathbf{I}$, the following sets of initial values need to have been calculated. The first, which corresponds to the same aggregated states consists of the initial moment $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t_0)$ (see the line marked by (I)); the initial moments $\boldsymbol{M}_{\mathbf{I}'}(\boldsymbol{\gamma},t_0)$ for moment vectors $\mathbf{I}'$ with $Ord(\mathbf{I}') < Ord(\mathbf{I})$ (the lines marked by (V) and (VI)); and the initial conditional expectation $\mu(\boldsymbol{\gamma},t_0)$) (the line marked by (V)). The second set corresponds to the initial values related to the aggregated states $(\boldsymbol{\gamma} - \mathcal{V}_{\alpha}^{s})$ which enable transitions into $\boldsymbol{\gamma}$. This consists of the initial moments $\boldsymbol{M}_{\mathbf{I}''}(\boldsymbol{\gamma} - \mathcal{V}_{\alpha}^{s},t_0)$: $Ord(\mathbf{I}'') \leq Ord(\mathbf{I})$ (line marked by (II)); the initial derivatives $\frac{d\,\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}-\mathcal{V}_{\alpha}^{s},t)}{d\,t}\big|_{t_0}$ such that $\mathbf{X} \leq \mathbf{I}$ and $\nabla_{\boldsymbol{\gamma}-\mathcal{V}_{\alpha}^{s}} < \nabla_{\boldsymbol{\gamma}}$ (states with strictly less distance) (lines marked by (III) and (IV)); and finally, the initial conditional expectations $\mu(\boldsymbol{\gamma} - \mathcal{V}_{\alpha}^{s},t_0)$ (the same lines). Following this structure, we can propose a recursive algorithm for finding $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t_0)$. The pseudo code of this algorithm is shown in Alg. 6.

8.4.3 Approximating the Initial Values

In Sections 6.4 and 8.4, we considered the problem of finding the initial values $\mu(\boldsymbol{\gamma},t_0)$ and $\frac{d\,\mu(\boldsymbol{\gamma},t)}{d\,t}\Big|_{t_0}$, and $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t_0)$ and $\frac{d\,\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\,t}\big|_{t_0}$, for the aggregated states $\boldsymbol{\gamma} \in D_z^{agg}$. As described, the rigorous derivation of these initial values requires the construction of a system of equations consisting of the instances of Eq.(6.52) and (6.57) for conditional expectations, and Eq.(8.36) and (8.37) for higher-order moments.

The equations in this system capture the complex dependencies between the initial values. To deal with the inherent complexity and in order to reduce the human error, the correct construction of the equations can only be done with a robust software tool. For this thesis, some exploratory implementation was done but time constraints did not allow this to be developed into a fully-fledged tool. Therefore, we propose and utilize an alternative approach based on stochastic simulation which has a reduced complexity and can still approximate the required initial values with high accuracy.

The complex calculation of the initial values for D_z^{agg} states can be circumvented by exploiting one important feature of the evolution of the model's complete probability distribution. Recall that given an initial complete distribution $\mathbb{P}_{t_0}(\boldsymbol{\xi})$, the conditional distributions $\mathbb{P}_{t_0}(\boldsymbol{\xi}^{l} \mid \boldsymbol{\gamma})$ are undefined for the sub-chains $\mathbb{Y}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma} \in D_z^{agg}$. Consequently, at $t = t_0$ their associated initial values cannot be computed through the initial distribution. Nevertheless, one observes that as time elapses, the initial probability masses

associated with $\mathbb{Y}_{\boldsymbol{\gamma}'}, \boldsymbol{\gamma}' \in D_{nz}^{agg}$ start to diffuse across the state space and the model gradually experiences *evolving probability distributions* where all sub-chains have non-zero probability masses. This is also observed with respect to the aggregated state space; at time $t = t_0$, for $\boldsymbol{\gamma} \in D_z^{agg}$ we have: $\mathbb{P}_{t_0}(\boldsymbol{\gamma}) = 0$. However, as time elapses, due to the probability fluxes out of D_{nz}^{agg} states, it will gradually emerge that $\forall\, \boldsymbol{\gamma} \in D^{agg}, \mathbb{P}_t(\boldsymbol{\gamma}) \neq 0$. In the emerging distributions, the conditional probability distributions $\mathbb{P}_t(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma})$ are definable and the initial values above can be calculated by applying Bayes' Law.

The spread of the probability within the state space forms the basis of our approximate method. The main idea is that instead of finding the initial values at exactly t_0 and having to deal with D_z^{agg} states, we obtain the values at a point of time t_{init} shortly after t_0 using the distribution $\mathbb{P}_{t_{init}}(S)$, in which no aggregated state (sub-chain) has zero probability mass. Assuming that t_0 and t_{init} are close, we essentially replace the analysis that starts from t_0 with one that focuses on time t_{init} onwards. Given this intuition, the steps of this approach are the following:

1. The modeller specifies a point of time t_{init} close to t_0 such that $\forall\, \mathbb{Y}_{\boldsymbol{\gamma}} : \mathbb{P}_{t_{init}}(\boldsymbol{\gamma}) \neq 0$. The derivation of t_{init} is discussed below.

2. Using the method of stochastic simulation, an approximate $\mathbb{P}^{apx}_{t_{init}}(S)$ to the probability distribution $\mathbb{P}_{t_{init}}(S), S \in D$ is derived. This requires running a sufficiently large number of stochastic simulations from t_0 until t_{init} in such a way that the requirements on the defined confidence intervals are met.

3. Using $\mathbb{P}^{apx}_{t_{init}}(S)$, the marginal distribution $\mathbb{P}^{apx}_{t_{init}}(\boldsymbol{\gamma})$ and the conditional probability distributions $\mathbb{P}^{apx}_{t_{init}}(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma})$ are derived.

4. For each $\boldsymbol{\gamma} \in D^{agg}$, we substitute $\mathbb{P}^{apx}_{t_{init}}(\boldsymbol{\gamma})$, $\mathbb{P}^{apx}_{t_{init}}(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma})$ in Def. 28 (definition of conditional expectations) and the approximate initial conditional expectation $\mu^{apx}(t_{init}, \boldsymbol{\gamma})$ is derived. Furthermore, the initial higher-order moments $\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}, t_{init})$ for the moment vectors $\mathbf{X}$: $Ord(\mathbf{X}) \leq Ord(\mathbf{I})$ are derived by Def. 35 and using $\mathbb{P}^{apx}_{t_{init}}(\boldsymbol{\gamma})$, $\mathbb{P}^{apx}_{t_{init}}(\boldsymbol{\xi}^l \mid \boldsymbol{\gamma})$, and $\mu^{apx}(t_{init}, \boldsymbol{\gamma})$.

5. The initial value $\mu^{apx}(t_{init}, \boldsymbol{\gamma})$ is substituted in the instance of Eq.(6.36) which captures the evolution of $\mu(\boldsymbol{\gamma}, t)$ and by simplification, $\frac{d\ \mu(\boldsymbol{\gamma},t)}{d\ t}\Big|_{t_{init}}$ is derived (this step need not be manually done by the modeller as the existing solvers can automatically find the derivatives by taking into account the algebraic constraints present in the DAEs.)

6. For the moment vectors $\mathbf{X}$, the initial approximate moments $\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}, t_{init})$ are substituted in the instances of Eq.(8.34) which capture the evolutions of $\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}, t)$ and the initial derivatives $\frac{d\,\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma},t)}{d\,t}\Big|_{t_{init}}$ are derived (similar to the previous step, this step is also automated by the solvers). This is done by starting from lower-order vectors $\mathbf{X}$ and then gradually increasing the order.

Using the above method, the values $\mu^{apx}(t_{init}, \boldsymbol{\gamma})$, $\frac{d\,\mu(\boldsymbol{\gamma},t)}{d\,t}\Big|_{t_{init}}$, $\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}, t_{init})$ and $\frac{d\,\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma},t)}{d\,t}\Big|_{t_{init}}$ are derived and specified as the initial values for DAEs. The solution shows the evolution of the conditional moments from $t = t_{init}$ onwards until the time of interest. Hereafter, the method is referred to as extraction of *approximate initial values* (AIV).

A key input to running AIV is the parameter t_{init}. The appropriate value of this can be approximated by transient analysis of the aggregated state space. The steps for finding t_{init} are the following.

1. Given the distribution $\mathbb{P}_{t_0}(S)$, $S \in D$, the initial marginal distribution $\mathbb{P}_{t_0}(\boldsymbol{\gamma})$, $\boldsymbol{\gamma} \in D^{agg}$ is derived.

2. The modeller specifies a parameter δ_t which represents a small time step close to zero. Using δ_t, the time points $t_0 + \delta_t, t_0 + 2\delta_t, \cdots, t_0 + i\delta_t, \cdots$ are generated. These mark the candidate values for t_{init}.

3. The modeller progressively considers $t_i = t_0 + i \cdot \delta_t$ for increasing values of i. For each t_i, the distribution $\mathbb{P}_{t_i}(\boldsymbol{\gamma})$ is found by a transient analysis of the aggregated state space. The solution shows the probability of being in each of the aggregated states (the probability of being in each sub-chain) at t_i. A time point t_i is chosen as t_{init} for AIV if the condition $\forall\, \boldsymbol{\gamma} \in D^{agg} : \mathbb{P}_{t_i}(\boldsymbol{\gamma}) \neq 0$ is satisfied. The search terminates when the the first suitable t_i is found.

Having described the steps of the AVI, let us focus on its efficiency. In the first phase, the appropriate t_{init} is found. As the size of the aggregated state is relatively small and the time points of interest are usually close to zero, the transient analyses performed are done efficiently with negligible computational cost. Furthermore, in the second phase and when the simulations are executed to find $\mathbb{P}^{apx}_{t_{init}}(S)$, the simulations' end time is again close to t_0. Here, although a large number of trajectories are produced, each one covers a short time span (the first steps of the model's evolution) and they collectively introduce very little computational cost. Furthermore, the simulation

runs are independent and can be executed in parallel. From this analysis, we conclude that the AIV offers a high degree of efficiency and is useful in practice. An example of this method is presented in the Chapter 10.

Algorithm 5 Given an aggregated state $\boldsymbol{\gamma} \in D^{agg}$ and a moment vector $\mathbf{I} > \langle 0 \rangle$, return $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t_0)$

function INIT_COND_MOMENT(aggregated state $\boldsymbol{\gamma}$, moment vector $\mathbf{I}$)
  $\mathbb{P}_{t_0}(D)$ // initial distribution over the complete state space ;
  Sol^{agg} // solution of the analysis of D^{agg} by the ∇_{max}-th order;
  if $(Ord(\mathbf{I}) = 1)$ **then** // Base case
    return 0;
  end if
  if $(\boldsymbol{\gamma} \in D^{agg}_{nz})$ **then** // Base case
    $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t_0) \leftarrow$ derive $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t_0)$ using $\mathbb{P}_{t_0}(D)$;
    return $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t_0)$;
  end if
  if $(\boldsymbol{\gamma} \in D^{agg}_{z})$ **then**
    $origin\_states \leftarrow$ set of $\boldsymbol{\gamma}' \in D^{agg}$ that enable transition $\boldsymbol{\gamma}' \rightarrow \boldsymbol{\gamma}$, $\nabla_{\boldsymbol{\gamma}'} = \nabla_{\boldsymbol{\gamma}} - 1$;
    $moment\_vectors \leftarrow \{ \mathbf{I}' \mid \langle 0 \rangle \leq \mathbf{I}' \leq \mathbf{I} \}$;
    $origin\_cond\_moments \leftarrow \{\}$; // all moments required for $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t_0)$
    for (aggregated state $\boldsymbol{\gamma}' \in origin\_states$) **do**
      $relevant\_moments_{tmp} \leftarrow \{\}$; // moments related to $\boldsymbol{\gamma}'$
      for (moment vector $\boldsymbol{I}' \in moment\_vectors$) **do**
        $moment_{tmp} \leftarrow$ INIT_COND_MOMENT($\boldsymbol{\gamma}', \boldsymbol{I}'$);
        append $moment_{tmp}$ to the list $relevant\_moments_{tmp}$;
      end for
      append $relevant\_moments_{tmp}$ to $origin\_cond\_moments$;
    end for
    // by this point, the moments $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t_0)$ depends on are ready
    $\boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t_0) \leftarrow$ apply Eq.(8.36) using $origin\_cond\_moments$ and Sol^{agg} ;
    return $\boldsymbol{M}_{\boldsymbol{I}}(\boldsymbol{\gamma}, t_0)$;
  end if
end function

Algorithm 6 Given an aggregated state $\boldsymbol{\gamma} \in D^{agg}$ and a moment vector $\mathbf{I} > \langle 0 \rangle$, return $\frac{d\, \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\, t}\big|_{t_0}$

function INIT_COND_MOMENT_DER(aggregated state $\boldsymbol{\gamma}$, moment vector $\mathbf{I}$)
  $\mathbb{P}_{t_0}(D)$ // initial distribution over the complete state space ;
  Sol^{agg} // solution of the analysis of D^{agg} by the $(\nabla_{max}+1)$-th order;
  if $(Ord(\mathbf{I}) = 1)$ **then** // Base case
    return 0;
  end if
  if $(\boldsymbol{\gamma} \in D^{agg}_{nz})$ **then** // Base case
    $\frac{d\, \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\, t}\big|_{t_0} \leftarrow$ derive $\frac{d\, \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\, t}\big|_{t_0}$ using $\mathbb{P}_{t_0}(D)$;
    return $\frac{d\, \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma},t)}{d\, t}\big|_{t_0}$;
  end if
  if $(\boldsymbol{\gamma} \in D^{agg}_{z})$ **then**

    $moment\_vectors \leftarrow \{\, \mathbf{X} \mid \langle 0 \rangle \leq \mathbf{X} \leq \mathbf{I} \,\}$;
    $all\_moment\_vectors \leftarrow \{\, \mathbf{X} \mid 1 \leq Ord(\mathbf{X}) \leq (Ord(\mathbf{I})) \,\}$;
    $cond\_expecs \leftarrow \{\}$; // required initial conditional expectations
    $cond\_moments \leftarrow \{\}$; // required initial higher-order conditional moments
    $init\_derivatives \leftarrow \{\}$; // required initial derivatives

    // calculating initial values related to $\boldsymbol{\gamma}$
    $\mu(\boldsymbol{\gamma}, t_0) \leftarrow$ INIT_COND_EXP($\boldsymbol{\gamma}$);
    append $\mu(\boldsymbol{\gamma}, t_0)$ to $cond\_expecs$;

    for (moment_vector $\mathbf{X} \in all\_moment\_vectors$) **do**
      $\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}, t_0) \leftarrow$ INIT_COND_MOMENT($\boldsymbol{\gamma}, \mathbf{X}$);
      append $\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}, t_0)$ to $cond\_moments$;
    end for

    // calculating initial values related to states which enable transitions into $\boldsymbol{\gamma}$
    $origin\_states \leftarrow$ set of $\boldsymbol{\gamma}' \in D^{agg}$ that enable transition $\boldsymbol{\gamma}' \rightarrow \boldsymbol{\gamma}$

26: **for** (aggregated state $\boldsymbol{\gamma}' \in origin\_states$) **do**

27: $\mu(\boldsymbol{\gamma}', t_0) \leftarrow$ INIT_COND_EXP($\boldsymbol{\gamma}'$);
28: append $\mu(\boldsymbol{\gamma}', t_0)$ to *cond_expecs*;

29: **for** (moment_vector $\mathbf{X} \in moment\_vectors$) **do**

30: $\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}', t_0) \leftarrow$ INIT_COND_MOMENT($\boldsymbol{\gamma}', \mathbf{X}$);
31: append $\boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}', t_0)$ to *cond_moments*;

32: **end for**

33: **if** ($\nabla_{\boldsymbol{\gamma}'} < \nabla_{\boldsymbol{\gamma}}$) **then** // if $\boldsymbol{\gamma}'$ has strictly less distance

34: **for** (moment_vector $\mathbf{X} \in moment\_vectors$) **do**

35: $\frac{d\, \boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}', t)}{d\, t}\big|_{t_0} \leftarrow$ INIT_COND_MOMENT_DER($\boldsymbol{\gamma}', \mathbf{X}$);
36: append $\frac{d\, \boldsymbol{M}_{\mathbf{X}}(\boldsymbol{\gamma}', t)}{d\, t}\big|_{t_0}$ to *init_derivatives*;

37: **end for**

38: **end if**

39: **end for**

// all required initial values are stored in the variables *cond_expecs*,
// *cond_moments* and *init_derivatives*

40: $\frac{d\, \boldsymbol{M}_{\mathbf{I}}(\boldsymbol{\gamma}, t)}{d\, t}\big|_{t_0} \leftarrow$ apply Eq.(8.37) using variables Sol^{agg}, *cond_expecs*,
41: *cond_moments* and *init_derivatives*

42: return $\frac{d\, \boldsymbol{M}_{\boldsymbol{I}}(\boldsymbol{\gamma}, t)}{d\, t}\big|_{t_0}$;

43: **end if**

44: **end function**

Chapter 9

Analysis of Client-Server Model Using Higher-Order Conditional Moments

9.1 Introduction

In this chapter, we show an example of the method of higher-order conditional moments where it is applied to a version of our client-server model.

Our model was described in Sec. 7.1. It is initialised with the parameters shown in Table 7.1. The numerical vector used for the construction of the state space is $\boldsymbol{\xi} = \langle \boldsymbol{\xi}^s, \boldsymbol{\xi}^l \rangle = \langle S_i, S_l, S_b, C_t, C_r \rangle$ where $\boldsymbol{\xi}^s = \langle S_i, S_l, S_b \rangle$ and $\boldsymbol{\xi}^l = \langle C_t, C_r \rangle$. We assume that the model starts its evolution from the initial state shown in Eq.7.1.

In Chapter 7, the following analyses were applied to this model. First, we applied our aggregation method. The result is the aggregated state space D_{CS}^{agg} which captures the evolution of the servers only. By analysing D_{CS}^{agg}, we get the transient evolution of the marginal probability distribution $\mathbb{P}_t(\boldsymbol{\beta})$, $\boldsymbol{\beta} \in D_{CS}^{agg}$ where $\boldsymbol{\beta}$ represents any of the configurations that the servers experience.

Second, we applied the analysis of the conditional expectations. The group *Clients* is regarded as a large group and the evolution of the associated state vector $\langle C_t, C_r \rangle$ is studied via the conditional distributions $\mathbb{P}_t(\langle C_t, C_r \rangle \mid \boldsymbol{\beta})$, $\boldsymbol{\beta} \in D^{agg}$. In particular, given the importance of the behaviour of C_r we are interested in projections of these distributions on this variable, i.e. $\mathbb{P}_t(C_r \mid \boldsymbol{\beta})$, $\boldsymbol{\beta} \in D^{agg}$. The analysis of conditional expectations gives us the conditional expectations $\mu(\boldsymbol{\beta}, t)$ and $\mu_{(C_r)}(\boldsymbol{\beta}, t)$.

Moreover, we observed that the servers' breakdowns and fixes have significant impact on the evolution of the clients. The servers exhibit three modes of operation, corresponding to $S_b = 0$, $S_b = 1$ and $S_b = 2$. Using the method of conditional expectation, we could obtain $\mu(S_b = 0,t)$, $\mu(S_b = 1,t)$ and $\mu(S_b = 2,t)$, which capture the mean behaviour of the clients in each of these operational modes and allow us to quantitatively measure the impact of the dynamics of the servers on the clients.

9.2 Description of Moments

We extend our analysis by including conditional moments of the second order, that is, $\boldsymbol{M}_{\mathbf{I_2}}(\boldsymbol{\beta})$, $\boldsymbol{\beta} \in D^{agg}$. The vector $\boldsymbol{\xi}^l = \langle C_t, C_r \rangle$ has two state variables. Thus, the moment vectors of an order K are obtained by solving the equation $I_1 + I_2 = K$ with the restriction that I_1, I_2 are positive integers. The solutions for varying orders of K are shown in Fig. 9.1. Focusing on the order two, we see that there are three moment vectors that need to be considered: $\boldsymbol{I} = \langle 2,0 \rangle$, $\boldsymbol{I} = \langle 1,1 \rangle$ and $\boldsymbol{I} = \langle 0,2 \rangle$.

Using Def. 35, for any aggregated state $\boldsymbol{\beta}$ the conditional moments $\boldsymbol{M}_{\langle 2,0 \rangle}(\boldsymbol{\beta},t)$ and $\boldsymbol{M}_{\langle 0,2 \rangle}(\boldsymbol{\beta},t)$ are defined as:

$$\begin{aligned} \boldsymbol{M}_{\langle \mathbf{2},\mathbf{0} \rangle}(\boldsymbol{\beta}) &= \mathbb{E}_t \left[\Big(\langle C_t, C_r \rangle - \mu(\boldsymbol{\beta},t) \Big)^{\langle 2,0 \rangle} \mid \boldsymbol{\beta} \right] = \mathbb{E}_t \left[\Big(C_t - \mu_{(C_t)}(\boldsymbol{\beta},t) \Big)^2 \mid \boldsymbol{\beta} \right] \\ \boldsymbol{M}_{\langle \mathbf{0},\mathbf{2} \rangle}(\boldsymbol{\beta}) &= \mathbb{E}_t \left[\Big(\langle C_t, C_r \rangle - \mu(\boldsymbol{\beta},t) \Big)^{\langle 0,2 \rangle} \mid \boldsymbol{\beta} \right] = \mathbb{E}_t \left[\Big(C_r - \mu_{(C_r)}(\boldsymbol{\beta},t) \Big)^2 \mid \boldsymbol{\beta} \right] \end{aligned} \tag{9.1}$$

Respectively, these correspond to the *conditional variances*[1] of the variables C_t and C_r. For convenience, they will be denoted by $\mathbb{VAR}_t [C_t \mid \boldsymbol{\beta}]$ and $\mathbb{VAR}_t [C_r \mid \boldsymbol{\beta}]$. In addition, the conditional moment associated with the moment vector $\langle 1,1 \rangle$ is defined as:

$$\begin{aligned} \boldsymbol{M}_{\langle \mathbf{1},\mathbf{1} \rangle}(\boldsymbol{\beta}) = \mathbb{E}_t \left[\Big(\langle C_t, C_r \rangle - \mu(\boldsymbol{\beta},t) \Big)^{\langle 1,1 \rangle} \mid \boldsymbol{\beta} \right] = \\ \mathbb{E}_t \left[\Big(C_t - \mu_{(C_t)}(\boldsymbol{\beta},t) \Big) \Big(C_r - \mu_{(C_r)}(\boldsymbol{\beta},t) \Big) \mid \boldsymbol{\beta} \right] \end{aligned} \tag{9.2}$$

which captures the *conditional covariance* of the variables C_t and C_r.

Since there are six aggregated states in D^{agg}_{CS} and three vectors of the second order for each aggregated state, in total we have 18 second-order moments to study. These

[1] In general, for any random variable X: $\mathbb{VAR}_t [X \mid \boldsymbol{\beta}] = \mathbb{E}_t \left[\left(X - \mathbb{E}_t [X \mid \boldsymbol{\beta}] \right)^2 \mid \boldsymbol{\beta} \right]$.

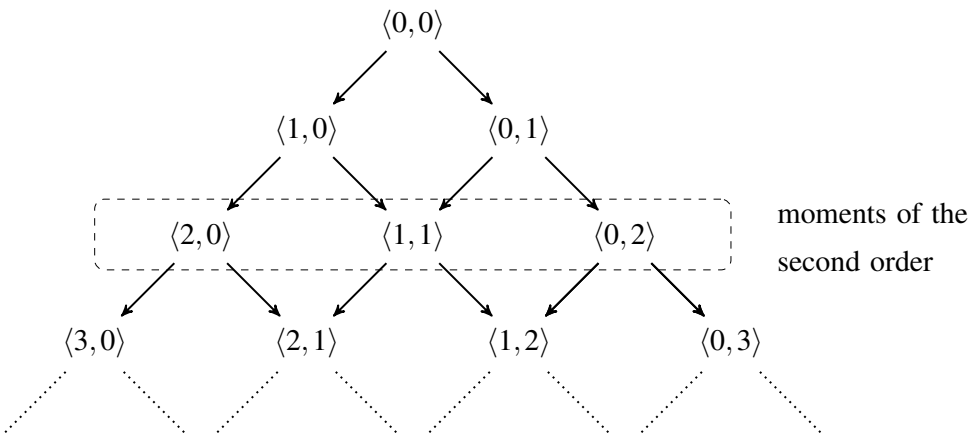


Figure 9.1: The moments related to the client-servers system where $\boldsymbol{\xi}^l = \langle C_t, C_r \rangle$. In this example, we focus on the second-order moments which capture the variances and covariance of the state variables C_t and C_r.

	$\boldsymbol{M}_{\langle 2,0\rangle}(\boldsymbol{\beta},t)$	$\boldsymbol{M}_{\langle 1,1\rangle}(\boldsymbol{\beta},t)$	$\boldsymbol{M}_{\langle 0,2\rangle}(\boldsymbol{\beta},t)$
$\langle 2,0,0\rangle$	$\boldsymbol{M}_{\langle 2,0\rangle}(\langle 2,0,0\rangle,t)$	$\boldsymbol{M}_{\langle 1,1\rangle}(\langle 2,0,0\rangle,t)$	$\boldsymbol{M}_{\langle 0,2\rangle}(\langle 2,0,0\rangle,t)$
$\langle 1,1,0\rangle$	$\boldsymbol{M}_{\langle 2,0\rangle}(\langle 1,1,0\rangle,t)$	$\boldsymbol{M}_{\langle 1,1\rangle}(\langle 1,1,0\rangle,t)$	$\boldsymbol{M}_{\langle 0,2\rangle}(\langle 1,1,0\rangle,t)$
$\langle 0,2,0\rangle$	$\boldsymbol{M}_{\langle 2,0\rangle}(\langle 0,2,0\rangle,t)$	$\boldsymbol{M}_{\langle 1,1\rangle}(\langle 0,2,0\rangle,t)$	$\boldsymbol{M}_{\langle 0,2\rangle}(\langle 0,2,0\rangle,t)$
$\langle 1,0,1\rangle$	$\boldsymbol{M}_{\langle 2,0\rangle}(\langle 1,0,1\rangle,t)$	$\boldsymbol{M}_{\langle 1,1\rangle}(\langle 1,0,1\rangle,t)$	$\boldsymbol{M}_{\langle 0,2\rangle}(\langle 1,0,1\rangle,t)$
$\langle 0,1,1\rangle$	$\boldsymbol{M}_{\langle 2,0\rangle}(\langle 0,1,1\rangle,t)$	$\boldsymbol{M}_{\langle 1,1\rangle}(\langle 0,1,1\rangle,t)$	$\boldsymbol{M}_{\langle 0,2\rangle}(\langle 0,1,1\rangle,t)$
$\langle 0,0,2\rangle$	$\boldsymbol{M}_{\langle 2,0\rangle}(\langle 0,0,2\rangle,t)$	$\boldsymbol{M}_{\langle 1,1\rangle}(\langle 0,0,2\rangle,t)$	$\boldsymbol{M}_{\langle 0,2\rangle}(\langle 0,0,2\rangle,t)$

Table 9.1: The second-order moments related to the state vector $\langle C_t, C_r \rangle$ and aggregated states $\boldsymbol{\beta} \in D_{CS}^{agg}$.

are shown in Table. 9.1. Having defined the moments, we now describe how their evolution is derived by the method of higher-order moments.

9.3 Second-Order Moment Equations

The evolution of the second-order moments is studied by setting up a system of DAEs, where for each moment one instance of Eq.(8.34) is constructed. This gives rise to 18 equations. In order to build a closed system of equations, we need to add the equations related to the conditional expectations $\mu(\boldsymbol{\beta},t)$ and the marginal probability distributions $\mathbb{P}_t(\boldsymbol{\beta})$ to the system of DAEs. This will add a further 18 equations. Therefore, the

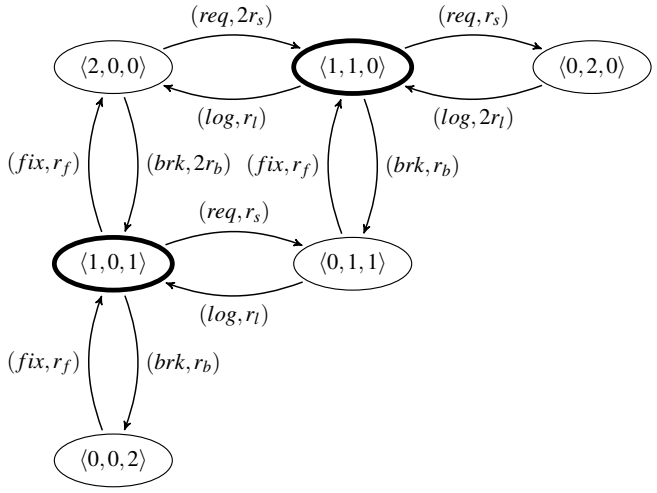


Figure 9.2: The aggregated state space of the client-server system with two servers.

analysis of the second-order moments gives rise to 36 equations.

We provide two examples from the equations related to the second-order moments. The first is related to state $\langle 1,1,0\rangle$ and moment $\mathbb{VAR}_t\left[C_r \,|\, \langle 1,1,0\rangle\right]$, and the second corresponds to state $\langle 1,0,1\rangle$ and moment $\boldsymbol{M}_{\langle 1,1\rangle}(\langle 1,0,1\rangle, t)$. To better understand the structure of these equations, we recommend considering the structure of Eq.(8.34) and the model's aggregated state space (shown in Fig.9.2). In particular focus on the transitions in and out of states $\langle 1,1,0\rangle$ and $\langle 1,0,1\rangle$. The equation that captures the evolution of $\mathbb{VAR}_t\left[C_r \,|\, \langle 1,1,0\rangle\right]$ is:

$$
\begin{aligned}
&\boldsymbol{M}_{\langle 0,2\rangle}(\langle 1,1,0\rangle, t) \cdot \frac{d\,\mathbb{P}_t(\langle 1,1,0\rangle)}{d\,t} + \frac{d\,\boldsymbol{M}_{\langle 0,2\rangle}(\langle 1,1,0\rangle, t)}{d\,t} \cdot \mathbb{P}_t(\langle 1,1,0\rangle) \;\approx \\
&\quad - \mathbb{P}_t(\langle 1,1,0\rangle) \cdot \boldsymbol{M}_{\langle 0,2\rangle}(\langle 1,1,0\rangle, t) \cdot \left[r_{log}(\langle 1,1,0\rangle) + r_{req}(\langle 1,1,0\rangle) + r_{brk}(\langle 1,1,0\rangle) \right] \\
&\quad + r_{req}(\langle 2,0,0\rangle) \cdot \mathbb{P}_t(\langle 2,0,0\rangle) \cdot \Bigg[\\
&\qquad \binom{\langle 0,2\rangle}{\langle 0,2\rangle} \Big(\mu(\langle 2,0,0\rangle, t) + \mathcal{V}^l_{req} - \mu(\langle 1,1,0\rangle, t) \Big)^{\langle 0,0\rangle} \cdot \boldsymbol{M}_{\langle 0,2\rangle}(\langle 2,0,0\rangle, t) \\
&\qquad + \binom{\langle 0,2\rangle}{\langle 0,1\rangle} \Big(\mu(\langle 2,0,0\rangle, t) + \mathcal{V}^l_{req} - \mu(\langle 1,1,0\rangle, t) \Big)^{\langle 0,1\rangle} \cdot \boldsymbol{M}_{\langle 0,1\rangle}(\langle 2,0,0\rangle, t) \\
&\qquad + \binom{\langle 0,2\rangle}{\langle 0,0\rangle} \Big(\mu(\langle 2,0,0\rangle, t) + \mathcal{V}^l_{req} - \mu(\langle 1,1,0\rangle, t) \Big)^{\langle 0,2\rangle} \cdot \boldsymbol{M}_{\langle 0,0\rangle}(\langle 2,0,0\rangle, t) \quad \Bigg]
\end{aligned}
$$

$$
\begin{aligned}
&+ r_{log}(\langle 0,2,0\rangle) \cdot \mathbb{P}_t(\langle 0,2,0\rangle) \cdot \Bigg[\\
&\quad \binom{\langle 0,2\rangle}{\langle 0,2\rangle} \Big(\mu(\langle 0,2,0\rangle,t) - \mu(\langle 1,1,0\rangle,t)\Big)^{\langle 0,0\rangle} \cdot \boldsymbol{M}_{\langle 0,2\rangle}(\langle 0,2,0\rangle,t) \\
&\quad + \binom{\langle 0,2\rangle}{\langle 0,1\rangle} \cdot \Big(\mu(\langle 0,2,0\rangle,t) - \mu(\langle 1,1,0\rangle,t)\Big)^{\langle 0,1\rangle} \cdot \boldsymbol{M}_{\langle 0,1\rangle}(\langle 0,2,0\rangle,t) \\
&\quad + \binom{\langle 0,2\rangle}{\langle 0,0\rangle} \Big(\mu(\langle 0,2,0\rangle,t) - \mu(\langle 1,1,0\rangle,t)\Big)^{\langle 0,2\rangle} \cdot \boldsymbol{M}_{\langle 0,0\rangle}(\langle 0,2,0\rangle,t) \quad \Bigg] \\
&+ r_{fix}(\langle 0,1,1\rangle) \cdot \mathbb{P}_t(\langle 0,1,1\rangle) \cdot \Bigg[\\
&\quad \binom{\langle 0,2\rangle}{\langle 0,2\rangle} \cdot \Big(\mu(\langle 0,1,1\rangle,t) - \mu(\langle 1,1,0\rangle,t)\Big)^{\langle 0,0\rangle} \cdot \boldsymbol{M}_{\langle 0,2\rangle}(\langle 0,1,1\rangle,t) \\
&\quad + \binom{\langle 0,2\rangle}{\langle 0,1\rangle} \Big(\mu(\langle 0,1,1\rangle,t) - \mu(\langle 1,1,0\rangle,t)\Big)^{\langle 0,1\rangle} \cdot \boldsymbol{M}_{\langle 0,1\rangle}(\langle 0,1,1\rangle,t) \\
&\quad + \binom{\langle 0,2\rangle}{\langle 0,0\rangle} \Big(\mu(\langle 0,1,1\rangle,t) - \mu(\langle 1,1,0\rangle,t)\Big)^{\langle 0,2\rangle} \cdot \boldsymbol{M}_{\langle 0,0\rangle}(\langle 0,1,1\rangle,t) \quad \Bigg] \\
&+ \mathbb{P}_t(\langle 1,1,0\rangle) \cdot \Bigg[\\
&\quad \binom{\langle 0,2\rangle}{\langle 0,1\rangle} \cdot \Big(\mathcal{V}^{l}_{think}\Big)^{\langle 0,1\rangle} \cdot \\
&\qquad \Big(r_t \cdot \mu_{(C_t)}(\langle 1,1,0\rangle,t) \cdot \boldsymbol{M}_{\langle 0,1\rangle}(\langle 1,1,0\rangle,t) + r_t \cdot \boldsymbol{M}_{\langle\langle 1,1\rangle\rangle}(\langle 1,1,0\rangle,t)\Big) \\
&\quad + \binom{\langle 0,2\rangle}{\langle 0,0\rangle} \cdot \Big(\mathcal{V}^{l}_{think}\Big)^{\langle 0,2\rangle} \cdot \\
&\qquad \Big(r_t \cdot \mu_{(C_t)}(\langle 1,1,0\rangle,t) \cdot \boldsymbol{M}_{\langle 0,0\rangle}(\langle 1,1,0\rangle,t) + r_t \cdot \boldsymbol{M}_{\langle\langle 1,0\rangle\rangle}(\langle 1,1,0\rangle,t)\Big) \quad \Bigg] \\
&- \mathbb{P}_t(\langle 1,1,0\rangle) \cdot \left(2 \cdot \frac{d\,\mu_{(C_r)}(\langle 1,1,0\rangle,t)}{d\,t} \cdot \boldsymbol{M}_{\langle 0,1\rangle}(\langle 1,1,0\rangle,t)\right)
\end{aligned}
\tag{9.3}
$$

The solution enables us to measure the deviation of C_r around the conditional expectation $\mathbb{E}_t\,[C_r \mid \langle 1,1,0\rangle\,]$, an important measure related to the conditional distribution $\mathbb{P}_t(\,\langle C_t, C_r\rangle \mid \langle 1,1,0\rangle\,)$ when projected on C_r.

As the second example, we present the equation constructed for $\boldsymbol{M}_{\langle 1,1\rangle}(\langle 1,0,1\rangle,t)$:

$$
\boldsymbol{M}_{\langle 1,1\rangle}(\langle 1,0,1\rangle,t) \cdot \frac{d\,\mathbb{P}_t(\langle 1,0,1\rangle)}{d\,t} + \frac{d\,\boldsymbol{M}_{\langle 1,1\rangle}(\langle 1,0,1\rangle,t)}{d\,t} \cdot \mathbb{P}_t(\langle 1,0,1\rangle) \;\approx
$$

$$
\begin{aligned}
&- \ \mathbb{P}_t(\langle 1,0,1\rangle)\cdot \boldsymbol{M}_{\langle 1,1\rangle}(\langle 1,0,1\rangle,t)\cdot\Big[r_{fix}(\langle 1,0,1\rangle)+r_{req}(\langle 1,0,1\rangle)+r_{brk}(\langle 1,0,1\rangle)\Big]\\
&+ r_{brk}(\langle 2,0,0\rangle)\cdot\mathbb{P}_t(\langle 2,0,0\rangle)\cdot\Bigg[\\
&\qquad \binom{\langle 1,1\rangle}{\langle 1,1\rangle}\Big(\mu(\langle 2,0,0\rangle,t)-\mu(\langle 1,0,1\rangle,t)\Big)^{\langle 0,0\rangle}\cdot\boldsymbol{M}_{\langle 1,1\rangle}(\langle 2,0,0\rangle,t)\\
&\qquad +\binom{\langle 1,1\rangle}{\langle 1,0\rangle}\cdot\Big(\mu(\langle 2,0,0\rangle,t)-\mu(\langle 1,0,1\rangle,t)\Big)^{\langle 0,1\rangle}\cdot\boldsymbol{M}_{\langle 1,0\rangle}(\langle 2,0,0\rangle,t)\\
&\qquad +\binom{\langle 1,1\rangle}{\langle 0,1\rangle}\cdot\Big(\mu(\langle 2,0,0\rangle,t)-\mu(\langle 1,0,1\rangle,t)\Big)^{\langle 1,0\rangle}\cdot\boldsymbol{M}_{\langle 0,1\rangle}(\langle 2,0,0\rangle,t)\Big)\\
&\qquad +\binom{\langle 1,1\rangle}{\langle 0,0\rangle}\Big(\mu(\langle 2,0,0\rangle,t)-\mu(\langle 1,0,1\rangle,t)\Big)^{\langle 1,1\rangle}\cdot\boldsymbol{M}_{\langle 0,0\rangle}(\langle 2,0,0\rangle,t)\qquad\Bigg]\\
&+ r_{log}(\langle 0,1,1\rangle)\cdot\mathbb{P}_t(\langle 0,1,1\rangle)\cdot\Bigg[\\
&\qquad \binom{\langle 1,1\rangle}{\langle 1,1\rangle}\cdot\Big(\mu(\langle 0,1,1\rangle,t)-\mu(\langle 1,0,1\rangle,t)\Big)^{\langle 0,0\rangle}\cdot\boldsymbol{M}_{\langle 1,1\rangle}(\langle 0,1,1\rangle,t)\\
&\qquad +\binom{\langle 1,1\rangle}{\langle 1,0\rangle}\cdot\Big(\mu(\langle 0,1,1\rangle,t)-\mu(\langle 1,0,1\rangle,t)\Big)^{\langle 0,1\rangle}\cdot\boldsymbol{M}_{\langle 1,0\rangle}(\langle 0,1,1\rangle,t)\\
&\qquad +\binom{\langle 1,1\rangle}{\langle 0,1\rangle}\cdot\Big(\mu(\langle 0,1,1\rangle,t)-\mu(\langle 1,0,1\rangle,t)\Big)^{\langle 1,0\rangle}\cdot\boldsymbol{M}_{\langle 0,1\rangle}(\langle 0,1,1\rangle,t)\\
&\qquad +\binom{\langle 1.1\rangle}{\langle 0,0\rangle}\cdot\Big(\mu(\langle 0,1,1\rangle,t)-\mu(\langle 1,0,1\rangle,t)\Big)^{\langle 1,1\rangle}\cdot\boldsymbol{M}_{\langle 0,0\rangle}(\langle 0,1,1\rangle,t)\qquad\Bigg]\\
&+ r_{fix}(\langle 0,0,2\rangle)\cdot\mathbb{P}_t(\langle 0,0,2\rangle)\cdot\Bigg[\\
&\qquad \binom{\langle 1,1\rangle}{\langle 1,1\rangle}\cdot\Big(\mu(\langle 0,0,2\rangle,t)-\mu(\langle 1,0,1\rangle,t)\Big)^{\langle 0,0\rangle}\cdot\boldsymbol{M}_{\langle\langle 1,1\rangle\rangle}(\langle 0,0,2\rangle,t)\\
&\qquad +\binom{\langle 1,1\rangle}{\langle 0,1\rangle}\cdot\Big(\mu(\langle 0,0,2\rangle,t)-\mu(\langle 1,0,1\rangle,t)\Big)^{\langle 1,0\rangle}\cdot\boldsymbol{M}_{\langle\langle 1,0\rangle\rangle}(\langle 0,0,2\rangle,t)\\
&\qquad +\binom{\langle 1,1\rangle}{\langle 1,0\rangle}\cdot\Big(\mu(\langle 0,0,2\rangle,t)-\mu(\langle 1,0,1\rangle,t)\Big)^{\langle 0,1\rangle}\cdot\boldsymbol{M}_{\langle\langle 1,0\rangle\rangle}(\langle 0,0,2\rangle,t)\\
&\qquad +\binom{\langle 1,1\rangle}{\langle 0,0\rangle}\cdot\Big(\mu(\langle 0,0,2\rangle,t)-\mu(\langle 1,0,1\rangle,t)\Big)^{\langle 1,1\rangle}\cdot\boldsymbol{M}_{\langle 0,0\rangle}(\langle 0,0,2\rangle,t)\qquad\Bigg]\\
&+\mathbb{P}_t(\langle 1,0,1\rangle)\cdot\Bigg[
\end{aligned}
$$

$$
\begin{aligned}
&\binom{\langle 1,1\rangle}{\langle 0,0\rangle} \cdot \left(\mathcal{V}^{l}_{think}\right)^{\langle 1,1\rangle} \cdot \\
&\qquad \left(r_t \cdot \mu_{(C_t)}(\langle 1,0,1\rangle,t) \cdot \boldsymbol{M}_{\langle 0,0\rangle}(\langle 1,0,1\rangle,t) + r_t \cdot \boldsymbol{M}_{\langle\langle 1,0\rangle\rangle}(\langle 1,0,1\rangle,t)\right) \\
&+ \binom{\langle 1,1\rangle}{\langle 0,1\rangle} \cdot \left(\mathcal{V}^{l}_{think}\right)^{\langle 1,0\rangle} \cdot \\
&\qquad \left(r_t \cdot \mu_{(C_t)}(\langle 1,0,1\rangle,t) \cdot \boldsymbol{M}_{\langle 0,1\rangle}(\langle 1,0,1\rangle,t) + r_t \cdot \boldsymbol{M}_{\langle\langle 1,1\rangle\rangle}(\langle 1,0,1\rangle,t)\right) \\
&+ \binom{\langle 1,1\rangle}{\langle 1,0\rangle} \left(\mathcal{V}^{l}_{think}\right)^{\langle 0,1\rangle} \\
&\qquad \left(r_t \cdot \mu_{(C_t)}(\langle 1,0,1\rangle,t) \cdot \boldsymbol{M}_{\langle 1,0\rangle}(\langle 1,0,1\rangle,t) + r_t \cdot \boldsymbol{M}_{\langle\langle 2,0\rangle\rangle}(\langle 1,0,1\rangle,t)\right) \Bigg]
\end{aligned}
$$

$$
\begin{aligned}
-\mathbb{P}_t(\langle 1,0,1\rangle) \cdot \Bigg[& 1 \cdot \frac{d\,\mu_{(C_t)}(\langle 1,0,1\rangle,t)}{d\,t} \cdot \boldsymbol{M}_{\langle\langle 0,1\rangle\rangle}(\langle 1,0,1\rangle,t) \\
& + 1 \cdot \frac{d\,\mu_{(C_r)}(\langle 1,0,1\rangle,t)}{d\,t} \cdot \boldsymbol{M}_{\langle\langle 1,0\rangle\rangle}(\langle 1,0,1\rangle,t) \Bigg]
\end{aligned} \tag{9.4}
$$

The solution of this equation shows the transient evolution of the covariance of C_t and C_r. We expect these variables to be always negatively correlated, as the increase in one implies a decrease in the other and vice versa (this is shown in Appendix E.1).

9.4 Initial Values

In order to solve the system of DAEs, a set of initial values needs to be specified by the modeller. This set consists of an initial probability distribution $\mathbb{P}_{t_0}(\boldsymbol{\beta})$ over D^{agg}_{CS}, the initial conditional expectations $\mu_{(C_t)}(\boldsymbol{\beta},t_0)$ and $\mu_{(C_r)}(\boldsymbol{\beta},t_0)$ for $\boldsymbol{\beta} \in D^{agg}_{CS}$ and the initial second-order moments $\boldsymbol{M}_{\boldsymbol{I}}(\boldsymbol{\beta},t_0)$, for $\boldsymbol{\beta} \in D^{agg}_{CS}$ and $\boldsymbol{I} \in \{\langle 2,0\rangle, \langle 1,1\rangle, \langle 0,2\rangle\}$. One way to find these values is the rigorous method presented in Sec. 8.4. However, due to its complexity, we do not use this approach and instead, apply the alternative method of Sec. 8.4.3, which uses stochastic simulations to derive the initial values approximately. The steps performed are the following:

1. We choose $t_{init} = 0.1$ as a point of time which is close to $t = 0$ and satisfies the condition $\forall\, \boldsymbol{\beta} \in D^{agg}_{CS} : \mathbb{P}_{t_{init}}(\boldsymbol{\beta}) \neq 0$. By the analysis of the aggregated state space for $t = 0.1$, we derive the probability distribution $\mathbb{P}_{0.1}(\boldsymbol{\beta})$, which is shown in Fig. 9.3.

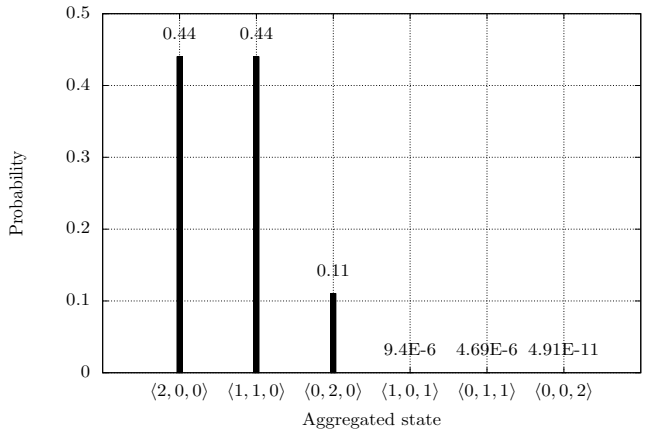


Figure 9.3: The approximate initial distribution over D_{CS}^{agg}, $\mathbb{P}_{0.1}(\boldsymbol{\beta})$, derived by the analysis of the C-K equations underlying the aggregated state space. The model starts its evolution from the state $\langle 2,0,0\rangle$.

2. Using stochastic simulation, we obtain the approximate conditional distribution $\mathbb{P}_{0.1}(\langle C_t, C_r\rangle \mid \boldsymbol{\beta})$ for each of the aggregated states in D_{CS}^{agg}. Here, we ran 10000 simulation runs, which, based on prior knowledge about the model, is sufficient to achieve the marginal error of less than 1 percent with respect to the exact distribution.

3. By applying Def. 28 and using the conditional distributions above, the initial values $\mu_{(C_r)}(\boldsymbol{\beta}, 0.1)$, $\boldsymbol{\beta} \in D_{CS}^{agg}$ are derived. These are shown in Table 9.2, the second and third columns.

4. Finally, the initial conditional expectations and the probability distribution over the aggregated states is substituted into instances of Eq.(34) to derive $\boldsymbol{M_I}(\boldsymbol{\beta}, 0.1)$ for $\boldsymbol{I} \in \{\langle 2,0\rangle, \langle 1,1\rangle, \langle 0,2\rangle\}$ and $\boldsymbol{\beta} \in D_{CS}^{agg}$. The result of this calculation is shown in Table 9.2, the fourth to the last columns.

By calculating the initial values as above, we are essentially replacing the analysis which starts from $t = 0$ by one starting from $t_{init} = 0.1$.

9.5 Solution

As stated, the complete system of DAEs constructed for the second-order moments consists of 36 equations. The structure of these equations is complex, as illustrated

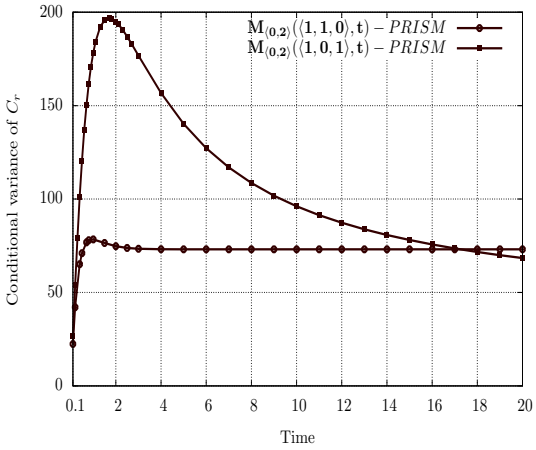
state $\boldsymbol{\beta}$	$\mu_{C_t}(\boldsymbol{\beta},0.1)$	$\mu_{C_r}(\boldsymbol{\beta},0.1)$	$\boldsymbol{M}_{\langle 2,0\rangle}(\boldsymbol{\beta},0.1)$	$\boldsymbol{M}_{\langle 1,1\rangle}(\boldsymbol{\beta},0.1)$	$\boldsymbol{M}_{\langle 0,2\rangle}(\boldsymbol{\beta},0.1)$
$\langle 2,0,0\rangle$	39.89	10.1	23.41	-23.41	23.41
$\langle 1,1,0\rangle$	140.43	9.56	22.43	-22.43	22.43
$\langle 0,2,0\rangle$	140.98	9.01	21.57	-21.57	21.57
$\langle 1,0,1\rangle$	136.20	13.79	26.51	-26.51	26.51
$\langle 0,1,1\rangle$	136.84	13.15	26.18	-26.18	26.18
$\langle 0,0,2\rangle$	132.47	17.52	29.31	-29.31	29.31

Table 9.2: The conditional expectations and second-order moments calculated approximately using the stochastic simulations. These are used as the initial values for the DAEs that we construct when analysing the second-order moments.

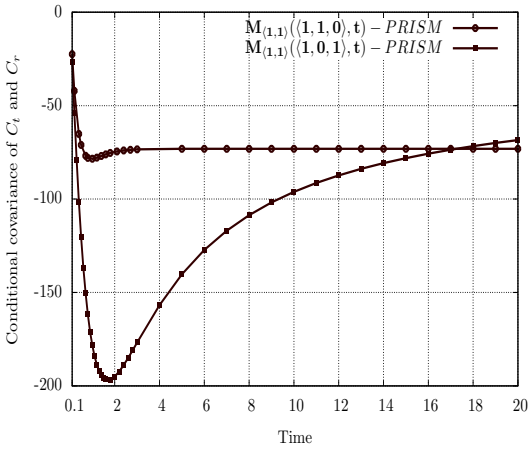
by the representative equations shown in Eq.(9.3) and Eq.(9.4). For almost all models these equations are too complex to be built by hand; this suggests that it is imperative to have software support for the derivation of these equations. The presented algorithm (Alg.4 on Page 184) shows that the derivations are amenable to development in software, but time constraints within our project precluded the tool's development and the construction and solution of the MCM equations remains a task for future work.

Nevertheless, in spite of not having the MCM software tool, we are still able to study the evolution of our second-order moments. Here, instead of using the MCM equations, we obtain these moments via the the computationally expensive analysis of the complete state space performed using PRISM. We chose to study the transient evolution of the moments $\boldsymbol{M}_{\boldsymbol{I}}(\boldsymbol{\beta},t)$ shown in Table 9.1 for the time period of $t = [t_{init}, 20]$. For this purpose, first the model's complete state space is constructed. Then, the evolution of the associated complete probability distribution is derived using uniformisation. Finally, Eq.(9.1), Eq.(9.2) are used to derive the transient evolution of our second-order moments. In spite of being expensive, this experiment allows us to observe the importance of the derivation of the MCM equations, especially in terms of the very useful information they capture.

Among the states $\boldsymbol{\beta} \in D_{CS}^{agg}$ and moment vectors $\boldsymbol{I} \in \{\langle 2,0\rangle, \langle 1,1\rangle, \langle 0,2\rangle\}$, we focus on the states $\langle 1,1,0\rangle$ and $\langle 1,0,1\rangle$ and moment vectors $\langle 1,1\rangle$ and $\langle 0,2\rangle$. In Fig 9.4, the transient evolution of the associated representative moments $\boldsymbol{M}_{\langle 1,1\rangle}(\langle 1,1,0\rangle,t)$, $\boldsymbol{M}_{\langle 1,1\rangle}(\langle 1,0,1\rangle,t)$, $\mathbb{VAR}_t\,[C_r \,|\, \langle 1,1,0\rangle\,] = \boldsymbol{M}_{\langle 0,2\rangle}(\langle 1,1,0\rangle,t)$ and $\mathbb{VAR}_t\,[C_r \,|\, \langle 1,0,1\rangle\,] = \boldsymbol{M}_{\langle 0,2\rangle}(\langle 1,0,1\rangle,t)$ is reported. When combined with the conditional expectations, the second-order moments enable us to gain a richer representation of the conditional dis-



(A) Evolution of the second-order moments $\mathbb{VAR}_t[C_r \mid \langle 1,1,0\rangle]$ and $\mathbb{VAR}_t[C_r \mid \langle 1,0,1\rangle]$ derived using PRISM.



(B) Evolution of the second order moments $\boldsymbol{M}_{\langle 1,1\rangle}(\langle 1,1,0\rangle, t)$, $\boldsymbol{M}_{\langle 1,1\rangle}(\langle 1,0,1\rangle, t)$ derived using PRISM. As expected, the correlation is always negative.

Figure 9.4: Transient evolution of the second-order moments $\mathbb{VAR}_t[C_r \mid \langle 1,1,0\rangle]$, $\mathbb{VAR}_t[C_r \mid \langle 1,0,1\rangle]$, $\boldsymbol{M}_{\langle 1,1\rangle}(\langle 1,1,0\rangle, t)$ and $\boldsymbol{M}_{\langle 1,1\rangle}(\langle 1,0,1\rangle, t)$ derived by the computationally expensive analysis of the complete state space. We propose a conjecture that the MCM equations should enable us to obtain the same solution significantly quicker.

tributions $\mathbb{P}_t(\langle C_t, C_r\rangle \mid \boldsymbol{\beta})$. For instance, the conditional variances $\mathbb{VAR}_t[C_r \mid \langle 1,1,0\rangle]$ and $\mathbb{VAR}_t[C_r \mid \langle 1,0,1\rangle]$ capture the widths of the distributions $\mathbb{P}_t(C_r \mid \langle 1,1,0\rangle)$ and $\mathbb{P}_t(C_r \mid \langle 1,0,1\rangle)$ around $\mu_{(C_r)}(\langle 1,1,0\rangle, t)$ and $\mu_{(C_r)}(\langle 1,0,1\rangle, t)$, a measure showing how dispersed the variable C_r is when the servers are in states $\langle 1,1,0\rangle$ and $\langle 1,0,1\rangle$.

9.6 Capturing the Impact of Server Breakdowns

The servers exhibit three modes of operation which are shown in Fig. 7.6. In studying the behaviour of the clients, we have already shown how the conditional moments $\mu_{(C_r)}(S_b = 0, t)$, $\mu_{(C_r)}(S_b = 1, t)$ and $\mu_{(C_r)}(S_b = 2, t)$ were calculated by the method of conditional expectations. These show the mean behaviour of the clients in each of the distinct operational modes.

The conditional second-order moments calculated over the aggregated states are the basis for deriving the conditional moments $\mathbb{VAR}[C_r \mid S_b = 0]$, $\mathbb{VAR}[C_r \mid S_b = 1]$ and $\mathbb{VAR}[C_r \mid S_b = 2]$ defined over the three modes of operation. These moments

capture the widths of the conditional distributions $\mathbb{P}_t(C_r \mid S_b = 0)$, $\mathbb{P}_t(C_r \mid S_b = 1)$ and $\mathbb{P}_t(C_r \mid S_b = 2)$. Obtaining them is important, since it is likely that the clients do not behave exactly as *expected* and their stochastic behaviour may potentially exhibit deviations from the conditionally averaged evolutions.

First, we focus on the derivation of $\mathbb{VAR}[C_r \mid S_b = 0]$. Using Def. (9.1), we have:

$$\begin{aligned}\mathbb{VAR}[C_r \mid S_b = 0] &= \sum_{\langle S_i,S_l,S_b,C_t,C_r\rangle \in D} \Big(C_r - \mu_{(C_r)}(S_b = 0,t)\Big)^2 \cdot \\ &\qquad \mathbb{P}_t(\langle S_i,S_l,S_b,C_t,C_r\rangle \mid S_b = 0) \\ &= \sum_{\langle S_i,S_l,S_b,C_t,C_r\rangle \in D} \frac{\Big(C_r - \mu_{(C_r)}(S_b = 0,t)\Big)^2 \cdot \mathbb{P}_t(\langle S_i,S_l,0,C_t,C_r\rangle)}{\mathbb{P}_t(S_b = 0)}\end{aligned}$$

Here, the summation on the right hand side sums over all states in the state space. We can expand this so that the summations happen with respect to each sub-chain separately:

$$\begin{aligned}\mathbb{VAR}[C_r \mid S_b = 0] &= \sum_{\langle 2,0,0,C_t,C_r\rangle \in D} \frac{\Big(C_r - \mu_{(C_r)}(S_b = 0,t)\Big)^2 \cdot \mathbb{P}_t(\langle 2,0,0,C_t,C_r\rangle)}{\mathbb{P}_t(S_b = 0)} \\ &+ \sum_{\langle 1,1,0,C_t,C_r\rangle \in D} \frac{\Big(C_r - \mu_{(C_r)}(S_b = 0,t)\Big)^2 \cdot \mathbb{P}_t(\langle 1,1,0,C_t,C_r\rangle)}{\mathbb{P}_t(S_b = 0)} \\ &+ \sum_{\langle 0,2,0,C_t,C_r\rangle \in D} \frac{\Big(C_r - \mu_{(C_r)}(S_b = 0,t)\Big)^2 \cdot \mathbb{P}_t(\langle 0,2,0,C_t,C_r\rangle)}{\mathbb{P}_t(S_b = 0)}\end{aligned}$$

By reformulating the probability terms in the conditional form, adding and subtracting $\mu_{(C_r)}(\langle 2,0,0\rangle,t)$, $\mu_{(C_r)}(\langle 1,1,0\rangle,t)$, $\mu_{(C_r)}(\langle 0,2,0\rangle,t)$ in the first, second and third terms on the right hand side and using the vectorial binomial expansion we get:

$$\begin{aligned}&\mathbb{VAR}[C_r \mid S_b = 0] = \\ &+ \frac{\mathbb{P}_t(\langle 2,0,0\rangle)}{\mathbb{P}_t(S_b = 0)} \Big[\mathbb{VAR}[C_r \mid \langle 2,0,0\rangle] + \big(\mu_{(C_r)}(\langle 2,0,0\rangle,t) - \mu_{(C_r)}(S_b = 0,t)\big)^2\Big] \\ &+ \frac{\mathbb{P}_t(\langle 1,1,0\rangle)}{\mathbb{P}_t(S_b = 0)} \Big[\mathbb{VAR}[C_r \mid \langle 1,1,0\rangle] + \big(\mu_{(C_r)}(\langle 1,1,0\rangle,t) - \mu_{(C_r)}(S_b = 0,t)\big)^2\Big] \\ &+ \frac{\mathbb{P}_t(\langle 0,2,0\rangle)}{\mathbb{P}_t(S_b = 0)} \Big[\mathbb{VAR}[C_r \mid \langle 0,2,0\rangle] + \big(\mu_{(C_r)}(\langle 0,2,0\rangle,t) - \mu_{(C_r)}(S_b = 0,t)\big)^2\Big]\end{aligned} \tag{9.5}$$

Eq.(9.5) is now closed in terms of the conditional expectations and second-order moments calculated for the aggregated states $\langle 2,0,0\rangle$, $\langle 1,1,0\rangle$ and $\langle 2,0,0\rangle$. By following a similar process for the operational mode associated with $S_b = 1$, we have:

$$\begin{aligned}
&\mathbb{VAR}\,[C_r \,|\, S_b = 1\,] \;\; = \\
&+ \;\; \frac{\mathbb{P}_t(\langle 1,0,1\rangle)}{\mathbb{P}_t(S_b = 1)} \left[\mathbb{VAR}\,[C_r \,|\, \langle 1,0,1\rangle\,] + \big(\mu_{(C_r)}(\langle 1,0,1\rangle, t) - \mu_{(C_r)}(S_b = 1, t)\big)^2 \right] \\
&+ \;\; \frac{\mathbb{P}_t(\langle 0,1,1\rangle)}{\mathbb{P}_t(S_b = 1)} \left[\mathbb{VAR}\,[C_r \,|\, \langle 0,1,1\rangle\,] + \big(\mu_{(C_r)}(\langle 0,1,1\rangle, t) - \mu_{(C_r)}(S_b = 1, t)\big)^2 \right]
\end{aligned} \tag{9.6}$$

Finally, the mode of operation related to $S_b = 2$ is associated with a single state $\langle 0,0,2\rangle$. Therefore, for this mode of operation we have: $\mathbb{VAR}\,[C_r \,|\, S_b = 2\,] = \mathbb{VAR}\,[C_r \,|\, \langle 0,0,2\rangle\,]$.

Using the solution over the aggregated state space and the above equations, the evolution of $\mathbb{VAR}_t\,[C_r \,|\, S_b = 0\,]$, $\mathbb{VAR}_t\,[C_r \,|\, S_b = 1\,]$ and $\mathbb{VAR}_t\,[C_r \,|\, S_b = 2\,]$ is derived. This is reported in Fig. 9.5A. Furthermore, we also derive the *standard deviations* of C_r for each mode of operation. Recall that for a random variable X, its standard deviation $\sigma(X)$ is equal to the square root of $Var[X]$. Our standard deviations are denoted by $\sigma_t(C_r \,|\, S_b = 0)$, $\sigma_t(C_r \,|\, S_b = 1)$ and $\sigma_t(C_r \,|\, S_b = 2)$, and their evolution is shown in Fig. 9.5B.

For this model, we have the knowledge that for $S_b = i$, $i \in \{0,1,2\}$ the conditional distributions $\mathbb{P}_t(C_r \mid S_b = i)$ are uni-modal and close to being normally distributed. By exploiting this knowledge and using the calculated conditional moments, we are able to build 95% confidence bounds of the conditional distributions.

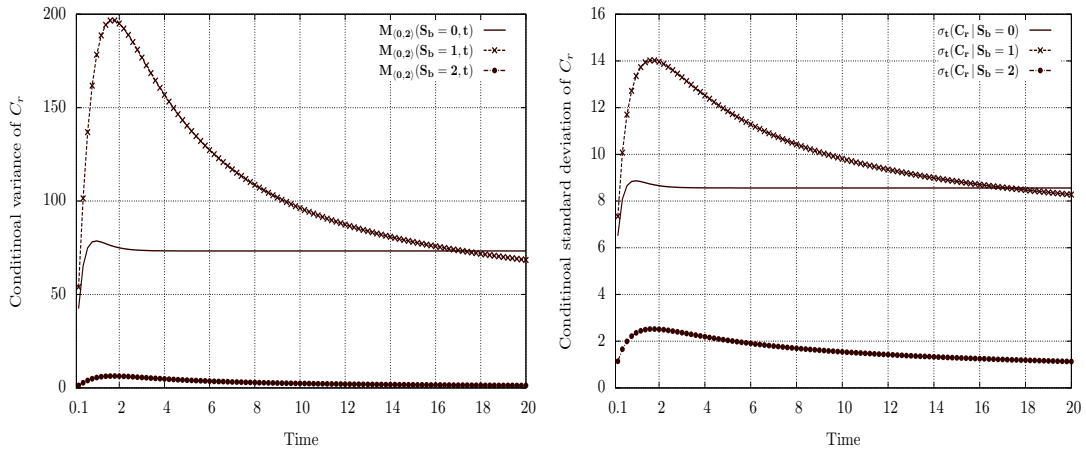
For any normally distributed random variable X with the mean $\mu(X)$ and the standard deviation $\sigma(X)$, we have:

$$\mathbb{P}(\;\mu(X) - 2\sigma(X) \leq X \leq \mu(X) + 2\sigma(X)\;) \approx 95\% \tag{9.7}$$

Applying this rule, we combine $\mu_{(C_r)}(S_b = i, t)$ and $\sigma_t(C_r \,|\, S_b = i)$ to build confidence bounds for the distributions $\mathbb{P}_t(C_r \mid S_b = i)$, since:

$$\begin{aligned}
\forall\, i \in \{0,1,2\} \;:\; \mathbb{P}_t(\; \Big[\mu_{(C_r)}(S_b = i, t) + 2\sigma_t(C_r \,|\, S_b = i)\Big] \;\leq\; C_r \;\wedge \\
C_r \;\leq\; \Big[\mu_{(C_r)}(S_b = i, t) + 2\sigma_t(C_r \,|\, S_b = 0)\Big] \;\mid\; S_b = 0 \;) \approx 95\%
\end{aligned}$$

The confidence bounds related to the mode of operation $S_b = 0$ are shown in Fig. 9.6. The transient tube around $\mu_{(C_r)}(S_b = 0, t)$ shows the values in the domain



(A) Evolution of the conditional variance of C_r given each of the modes of operations associated with $S_b = 0$, $S_b = 1$ and $S_b = 2$.

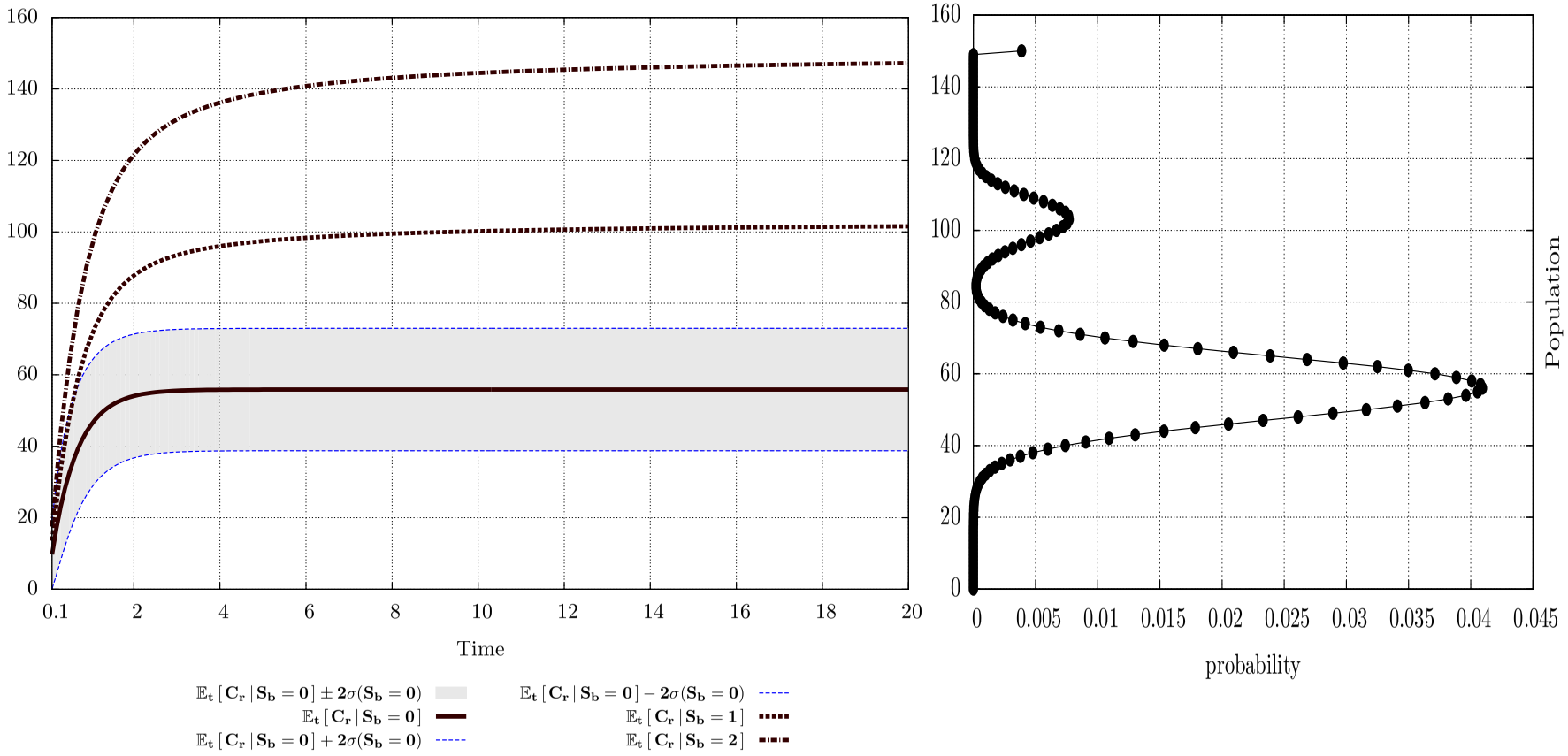
(B) Evolution of the conditional standard deviation of C_r given each of the modes of operations associated with $S_b = 0$, $S_b = 1$ and $S_b = 2$.

Figure 9.5: The transient evolution of the conditional variance of C_r and the standard deviations calculated using the second-order moments related to the aggregated states. These moments are derived by the analysis of the complete state space using PRISM.

of C_r that bear 95% of the probability mass when both servers are working $(S_b = 0)$. In this figure, we are also comparing the confidence bound at $t = 20$ (when the distribution stabilises into equilibrium) with the distribution of C_r for the same t. As illustrated, the 95% bounds are particularly useful for studying the variability of C_r in this mode of operation.

The same analysis was done for the operational mode $S_b = 1$ and $S_b = 2$. The results are respectively shown in Fig. 9.7 and Fig. 9.8. The tubes here show the most likely observations when the system is working under less optimal (and somewhat rare) modes of having server failure. These are particularly useful for tasks such as capacity planning, where we provision the system for optimal, sub-optimal and critical operational modes.

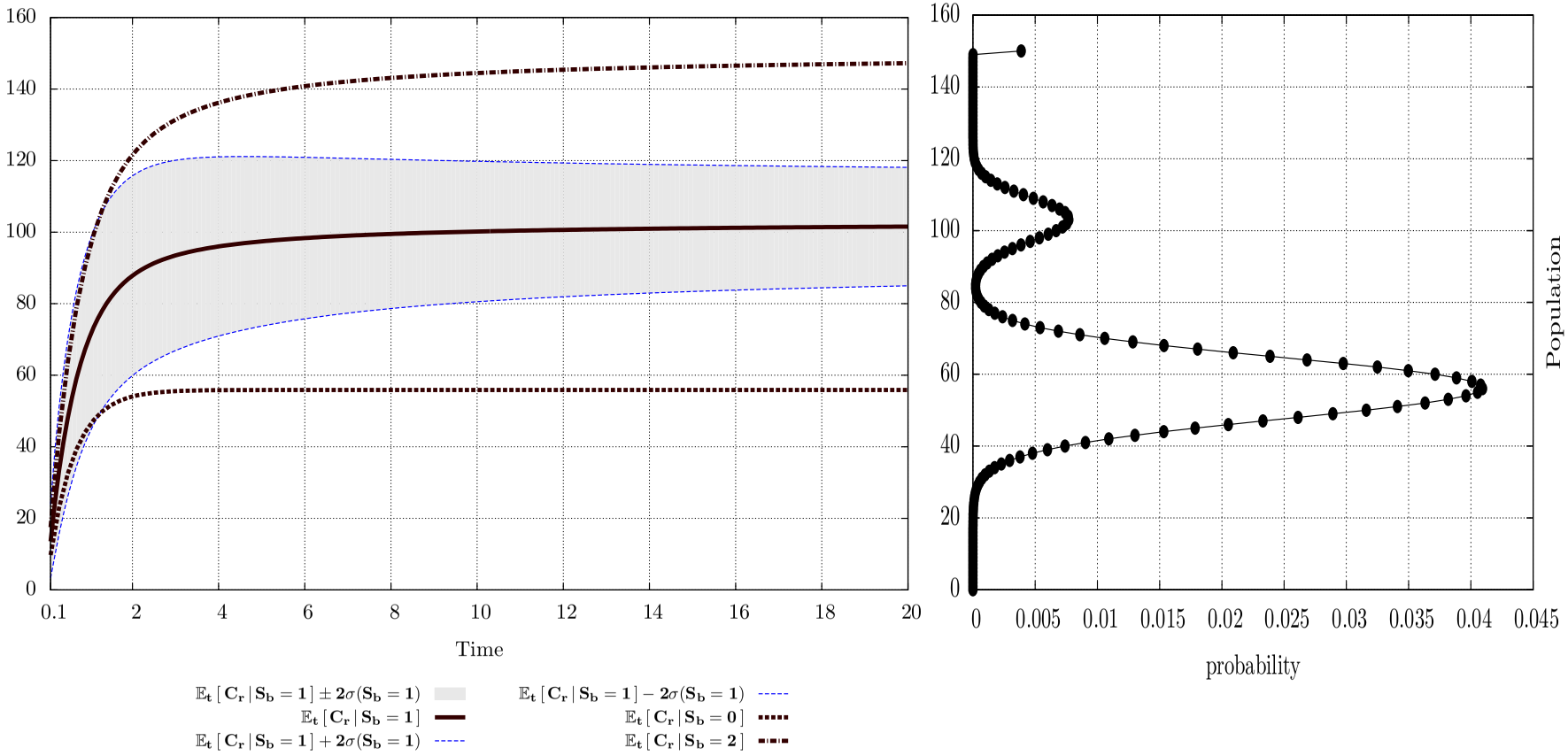
Having illustrated the usefulness of the second-order moments, in the next section we briefly comment on the importance of the derivation of the MCM equation which allows us to efficiently derive those moments.



(A) The transient evolution of conditional expectations of C_r given $S_b = 0$ and the related confidence boundary.

(B) The distribution of C_r at $t = 20$ obtained using PRISM.

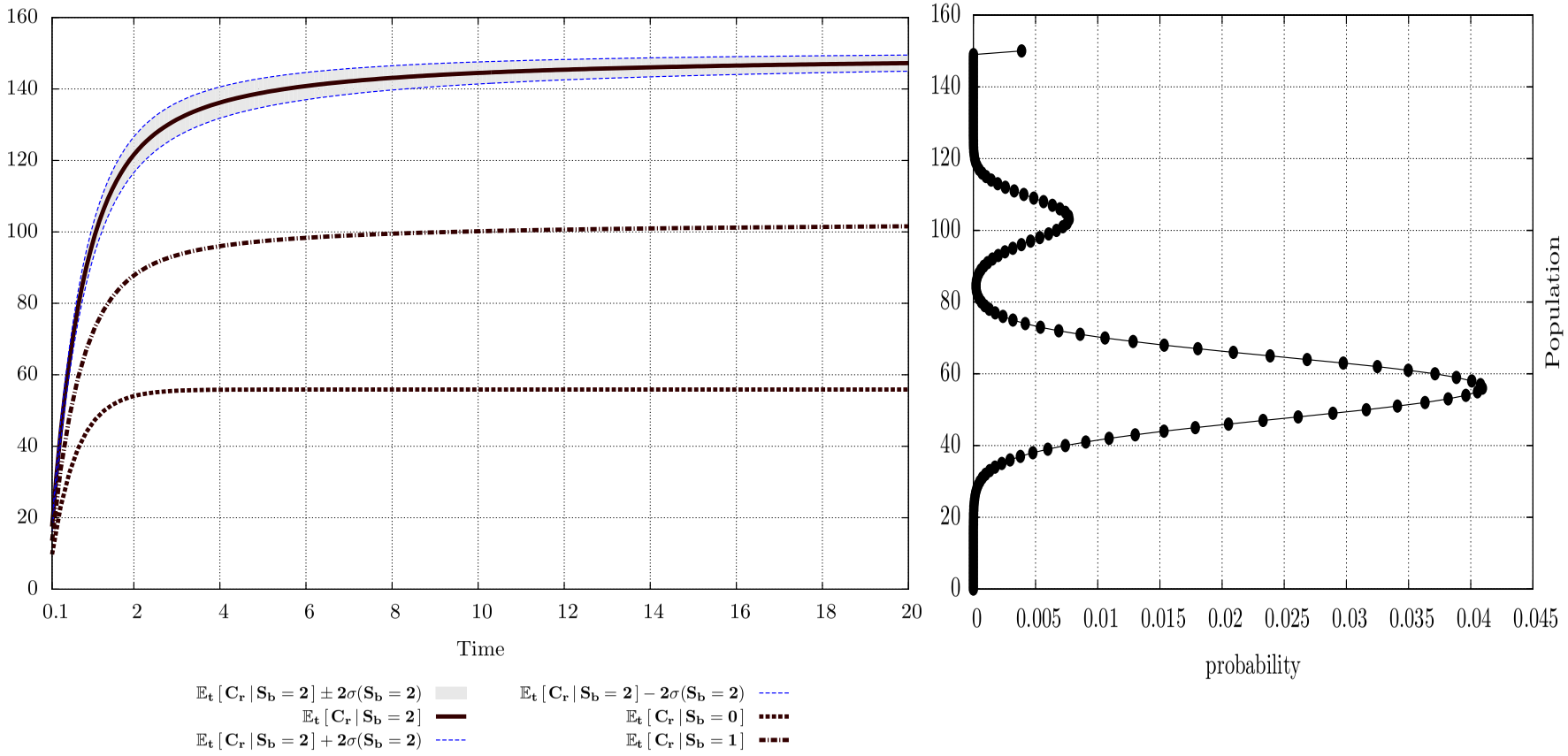
Figure 9.6: Comparison between the 95% confidence boundary derived from the conditional moments and the same measure derived from the steady state probability distribution. Here, we are focusing on the operational mode associated with $S_b = 0$.



(A) The transient evolution of conditional expectations of C_r given $S_b = 1$ and the related confidence boundary.

(B) The distribution of C_r at $t = 20$ obtained using PRISM.

Figure 9.7: Comparison between the 95% confidence boundary derived from the conditional moments and the same measure derived from the steady state probability distribution. Here, we are focusing on the operational mode associated with $S_b = 1$.



(A) The transient evolution of conditional expectations of C_r given $S_b = 2$ and the related confidence boundary.

(B) The distribution of C_r at $t = 20$ obtained using PRISM.

Figure 9.8: Comparison between the 95% confidence boundary derived from the conditional moments and the same measure derived from the steady state probability distribution. Here, we are focusing on the operational mode associated with $S_b = 2$.

9.7 Accuracy and Efficiency of the MCM approximation

The main contribution of the higher-order moments is that they extend the analysis of conditional expectations and provide the means to build a richer representation of the stochastic behaviour of the large groups. However, achieving the advantages through the analysis of complete state space (as done in the example) has a major drawback. In spite of producing accurate solutions, this type of analysis is not scalable with respect to the model's populations and for any large-scale model, the analysis becomes either computationally expensive or even infeasible. The analysis of the higher-order moments done via this path is simply not scalable.

By observing the complexity of dealing with the complete state space, we can deduce the significance of the MCM approach and in particular the importance of the derivation of Eq.(8.34). Using the MCM path, the higher-order moments are derived by solving a system of DAEs. Whilst still sensitive to the populations, the cost of solving a large-scale model's underlying system of DAEs is orders of magnitude lower compared to the analysis of its complete state space. Moreover, we expect our approximate solution to exhibit a high degree of accuracy. The accuracy of our approximations was explicitly shown for the case of the conditional expectations and we anticipate that this feature can be extended to the higher-order moments, as in the latter we are applying the same approximation steps (boundary state approximation, Taylor expansion, swapping the position of the *min* and expectation operators). Due to these features, i.e. higher efficiency and acceptable accuracy, we conclude that the MCM approach is the scalable path for the derivation of highly useful higher-order conditional moments.

We acknowledge that the rigorous validation of the MCM approach requires deriving the MCM solution and comparing against the exact solution coming from the analysis of the complete state space. However, such an experimentation cannot be readily done since the software tool for the MCM approach is still under development. Nevertheless, given our past experience with respect to validity of the conditional expectation equation (Eq.(6.36)) we can propose the following conjecture, concerning the validity of Eq.(8.34), which forms the core of the MCM approach.

Conjecture 9.7.1. *The system of DAEs constructed based on Eq.(8.34) for the analysis of higher-order moments can approximately calculate such moments with a high degree of accuracy. Moreover, using the MCM approach these moments are derived significantly more efficiently than through the analysis of the complete state space.*

Checking the validity of this conjecture will be done as a future task.

Chapter 10

Application of the Method of Conditional Moments to a Two-Tier Wireless Network

10.1 Introduction

In this chapter, we consider a case study where the method of conditional moments is applied to the performance evaluation of a complex model. The system we consider is a wireless network based on the two-tier architecture. The case study chapter has two goals. First, we describe some of the important aspects of the dynamics of networks based on this type of architecture and illustrate how they are captured in PEPA models. Second, we exhibit the usefulness of our methods when such models need to be efficiently and faithfully analysed.

In Sec 10.2, we present an overview of the two-tier architecture and the main benefits of adopting the architecture. In Sec. 10.3 we show our PEPA model and describe the performance questions that our analysis is focusing on. In Sec. 10.4, we describe how the model is analysed using our analysis techniques. Sec. 10.5 summarises the chapter.

We are planning to publish the content of this chapter as a case study; a conference paper is in preparation.

10.2 An Overview of Two-Tier Wireless Networks

First, we review the concept of two-tier network architecture and briefly discuss some of the important characteristics of wireless networks based on this architecture. We discuss the benefits of applying the architecture and explain the performance evaluation requirements that we deal with when analysing such networks.

10.2.1 Two-Tier Architecture For Wireless Networks

A *single*-tier wireless network is regarded as a network which consists of one base station and a population of wireless users. The base station is usually deployed at the centre and is responsible for serving all users located within its coverage. Compared to the single-tier architecture, in the two-tier architecture two layers of base stations are deployed [6]. A simple schematic of this type of network is shown in Fig. 10.1. As before, there is a main station usually placed at the centre. Here, this is referred to as the *macro-cell*. In addition, a number of less powerful base stations are also deployed, which are referred to as *femto-cells*. The femto-cells are light-weight in their capacity, and provide a strengthened signal locally and only within short radio ranges. The base station that serves a user depends on the user's location. If she is in the vicinity of a local femto-cell, then preferably she is served by this local station, enjoying the strengthened signal it provides. On the other hand, when no femto-cell can be reached, the user is served by the central macro-cell. The connectivity is brought to the femto-cells by their high speed wired links to the macro-cell.

There has been growing interest in using the two-tier architecture for wireless wide area networks (WWAN). Some examples include adopting the architecture for indoor areas or large urban sites [14, 30]. The macro-cell is usually assigned the task of covering locations towards the centre and the femto-cells are positioned at indoor locations or the edges of the network, where the wireless signals are more faded. Accordingly, users are categorised based on their locations, those within a defined central perimeter are referred to as *near users* whilst those located at the edge are referred to as *far users*.

Note here that in our presentation, the notion of *a user* is different from the one known in common parlance. Depending on our level of abstraction, a user refers to any entity that exploits the capacities of the network (for instance, for data transmission) as part of its evolution. In this sense, the data packets waiting for transmission on a data

link and within a femto-cell can be regarded as network users. We will elaborate more on this when describing our model.

10.2.2 Benefits and Advantages

Compared to the single-tier approach, the two-tier architecture potentially achieves significant improvements in terms of reduced wireless interference and improved energy efficiency [30]. Let us briefly elaborate on these improvements.

10.2.2.1 Reduced Interference

In wireless networking, the *interference* of connections refers to situations where the transmission from one node on a certain frequency coincides with one or more other nodes transmitting on the same frequency. When two signals interfere, they get convoluted and this results in the recipient not being able to distinguish between them. Therefore, the packets that the signals convey are lost. From this perspective, the single-tier architecture is not suitable for WWANs [6]. Here, the central base station needs to be panoptic, and all connections need to be established within a very wide single cell. When the demand grows (for instance, through a large population of users) the rate of coincidence increases and the network's performance degrades as the high interference leads to excessive data loss and consequently, frequent message re-transmission.

The two-tier architecture is one effective solution in reducing the interference levels. Using this architecture, each femto-cell covers a small area in its close vicinity and the active channels within one femto-cell have limited impact on connections that are active in other femto-cells. This lowers the probability of *co-tier interference*, i.e. connections present in different femto-cells affecting each other. Moreover, the macro-cell covers a reduced central area too. Therefore, the connections between the macro-cell and near users become somewhat isolated from the connections within the femto-cells. As a result, the *cross-tier interference* is decreased. The reduced levels of the two types of interference lowers the odds of signals being convoluted (packets getting lost) and consequently both near and far users experience better transmission rates.

10.2.2.2 Energy Efficiency

In wireless networking, the distances between entities play a key role in the amount of energy spent for maintaining their connections. The wireless signals tend to *fade*, that is, the power of the signals decreases as the distance increases. To cope with this distance, when two entities communicate the signals they exchange must be adequately strengthened, such that the concomitant fading is overcome. Although it is not an issue for connections between entities that are close, the fading has important implications for long distance connections (LDC). For LDCs, the signals need extra strength. To achieve this, the entities need to enter their high power mode, which makes them consume more energy. Therefore, the network with frequent LDCs will have worse energy efficiency than the one where such connections occur less often.

In a single-tier network, the LDCs are frequently established between the central station and far users, and the macro-cell frequently enters its high power mode. This degrades the efficiency of the network infrastructure. Far users are also severely affected, as they are usually equipped with limited energy sources (batteries) which are quickly depleted by the energy demanding LDCs to the macro-cell.

Using the two-tier architecture, the likelihood of having LDCs is reduced. The users are divided into separate regions and the network's load is divided between the macro-cells and femto-cells. Although the macro-cell can potentially serve far users, they are primarily served by their local femto-cells using local connections. Here, the inevitable LDCs of the single-tier architecture are to a high degree avoided. Consequently, the network offers a better energy footprint, both for the infrastructure and the far users.

10.2.3 Modes of Operation

In an *ideal* two-tier network the load is perfectly divided between the two layers; each far user is served by its local femto-cell and each near user is served by the central macro-cell. However, such a division is not fully adhered to in practice. In realistic implementations, it is possible to observe a state where a far user is being served by the macro-cell by a LDC. This happens when the far user is rejected by its local femto-cell or is simply not covered by any. The rejection may happen due to a number of reasons, some of which are now described. The user might be located at the edge of

the femto-cell's coverage and might receive better service by the macro-cell. Or the femto-cell might be already fully loaded and not able to accept new requests. Also, the load balancing protocol (cell association protocol) of the femto-cell might transfer the service to the macro-cell, because of the decisions it makes based on the history of, and the current load conditions in, the network.

When a far user is not served by its femto-cell, she attempts to initiate a LDC to the macro-cell. The macro-cell has a number of channels dedicated to serving far users. If such a channel is available and free, the far user is admitted. If all channels are taken, then the far user is rejected. When admitted, a LDC is established and the macro-cell enters the *power boost* mode. When serving the far user is completed, the channel which was allocated is released. When the macro-cell has no active LDC, it returns to the *normal* power mode.

The network exhibits some important characteristics during the time that the macro-cell is serving at least one far user. First, the macro-cell enters the high power mode and consumes more energy. Second, since the macro-cell's signals are strengthened the data loss rate decreases *for the near users* and they enjoy *better communication* with the macro-cell. Third, the highly powered signals exchanged with the macro-cell increase the cross-tier interference experienced within the femto-cells and as a result, the femto-cell local users (far users) suffer from lowered transmission rates. These important characteristics suggest that the network has two distinct modes of operation [43]. The first is the *normal* mode (corresponding to the normal power consumption), which refers to the periods of time when all far users are served by their local femto-cells and no LDC exists. The second is the *power boost* mode, which refers to the periods of time when there is at least one active LDC within the macro-cell.

10.2.4 Requirements of the Performance Evaluation and the Model

The benefits of the two-tier architecture are achieved if the network is well configured. When designing the network, particular attention is paid to the possibility of the network experiencing the aforementioned modes of operation. Since the power boost mode has less desirable performance, the network is designed in such a way that the network is less often in this mode than the normal mode.

The possibility of having two modes of operation defines important requirements for the performance evaluation of two-tier networks. The model, being an abstract

reflection of the reality, needs to be capable of capturing the distinct operational modes. On the other hand, when analysing the model it is imperative to study the effect of mode alternations, and analyse the dynamics of the network in both modes. The normal mode represents the *usual* and *mostly observed* mode of operation and the analysis shows the behaviour when the system is working under the desired circumstances. By analysing the power boost mode, we aim to study the system's behaviour when performing in the *sub-optimal* and *less likely* states. Faithful and robust performance analysis requires the analysis of both modes.

Given a two-tier network, two key fundamental performance questions are: a) what is the probability of being in each of the different modes of operation and b) what are the consequences of mode alternations, particularly on users? In this chapter, we propose a model which allows us to answer these questions. In the next section, we introduce the model and explain the key performance questions in more detail.

10.3 Modelling

In this section, we describe a concrete scenario for a two-tier wireless network. First, we review the network's spatial properties. Second, we describe its key entities and how they are modelled in PEPA. Finally, we describe the performance questions that we focus on when analysing the model.

10.3.1 Spatial Distribution

Fig. 10.1 is the sketch of a two-tier wireless network [43]. We assume that the macro-cell is located at the center and it covers the area within the radius R_m. There are N_f femto-cells, indexed by $i: 1 \leq i \leq N_f$ which are located at the edges of the network. Each femto-cell covers a radius $R_f << R_m$. We assume that these cells are within an annulus at the interior of the area marked by R_m.

The macro-cell's main area of concern is the reduced area with radius $R_{m'} < R_m$, $R_{m'} = R_m - 2R_f$. The users within this area are called the *near users*. On the other hand, those with distances r : $R_{m'} < r < R_m$ are regarded as *far users*. These are preferably served by their femto-cell. In our modelling we have the simplifying assumption that the femto-cells are homogeneously distributed and their coverages are

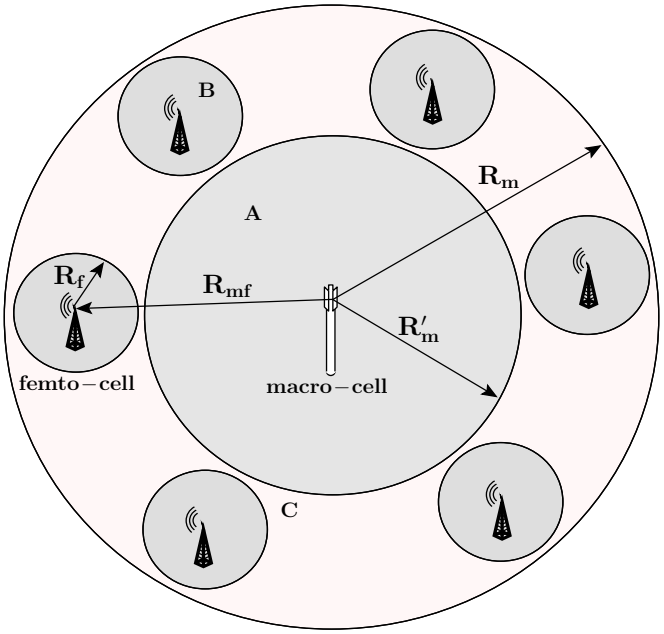


Figure 10.1: Structure of a two-tier wireless network.

non-overlapping. This means that we can abstract from the co-tier interference. Nevertheless, we allow the macro-cell's coverage to be overlaid by the femto-cells' and thereby, the impact of the cross-tier interference is still captured.

10.3.1.1 Patterns of User Arrival

The arrival of a user to a cell (base station) is categorised based on the location of the user. The categories are:

1. Arrival of a user in the vicinity of the macro-cell. The user is within the radius $R_{m'}$ (marked by A in Fig 10.1) and is served by the macro-cell.

2. Arrival of a user *in close vicinity* of a femto-cell. Such a user (marked by B in Fig 10.1) is referred to as a *strongly coupled* user (SCU). Due to the possibility of enjoying the femto-cell's strengthened signals, the user receives service from its local femto-cell.

3. Arrival of a user located within common areas. The user is logically neither close to a macro-cell nor to a particular femto-cell (marked by C in Fig 10.1). It can be in locations where the coverages of the macro and femto cells overlap (edges of the radius R_M) or at far distances not strongly covered by any femto-cell. Since no particular base station has the dominance, the user can potentially be served

by a base station of either type. We assume that each femto-cell has a number of users weakly connected to it (weakly connected users (WCU)).

The main reason for introducing the third class of users (in addition to those being close to either the macro-cell or a femto-cell) is the fact that in wireless networking there is no way to specify a clear-cut border for the coverage of the base stations. In contrast to *wired* networking where the link that a user communicates over is topologically deterministic, the wireless user is usually under the coverage of potentially many base stations and the choice of which link to use is more dynamic. Normally, the user chooses the station which offers the best transmission rate. In our two-tier wireless network, the users for whom neither the macro-cell nor a femto-cell provide the *dominant* means of communication are considered to be in this class. These users are the main factor giving rise to LDC and therefore, it is important to include them as a separate component in the model.

Having stated the categories of the users, now we explain how the interaction between the users and base stations can be captured in our PEPA model.

Before presenting our model, it is worth mentioning the work in [41] that uses PEPA to evaluate a two-tier cellular network. In that work, the main feature under consideration is user *mobility*, and how the network's so-called *handover protocol* maintains quality of service to a mobile user that switches its current location. Another important feature is that the focus is on the behaviour of individual users and the authors do not consider state space reduction techniques based on counting abstraction; to avoid state space explosion, notions of process equivalence developed in [61] are heavily used. Our modelling has a different direction. We assume that users are fixed in their location and *some of them* have the possibility of being served by either a macro-cell or femto-cell. Furthermore, we use counting abstraction as the primary means of compact state space construction and investigate how the methods based on differential equations can tackle our model's large state space.

10.3.2 PEPA Model of the Two-Tier Wireless Network

In this section, we present the different entities that our two-tier wireless network consists of and the way PEPA is used to model them. In our modelling, we extensively use compositionally and the cooperation operator. This helps us to build our complex model from the bottom up; first we focus on the behaviour of the entities in isolation

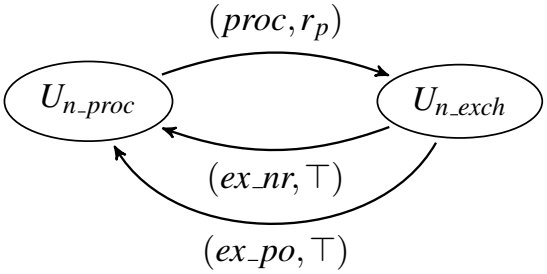


Figure 10.2: The behaviour of a user close to the macro-cell.

and then combine them to build the model of the whole. With our presentation, we show that building the model of a system as complex as a two-tier network, which inherently features a multitude of interacting components, is indeed a hard task, and one which is helped considerably by using a compositional language such as PEPA. We start by describing the behaviour of the users.

10.3.2.1 Near Users Close to the Macro-cell

The first category of users consists of near users. The behaviour of a user of this kind is:

$$\begin{aligned} U_{n\_proc} &\stackrel{def}{=} (proc, r_p).U_{n\_exch} \\ U_{n\_exch} &\stackrel{def}{=} (ex\_nr, \top).U_{n\_proc} + (ex\_po, \top).U_{n\_proc} \end{aligned} \tag{10.1}$$

as illustrated in Fig. 10.2. The user starts in the state U_{n_proc}, where it performs some independent and internal processing. The state represents the behavioural phase when the user does not require any communication with the network. When the internal processing finishes, the user enters the state U_{n_exch} where it exchanges some data with the macro-cell. This scenario, for instance, captures the behaviour of a node in a two-tier wireless sensor network, where the nodes first gather some data independently and then upload it to a central sink. It can also represent a two-tier file storage service where the users issue a periodic fetch-me-a-file command while mostly working on their offline jobs.

The macro-cell is equipped with a number of channels to serve near users. When a user is ready for the data exchange and there exists a free channel, the user can immediately start the communication. When there is no free channel, the user waits for the next available one. The rate offered for the data exchange depends on the net-

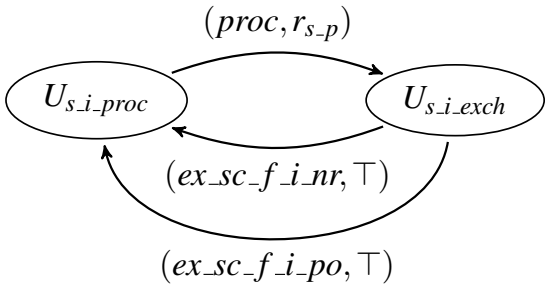


Figure 10.3: The behaviour of a user close to a femto-cell indexed by i: $1 \leq i \leq N_f$.

work's mode of operation (see Sec. 10.2.3); the rate is higher in the power boost mode compared to the normal mode. Given the distinct rates, we model the data exchange with two different action type; *ex_nr* and *ex_po*. The first is enabled when the system is in the normal mode and the second when operating in the power boost mode. At any given time, only one of these actions is enabled, on which the user passively synchronises. The nature of this synchronisation will be described later.

Given the wide area that the macro-cell covers, we assume that the near users form a large population. This assumption is in line with our performance evaluation goals, particularly that we aim to study the system's behaviour when performing under heavy load. The behaviour of the macro-cell channels is described when presenting the PEPA model of the macro-cell.

10.3.2.2 Far Users Strongly Coupled to Femto-cells

The second user category consists of SCUs (strongly coupled users with respect to femto-cells). The behaviour associated with this type of user is:

$$\begin{aligned} U_{s\_i\_proc} &\stackrel{def}{=} (proc, r_{s\_p}).U_{s\_i\_exch} \\ U_{s\_i\_exch} &\stackrel{def}{=} (ex\_sc\_f\_i\_nr, \top).U_{s\_i\_proc} + (ex\_sc\_f\_i\_po, \top).U_{s\_i\_proc} \end{aligned} \quad (10.2)$$

and is shown in Fig. 10.3. The only difference is that instead of being served by the macro-cell, these users are served by their local femto-cells.

Each femto-cell is equipped with a number of channels for serving its local SCUs. If a user arrives and there is no free channel, then the user waits, with the anticipation that a channel will be available soon, and there would then be no need to establish a LDC to the macro-cell. The user communicates with the femto-cell by undertaking either *ex_sc_f_i_nr* or *ex_sc_f_i_po*, which respectively correspond to the data exchanges

at normal and power boost modes. The femto-cells are usually placed at locations where there is a dense cluster of user demand. Therefore, we assume that SCUs form relatively large populations. However, note that due to the femto-cell's limited capacity ($R_f << R_m$) SCU populations are still smaller than the near users around the macro-cell. This allows us to study the dynamics of a femto-cell (from the perspective of both the femto-cell and its SCUs) when there is heavy load on the femto-cell.

10.3.2.3 Far Users Weakly Coupled to Femto-cells

The third user category consists of WCUs (weakly connected users). Focusing on the femto-cell indexed as i, the behaviour of any WCU around i is shown in Fig. 10.4. The PEPA process capturing this behaviour is:

$$\begin{aligned}
U_{w_i_proc} &\stackrel{def}{=} (proc, r_{w-p}).U_{w_req_f_i} \\
U_{w_req_f_i} &\stackrel{def}{=} (req_f_i, r_{ctrl}).U_{w_wait_f_i} \\
U_{w_wait_f_i} &\stackrel{def}{=} (f_i_acc, r_{ctrl}).U_{w_exch_f_i} + (f_i_rej, r_{ctrl}).U_{w_i_req_mcr} \\
U_{w_exch_f_i} &\stackrel{def}{=} (ex_wc_f_i_po, \top).U_{w_proc} + (ex_wc_f_i_nr, \top).U_{w_i_proc} \\
U_{w_i_req_mcr} &\stackrel{def}{=} (req_mcr, r_{ctrl}).U_{w_i_wait_mcr} \\
U_{w_i_wait_mcr} &\stackrel{def}{=} (mcr_acc, r_{ctrl}).U_{w_i_exch_mcr} + (mcr_rej, r_{ctrl}).U_{w_i_proc} \\
U_{w_i_exch_mcr} &\stackrel{def}{=} (ex_wc_mcr, \top).U_{w_proc}
\end{aligned} \tag{10.3}$$

The user starts from the state $U_{w\_i\_proc}$ where it is undertaking an independent processing step. Having done that, it moves to state $U_{w\_req\_f\_i}$ where it sends a request for a femto-cell WCU channel, and then enters the state $U_{w\_wait\_f\_i}$ where it waits for the response. For a femto-cell, the service to WCUs has lower transmission rate (longer transmission time) and is more energy demanding than the SCUs. Therefore, it is equipped with only a limited number of WCU channels. When a WCU requests and there is free WCU channel, the femto-cell accepts the user and sends the f_i_acc message. This makes the user move to the state $U_{w\_exch\_f\_i}$ where it starts its data exchange. Conversely, if all channels are busy then the user is rejected (by receiving the f_i_rej message) and enters the state $U_{w\_i\_req\_mcr}$.

Having been rejected from the femto-cell, the user now sends a request to the macro-cell for a LDC and enters the state $U_{w\_i\_wait\_mcr}$ where it waits for the response. Given the detrimental effects of LDCs, the macro-cell offers only a limited number of channels for such connections. If such a channel is free, the far user is admitted

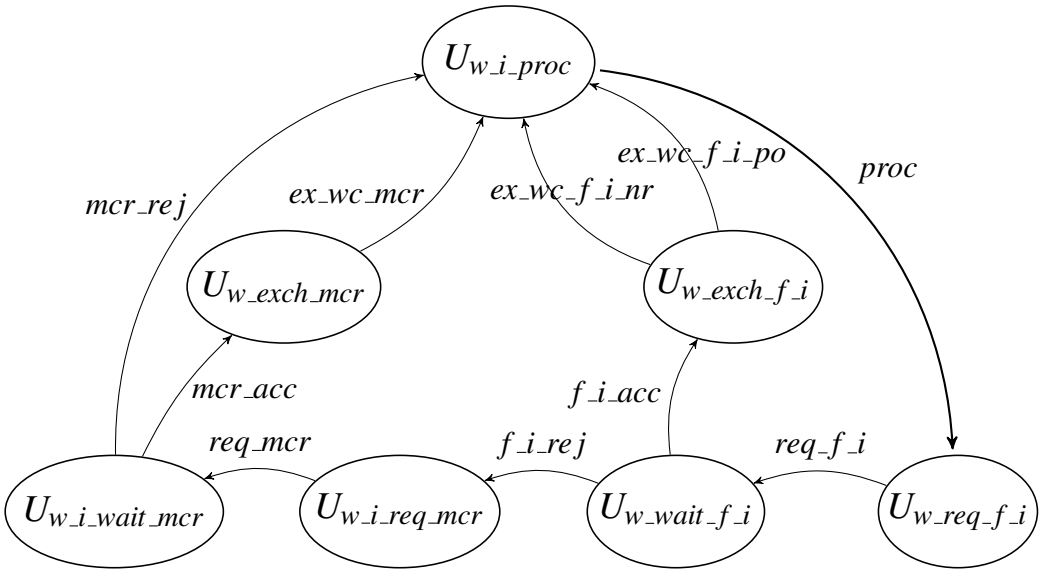


Figure 10.4: The behaviour of a user weakly coupled to a femto-cell i: $1 \leq i \leq N_f$.

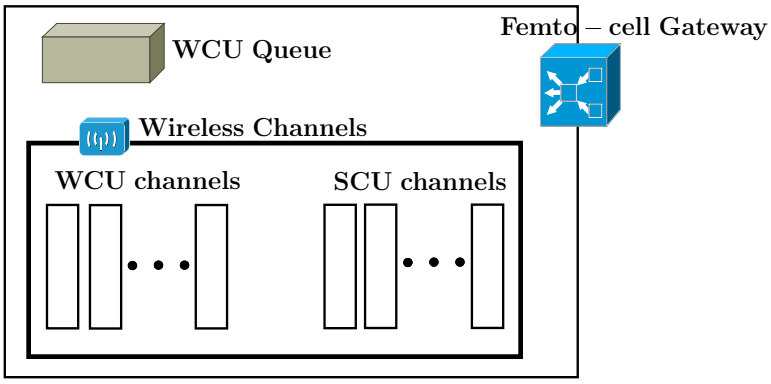


Figure 10.5: A femto-cell and the internal components defined for capturing its behviour.

and sent the *mcr_acc* message. This enables the user to move to the state $U_{w\_i\_exch\_mcr}$ where it starts the data exchange with the macro-cell. The transmission rate for this type of connection is significantly lower compared to the short distance connections and it takes a relatively long time for the user to be served. If the macro-cell has no free LDC channel, then the user is rejected again (this time it receives the *mcr_rej* message). We assume that a WCU who is rejected again returns to its initial state.

10.3.2.4 Femto-cells

Now we focus on a femto-cell and how it is modelled. In order to faithfully capture the interactions with SCUs and WCUs, we define three components which collectively exhibit the behaviour expected from a femto-cell. These components are:

1. Femto-cell gateway, referred to as FG.
2. A queue for managing the WCUs, referred to as $WCUQ$.
3. Femto-cell communication channels dedicated to SCUs, referred to as $FSCCH$.

These components are shown in Fig. 10.5, which shows a very simplified structure for a femto-cell. For simplicity, we assume that in our network all femto-cells have the same specification.

In the following, we describe the behaviour of each of these components separately using a PEPA process, and then combine these to construct the model of a femto-cell. In our presentation, we describe the behaviour of a femto-cell indexed by i.

Femto-cell Gateway

As stated, each femto-cell can serve a few WCUs at any given time. The gateway manages the requests for WCU connections. The SCU connections are happening at a fast rate and are directly handled by the SCU channels. The behaviour of the gateway is:

$$\begin{aligned} FG_{i\_idle} &\stackrel{def}{=} (req\_f\_i, r_{ctrl}).FG_{i\_check} \\ FG_{i\_check} &\stackrel{def}{=} (wcuq_{i\_ne}, r_{ctrl}).FG_{i\_acc} + (wcuq_{i\_e}, r_{ctrl}).FG_{i\_acc} + \\ &\quad (wcuq_{i\_full}, r_{ctrl}).FG_{i\_rej} \\ FG_{i\_acc} &\stackrel{def}{=} (f\_i\_acc, r_{ctrl}).FG_{i\_idle} \end{aligned}$$

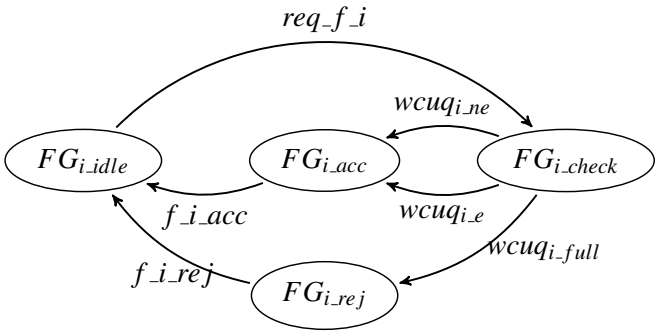


Figure 10.6: The behaviour of the gateway of the femto-cell indexed by i. The rates of all actions are r_{ctrl}.

$$FG_{i\_rej} \stackrel{def}{=} (f\_i\_rej, r_{ctrl}).FG_{i\_idle}$$

which is illustrated in Fig. 10.6. The initial state is FG_{i_idle} where the gateway is ready to receive requests from any WCU. When a request arrives, the gateway enters the state FG_{i_check} where it consults with the $WCUQ_i$ to decide whether the new request can be accepted. If $WCUQ_i$ sends the control signal $wcuq_{i\_e}$ or $wcuq_{i\_ne}$, meaning there are still some free WCU channels in the femto-cell, then the gateway enters the state FG_{i_acc} where it informs the user that its request is accepted. However, if the signal $wcuq_{i\_full}$ is received, then the femto-cell has reached its maximum capacity and the gateway enters the state FG_{i_rej} where it informs the user that its request is rejected. When the response is sent, the gateway returns to its initial state.

Weakly Coupled Users Queue

Let N_{f_w} denote the upper bound for the number of WCU connections that may simultaneously be active within a femto-cell. The component $WCUQ_i$ acts as a counter and enforces the bound N_{f_w}. Since it holds the state of active WCU connections, this component is also responsible for sending *control* signals to FG.

The behaviour of $WCUQ_i$ is:

$$\begin{aligned}
WCUQ_{i_0} &\stackrel{def}{=} (f_i_acc, r_{ctrl}).WCUQ_{i_1} + (wcuq_{i_e}, r_{ctrl}).WCUQ_{i_0};\\
&\vdots\\
WCUQ_{i_j} &\stackrel{def}{=} (f_i_acc, r_{ctrl}).WCUQ_{i_j+1} + (wcuq_{i_ne}, r_{ctrl}).WCUQ_{i_j} +\\
&\quad (ex_wc_f_i_nr, j \times r_{ex_wc_f_i_nr}).WCUQ_{i_j-1} \;+\\
&\quad (ex_wc_f_i_po, j \times r_{ex_wc_f_i_po}).WCUQ_{i_j-1};
\end{aligned}$$

$$\vdots \tag{10.4}$$

$$\begin{aligned} WCUQ_{i\_N_{f\_w}} \stackrel{def}{=} & (ex\_wc\_f\_i\_nr, N_{f\_w} \times r_{ex\_wc\_f\_i\_nr}).WCUQ_{(i\_N_{f\_w}-1)} + \\ & (ex\_wc\_f\_i\_po, N_{f\_w} \times r_{ex\_wc\_f\_i\_po}).WCUQ_{(i\_N_{f\_w}-1)} + \\ & (wcuq_{i\_full}, r_{ctrl}).WCUQ_{i\_N_{f\_w}}; \end{aligned}$$

shown in Fig. 10.7. It starts in state $WCUQ_{i\_0}$, denoting that there are currently no active WCU connections. Here, the action $wcuq_{i\_e}$ is enabled, conveying the control signal that $WCUQ_i$ is *empty*, which then helps the gateway determine that a new request for a WCU connection can be accepted. If this happens, the state changes from $WCUQ_{i\_j}$ to $WCUQ_{i\_j+1}$. Similarly, $wcuq_{i\_ne}$ (non-empty) is enabled for all $0 \leq j \leq (N_{f\_w} - 1)$ which denotes that there is at least one active WCU within the femto-cell and there is still some capacity left. When the component is in state $WCUQ_{i\_N_{f\_w}}$, then the control signal $wcuq_{i\_full}$ is sent, expressing that no more WCU connections can be accepted. This signal helps the gateway to reject the WCU requests when the full capacity of the femto-cell is utilised.

Depending on the network's mode of operation, the $WCUQ_i$ enables two different actions for serving WCUs. These are $ex\_wc\_f\_i\_nr$ and $ex\_wc\_f\_i\_po$ respectively related to the normal mode and power boost mode. The component which is aware of the mode (the existence of LDC connections) is not the femto-cell, but rather the macro-cell. Therefore, when modelling a WCU data exchange, although the rate is decided by $WCUQ_i$ the action type that is actually enabled is determined by a synchronisation with the macro-cell. The influence of the macro-cell will be explained when describing its PEPA component. When a WCU connection is finished, the $WCUQ_i$ is updated again; it moves from a current state $WCUQ_{i\_j}$ to $WCUQ_{i\_j-1}$.

Actions $wcuq_{i\_e}$, $wcuq_{i\_ne}$, $wcuq_{i\_full}$, f_i_acc take place with rate r_{ctrl}. In each state $WCUQ_{i\_j}$, the action $ex\_wc\_f\_i\_nr$ takes place at rate $j \times r_{ex\_wc\_f\_i\_nr}$. Moreover, the action $ex\_wc\_f\_i\_po$ takes place at rate $j \times r_{ex\_wc\_f\_i\_po}$. Due to the higher interference levels in power boost mode, we have: $r_{ex\_wc\_f\_i\_po} < r_{ex\_wc\_f\_i\_nr}$.

Femto-cell Channel for Strongly Connected Users

The SCU channels are responsible for serving SCUs. The behaviour of this component is:

$$FSCCH_i \stackrel{def}{=} (ex\_sc\_f\_i\_nr, r_{ex\_sc\_f\_i\_nr}).FSCCH_i + (ex\_sc\_f\_i\_po, r_{ex\_sc\_f\_i\_po}).FSCCH_i$$

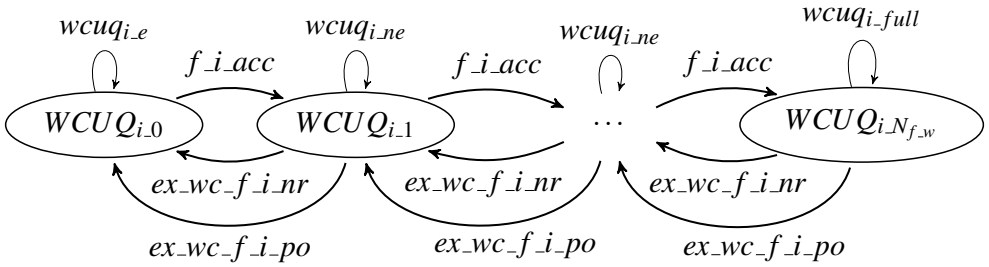


Figure 10.7: The behaviour of $WCUQ_i$ in femto-cell i.

Figure 10.8: The behaviour of a SCU channel in femto-cell i. We assume that the channel stays in a constant state and enables the action types for serving SCUs.

which is also shown in Fig. 10.8. Here, the important behavioural aspect is not the set of states a channel experiences, but rather the actions $exch\_sc\_f\_i\_nr$ and $exch\_sc\_f\_i\_po$ that it repeatedly undertakes. These actions capture the frequent data exchanges that the channels perform in dealing with the flow of SCUs into the femto-cell. We assume that when a SCU attempts to communicate, it acquires the first free channel available without negotiating with the femto-cell's gateway. In the process above: $r_{ex\_sc\_f\_i\_po} < r_{ex\_sc\_f\_i\_nr}$.

Femto-cell Composition

Using PEPA's cooperation operator, the described components are combined to specify the behaviour of a femto-cell as a whole:

$$FC\_i \stackrel{def}{=} \Big[F\_i\_Gateway \{ FG_{i\_idle}[1] \} \underset{L}{\bowtie} WQ\_i \{ WCUQ_{i\_0}[1] \} \Big] \;||\; FCH\_i \{ FSCCH_i[n_{s\_ch}] \} \quad (10.5)$$

where $F\_i\_Gateway$, WQ_i and FCH_i are respectively labels of the groups $FG_{i\_idle}[1]$, $WCUQ_{i\_0}[1]$ and $FSCCH_i[n_{s\_ch}]$; $L = \{wcuq_{i\_e}, wcuq_{i\_ne}, wcuq_{i\_full}; and f\_i\_acc\}$, n_{s_ch} is the number of SCU channels.

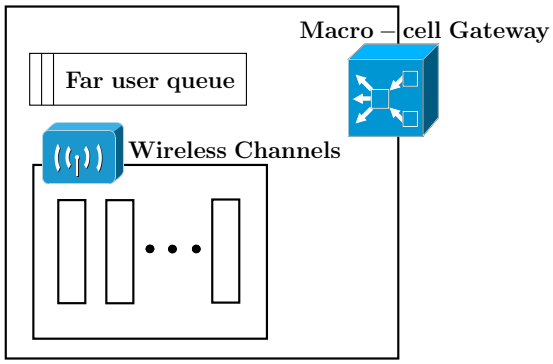


Figure 10.9: The components used for modelling a macro-cell.

10.3.2.5 Macro-cell

The macro-cell serves near users and far WCUs who have been rejected by their local femto-cells. The components that form a macro-cell are shown in Fig. 10.9 and are:

1. Macro-cell gateway, referred to as MG.
2. A queue for managing LDCs made to far WCUs. This is referred to as LDCQ.
3. Wireless channels for serving near users, referred to as NUCH.

First, we describe the PEPA process constructed for each component. Then, these are composed to express the behaviour of the macro-cell as a whole.

Macro-cell Gateway

The macro-cell distinguishes between the connections made to near users and the LDCs made to far WCUs. The gateway manages the requests for LDCs. The behaviour of this component is:

$$\begin{aligned} MG_{idle} &\stackrel{def}{=} (req\_mcr, r_{ctrl}).MG_{check} \\ MG_{check} &\stackrel{def}{=} (ldcq_{ne}, r_{ctrl}).MG_{acc} + (ldcq_{e}, r_{ctrl}).MG_{acc} + \\ &\quad (ldcq_{full}, r_{ctrl}).MG_{rej} \\ MG_{acc} &\stackrel{def}{=} (mcr\_acc, r_{ctrl}).MG_{idle} \\ MG_{rej} &\stackrel{def}{=} (mcr\_rej, r_{ctrl}).MG_{idle} \end{aligned}$$

which is also illustrated in Fig. 10.10. In the idle state MG_{idle}, the gateway is ready to receive LDC requests. When one arrives, the gateway enters the state MG_{check} where it

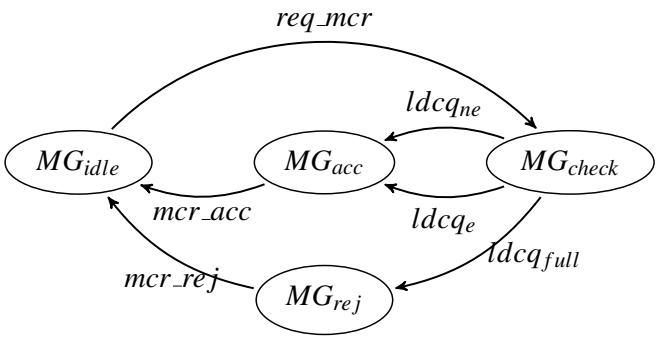


Figure 10.10: The behaviour of the macro-cell gateway. This component checks the LDCQ's control signals to see if a far WCU user asking for a LDC connection can be admitted. The rates of all actions are r_{ctrl}.

consults with LDCQ to check if the new request can be accepted. There exists a bound on the number of LDCs that the macro-cell can simultaneously hold. If the control messages $ldcq_e$ or $ldcq_{ne}$ arrive, then the user is admitted and a LDC is established. As a result, the macro-cell enters the high power mode. Conversely, if the signal $ldcq_{full}$ arrives, then the pending WCU is rejected. When the response is sent, the gateway returns to the idle state.

Long Distance Connection Queue

Let N_{LDC} denote the upper bound on the number of LDCs that a macro-cell can simultaneously maintain. The component $LDCQ$ holds the state of active LDCs and enforces this upper bound. The behaviour of this component is:

$$
\begin{aligned}
LDCQ_0 &\stackrel{def}{=} (ldcq_e, r_{ctrl}).LDCQ_0 + (ex\_nr, \top).LDCQ_0 \; + \\
&\quad (ex\_sc\_f\_i\_nr, \top).LDCQ_0 + (ex\_wc\_f\_i\_nr, \top).LDCQ_0 \; + \\
&\quad (acc\_mcr, r_{ctrl}).LDCQ_1; \\
&\vdots \\
LDCQ_j &\stackrel{def}{=} (ldcq_{ne}, r_{ctrl}).LDCQ_j + (ex\_po, \top).LDCQ_j + \\
&\quad (ex\_sc\_f\_i\_po, \top).LDCQ_j + (ex\_wc\_f\_i\_po, \top).LDCQ_j + \\
&\quad (ex\_wc\_mcr, j \times r_{ex\_wc\_mcr}).LDCQ_{(j-1)} + (acc\_mcr, r_{ctrl}).LDCQ_{j+1}; \\
LDCQ_{N_{LDC}} &\stackrel{def}{=} (ldcq_{full}, r_{ctrl}).LDCQ_{N_{LDC}} + (ex\_po, \top).LDCQ_{N_{LDC}} \; + \\
&\quad (ex\_sc\_f\_i\_po, \top).LDCQ_{N_{LDC}} + (ex\_wc\_f\_i\_po, \top).LDCQ_{N_{LDC}} + \\
&\quad (ex\_wc\_mcr, N_{LDC} \times r_{ex\_wc\_mcr}).LDCQ_{(N_{LDC}-1)} \; ;
\end{aligned}
$$

which is also shown in Fig. 10.11. It starts from the state $LDCQ_0$, where there is no active LDC within the macro-cell. Here, the control message $ldcq_e$ is sent to the gateway, denoting that the macro-cell can admit new LDC requests. Since there is no active LDC, this control message also means that the network is performing in normal mode. For all $LDCQ_j$, $1 \leq j \leq (N_{LDC}-1)$ the control signals $ldcq_{ne}$ is sent, indicating that further LDCs can be accepted and also, that the system is performing in power boost mode. When a LDC is admitted, the state changes from $LDCQ_j$ to $LDCQ_{j+1}$. In state $LDCQ_{N_{LDC}}$, the component sends the signal $ldcq_{full}$ which expresses that no further LDC can be accepted and again that the system is in the power boost mode. The control signals are used by MG when deciding whether a pending request for a LDC should be accepted.

A LDC is served when $LDCQ$ performs the shared action ex_wc_mcr with a far WCU. In any state $LDCQ_j$, the rate of this action is $j \times r_{ex\_wc\_mcr}$ where $r_{ex\_wc\_mcr}$ is the rate of serving an individual LDC. When a LDC is finished (action ex_wc_mcr), LDCQ is updated again; it moves from state $LDCQ_j$ to $LDCQ_{j-1}$.

The component $LDCQ$ knows the network's mode of operation and this component is responsible for setting the transmission rates for connections within the femto-cells. Such connections are related to either SCUs or WCUs. When in state $LDCQ_0$, the actions $ex\_wc\_f\_i\_nr$ and $ex\_sc\_f\_i\_nr$, $1 \leq i \leq N_f$ are enabled, which makes $WCUQ_i$ and femto-cell channels respectively enable the actions $ex\_wc\_f\_i\_nr$ and $ex\_sc\_f\_i\_nr$. On the other hand, when there is at least one LDC connection active (states $LDCQ_1 \cdots LDCQ_{N_{LDC}}$), the actions $ex\_wc\_f\_i\_po$ and $ex\_sc\_f\_i\_po$ are enabled, which makes $WCUQ_i$ and femto-cell channels enable $ex\_wc\_f\_i\_po$ and $ex\_sc\_f\_i\_po$. By the synchronisation between the femto-cell and macro-cell we capture the impact of the alterations in the interference levels on the experience of far users.

The mode alterations also affect the transmission rates offered to near users. For these users, the transmission rate is higher in the power boost mode than the normal mode (Sec. 10.2.3). The component $LDCQ$ sends control signals to the channels serving the near users to adjust their transmission rates. When it is in state $LDCQ_0$, the control signal ex_nr is sent, which indicates that the channels must perform the data exchange according at the rate r_{ex_nr} related to the normal mode. On the other hand, in states $LDCQ_1 \cdots LDCQ_{N_{LDC}}$, the signal ex_po is sent, which enforces the channels to exchange data at the higher rate r_{ex_po}.

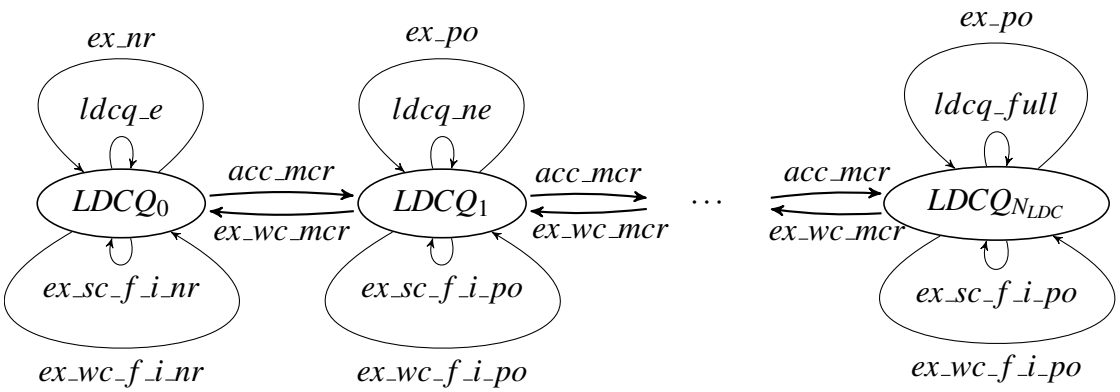


Figure 10.11: The symbolic behaviour of LDCQ. The control signals $ex\_sc\_f\_i\_nr$, $ex\_wc\_f\_i\_nr$, $ex\_sc\_f\_i\_po$ and $ex\_wc\_f\_i\_po$ are instantiated for all i : $1 \leq i \leq N_f$.

Macro-cell Channels for Near Users

The behaviour of a macro-cell channel is:

$$NUCH \stackrel{def}{=} (ex\_nr, r_{ex\_nr}).NUCH + (ex\_po, r_{ex\_po}).NUCH$$

which is also shown in Fig. 10.12. Depending on the current mode of operation, the component repeatedly undertakes either the action ex_nr or ex_po. The group of NUCHs collectively deal with the data exchanges requested by near users. We assume that when a near user needs the service, it acquires the first available channel without consulting with the gateway.

Macro-cell Composition

Using the behaviours stated for MG, $LDCQ$ and $NUCH$, the behaviour of the macro-cell is expressed as:

$$MC \stackrel{def}{=} (\ G\_Gateway\{\ MG[1]\ \} \underset{L_1}{\bowtie} LQ\{\ LDCQ_0[1]\ \}\ ) \underset{L_2}{\bowtie} NCH\{\ NUCH[N_{n\_ch}]\ \} \tag{10.6}$$

Figure 10.12: The behaviour of a macro-cell channel used for serving near users. The rate of the action ex_nr is r_{ex_nr} and the rate of ex_po is r_{ex_po} where $r_{ex\_po} > r_{ex\_nr}$.

where $G\_Gateway$, LQ and NCH are respectively the labels of the groups $MG[1]$, $LDCQ[1]$ and $NUCH[N_{n\_ch}]$, and $L_1 = \{ldcq_e, ldcq_{ne}, ldcq_{full}, mcr\_acc\}$ and $L_2 = \{ex\_po, ex\_nr\}$.

10.3.2.6 Capturing the Behaviour of the Network

Using the processes defined for a near user, a far user, a femto-cell and the macro-cell, the process that captures the dynamics of the whole network is defined as:

$$
\begin{aligned}
Network \stackrel{def}{=} & \left(Near\_users\{\, U_{n\_proc}[N_{nu}] \,\} \underset{A}{\bowtie} MC \right) \underset{B}{\bowtie} \\
& \Bigg[\left(FC\_1 \underset{L_1}{\bowtie} \left(U\_s\_1\{\, U_{s\_proc}[n_{s\_1}] \,\} \;||\; U\_w\_1\{\, U_{w\_proc}[n_{w\_1}] \,\} \right) \right) \;\; || \\
& \qquad \vdots \\
& \left(FC\_i \underset{L_i}{\bowtie} \left(U\_s\_i\{\, U_{s\_proc}[n_{s\_i}] \,\} \;||\; U\_w\_i\{\, U_{w\_proc}[n_{w\_i}] \,\} \right) \right) \;\; || \\
& \qquad \vdots \\
& \left(FC\_N_f \underset{L_{N_f}}{\bowtie} \left(U\_s\_N_f\{\, U_{s\_proc}[n_{s\_N_f}] \,\} \;||\; U\_w\_N_f\{\, U_{w\_proc}[n_{w\_N_f}] \,\} \right) \right) \Bigg]
\end{aligned}
\tag{10.7}
$$

where the cooperation sets are:

$$A = \{ex\_nr, ex\_po\}$$

$$B = \{\underbrace{req\_mcr, mcr\_acc, mcr\_rej}_{\text{control signals to WCU}}, \underbrace{ex\_wc\_mcr}_{\text{serving LDC}}\}$$

$$L_i = \{\underbrace{req\_f\_i, f\_i\_acc, f\_i\_rej}_{\text{femto-cell dealing with a WCU}}, \underbrace{ex\_sc\_f\_i\_nr, ex\_sc\_f\_i\_po, ex\_wc\_f\_i\_nr, ex\_wc\_f\_i\_po}_{\text{control signals informing the current mode of operation}}\}$$

and N_{nu} denotes the number of near users, for i: $1 \leq i \leq N_f$; and n_{s_i} and n_{w_i} represent respectively the number of SCUs and WCUs around femto-cell i.

10.3.2.7 Model Parameters

We consider a network with one macro-cell and two femto-cells ($N_f = 2$). Thus the PEPA process of the network is:

$$Network \stackrel{def}{=} \left(Near\_users\{\, U_{n\_proc}[N_{nu}] \,\} \underset{A}{\bowtie} MC \right) \underset{B}{\bowtie}$$

parameter	description	value
N_{nu}	number of users near to the macro-cell	150
N_{s_i}	number of SCUs around femto-cell $i : 1 \leq i \leq N_f$	30
N_{w_i}	number of WCUs around femto-cell $i : 1 \leq i \leq N_f$	2
N_{n_ch}	number of channels available to the macro-cell for serving near users	20
N_{s_ch}	number of channels available to a femto-cell $i : 1 \leq i \leq N_f$ for serving local SCUs	4
N_{f_w}	capacity of femto-cells with respect to connections to WCUs	1
N_{LDC}	capacity of macro-cell with respect to LDCs	1

Table 10.1: Parameters related to the number of instances within the $Near\_users$, U_s_i, U_w_i, NCH and FCH_i groups (for $1 \leq i \leq N_f$); the capacity of femto-cells' $WCUQ$s; and the capacity of the macro-cell's $LDCQ$.

$$\left[\left(FC\_1 \underset{L_1}{\bowtie} \left(U\_s\_1\{\, U_{s\_proc}[n_{s\_1}] \,\} \;||\; U\_w\_1\{\, U_{w\_proc}[n_{w\_1}] \,\} \right) \right) \;||\; \left(FC\_2 \underset{L_2}{\bowtie} \left(U\_s\_N_f\{\, U_{s\_proc}[n_{s\_2}] \,\} \;||\; U\_w\_N_f\{\, U_{w\_proc}[n_{w\_2}] \,\} \right) \right) \right] \tag{10.8}$$

The parameters related to the populations of different types of users are shown in Table 10.1. In this table, we also show the capacities N_{f_w} and N_{LDC} defined respectively for the femto-cells and macro-cell. The parameters related to the rates of the activities are shown in Table 10.2. We are comparing our model to the one presented in [43], In that work, the network parameters are specified using notions such as wireless link power, signal to interference and noise ratio, etc. which are usually used in the field of system engineering. It was a hard task to translate these parameters into the notion of action rates accepted in PEPA language. One approach was to use the *linear rate-SINR transformation*, which is also proposed by the same authors. This transformation maps measures related to signal strength to those related to the transmission and service rates. Providing the details of our transformation and its rigorous validation is out of the scope of this thesis. Nevertheless, using our parameters, we can still show the usefulness of our analysis methods and how they are used in dealing with complex models.

In Table 10.2 it is important to observe the following points:

parameter	description	value
r_{ctrl}	the transmission rate for exchanging control messages	$1e10$
r_p	the rate of undertaking the action *proc* by a near user	5
r_{s_p}	the rate of undertaking the action *proc* by a SCU	5
r_{w_p}	the rate of undertaking the action *proc* by a WCU	0.1
$r_{ex\_wc\_f\_i\_nr}$	the rate at which a WCU is served in the normal mode	2.5
$r_{ex\_wc\_f\_i\_po}$	the rate at which a WCU is served in the power boost mode	1.5
$r_{ex\_sc\_f\_i\_nr}$	the rate at which a SCU is served in the normal mode	20
$r_{ex\_sc\_f\_i\_po}$	the rate at which a SCU is served in the power boost mode	10
$r_{ex\_wc\_mcr}$	the rate at which a WCU is served by the macro-cell	0.2
r_{ex_nr}	the rate at which a near user is served in the normal mode	15
r_{ex_po}	the rate at which a near user is served in the power boost mode	22

Table 10.2: Parameters related to the rates of the activities. The unit of time is a second.

1. $r_{ex\_wc\_f\_i\_nr} < r_{ex\_wc\_f\_i\_po}$ and $r_{ex\_sc\_f\_i\_nr} < r_{ex\_sc\_f\_i\_po}$: the femto-cell serves its users at a lower rate in the power boost mode compared to the normal mode.

2. $r_{ex\_wc\_f\_i\_nr} < r_{ex\_sc\_f\_i\_nr}$ and $r_{ex\_wc\_f\_i\_po} < r_{ex\_sc\_f\_i\_po}$: a femto-cell serves WCUs at a lower rate than SCUs.

3. $r_{ex\_nr} < r_{ex\_po}$: near users are served at a higher rate in the power boost mode compared to the normal mode.

4. $r_{ex\_wc\_mcr}$ is significantly smaller than other types of communication. This means that LDCs take longer to terminate.

The above observations are in compliance with the model in [43].

10.3.2.8 State Vector

We define the state vector $\boldsymbol{\xi}_{net} = \langle\ \boldsymbol{\xi}_{n\_u}, \boldsymbol{\xi}_{s\_1}, \boldsymbol{\xi}_{w\_1}, \boldsymbol{\xi}_{s\_2}, \boldsymbol{\xi}_{w\_2}, \boldsymbol{\xi}_{f\_1}, \boldsymbol{\xi}_{f\_2}, \boldsymbol{\xi}_{mcr}\ \rangle$ for capturing the state of the model. The sub-vectors which compose $\boldsymbol{\xi}_{net}$ are:

1. $\boldsymbol{\xi}_{n\_u} = \langle \xi_{n\_proc}, \xi_{n\_exch} \rangle$: captures the state of the near users.

2. For $i \in \{1,2\}$, $\boldsymbol{\xi}_{s\_i} = \langle \xi_{s\_i\_proc}, \xi_{s\_i\_exch} \rangle$: captures the state of SCUs close to femto-cell i.

3. For $i \in \{1,2\}$, $\boldsymbol{\xi}_{w\_i} = \langle\ \xi_{w\_i\_proc}, \xi_{w\_req\_f\_i}, \xi_{w\_wait\_f\_i}, \xi_{w\_exch\_f\_i}, \xi_{w\_i\_req\_mcr}, \xi_{w\_i\_wait\_mcr}, \xi_{w\_i\_exch\_mcr}\ \rangle$: captures the states of the WCUs around femto-cell i.

4. For $i \in \{1,2\}$, $\boldsymbol{\xi}_{f\_i} = \langle\ \xi_{FG\_i\_idle}, \xi_{FG\_i\_check}, \xi_{FG\_i\_acc}, \xi_{FG\_i\_rej}, \xi_{WCUQ\_i\_0}, \xi_{WCUQ\_i\_1}, \xi_{FSCCH\_i}\ \rangle$: captures the states of the components within femto-cell i. Note that each femto-cell can only serve at most one WCU. Thus, $WCUQ$ has only two states $WCUQ_0$ and $WCUQ_1$.

5. $\boldsymbol{\xi}_{mcr} = \langle\ \xi_{MG\_idle}, \xi_{MG\_check}, \xi_{MG\_acc}, \xi_{MG\_rej}, \xi_{LDCQ\_0}, \xi_{LDCQ\_1}, \xi_{NUCH}\ \rangle$: captures the states of the components within the macro-cell.

The state vector $\boldsymbol{\xi}_{net}$ has 27 state variables in total.

10.3.2.9 Categorising Groups into being Small and Large

The set of groups in the model is $\mathcal{G}(net)$:

$$\mathcal{G}(net) = \{\ Near\_users,\ \overbrace{U\_s\_1, U\_w\_1}^{\text{users at femto-cell 1}},\ \overbrace{U\_s\_1, U\_w\_1}^{\text{users at femto-cell 2}},\ \overbrace{M\_Gateway, LQ, NCH}^{macro-cell},$$
$$\underbrace{F\_1\_Gateway, WQ\_1, FCH\_1}_{\text{femto-cell 1}}, \underbrace{F\_2\_Gateway, WQ\_2, FCH\_2}_{\text{femto-cell 2}}\ \} \quad (10.9)$$

We define the partition $\Delta_{\mathcal{G}} = \{\mathcal{G}_s(net), \mathcal{G}_l(net)\}$ where $\mathcal{G}_s(net)$ is:

$$\mathcal{G}_s(net) = \{\ U\_w\_1, U\_w\_1, M\_Gateway, LQ, F\_1\_Gateway,$$
$$WQ\_1, F\_2\_Gateway, WQ\_2, NCH, FCH\_1, FCH\_2\ \} \quad (10.10)$$

and $\mathcal{G}_l(net)$ is:

$$\mathcal{G}_l(net) = \{Near\_users, U\_s\_1, U\_s\_2\} \quad (10.11)$$

Our large groups correspond to users near to the macro-cell and SCUs close to the femto-cells. Using $\Delta_{\mathcal{G}}$, the state vector is divided as $\boldsymbol{\xi} = \langle \boldsymbol{\xi}^s, \boldsymbol{\xi}^l \rangle$ where $\boldsymbol{\xi}^s$:

$$i \in \{1,2\}\ : \boldsymbol{\xi}^s = \langle\ \xi_{w\_i\_proc}, \xi_{w\_req\_f\_i}, \xi_{w\_wait\_f\_i}, \xi_{w\_exch\_f\_i}, \xi_{w\_i\_req\_mcr}, \xi_{w\_i\_wait\_mcr},$$
$$\xi_{WCUQ\_i\_1}, \xi_{WCUQ\_i\_0}, \xi_{w\_exch\_mcr}, \xi_{FG\_i\_idle}, \xi_{FG\_i\_check}, \xi_{FG\_i\_acc}, \xi_{FG\_i\_rej},$$
$$\xi_{MG\_idle}, \xi_{MG\_check}, \xi_{MG\_acc}, \xi_{MG\_rej}, \xi_{LDCQ\_0}, \xi_{LDCQ\_1}, \xi_{FSCCH\_i}, \xi_{NUCH}\ \rangle \quad (10.12)$$

captures the state of the small groups in $\mathcal{G}_s(net)$, and

$$i \in \{1,2\} \quad \boldsymbol{\xi}^l = \langle\, \xi_{n\_proc}\,, \xi_{n\_exch}\,, \xi_{s\_i\_proc}\,, \xi_{s\_i\_exch}\,\rangle \tag{10.13}$$

captures the state of the large groups in $\mathcal{G}_l(M)$.

10.3.2.10 Initial state

We assume that the system starts from the initial state where no user is being served in the system; SCUs are in state $U_{s\_i\_proc}$; WCUs are in state $U_{w\_i\_proc}$; near users are in state U_{n_proc}; the femto-cells' gateways are in state FG_{idle}; $WCUQ$s are in state $WCUQ_{i\_0}$; the macro-cell's gateway is in state MG_{idle}; and finally, the $LDCQ$ is in state $LDCQ_0$.

10.4 Model Analysis

In our model, there are two WCUs around each femto-cell whilst each femto-cell has the capacity of serving at most one WCU at any given time. A LDC is activated when one WCU is already being served by a femto-cell, the second WCU arrives, gets rejected by the femto-cell and is accepted by the macro-cell. The associated change of mode (from normal to power boost) causes a deterioration of the femto-cell's local connections and a sharp increase in $\xi_{s\_1\_exch}$ and $\xi_{s\_2\_exch}$ (the number of SCU waiting to exchange data with their femto-cell). This also causes a decrease in ξ_{n_exch} (the number of near users exchanging data with the macro-cell). These fluctuations are shown in Fig. 10.13 and 10.14, which show two sample trajectories of the system. In these figures, particularly note the changes in LDCQ (top left figure) and the corresponding alterations in $\xi_{s\_1\_exch}$, $\xi_{s\_2\_exch}$ and ξ_{n_exch}. The sharp drops in ξ_{n_exch} are correlated to the activation of LDCs, which simultaneously, causes $\xi_{s\_1\_exch}$, $\xi_{s\_2\_exch}$ to experience higher values. The termination of the LDCs has the reverse effect.

We performed a exploratory analysis using the method of stochastic simulation and obtained the results shown in Fig. 10.15 and 10.16, related to variables $\xi_{s\_1\_exch}$ and ξ_{n_exch}. Fig 10.15A shows the probability distribution $\mathbb{P}_t(\xi_{s\_1\_exch})$ for $t = 20$. Here, note the heavy tail towards the right, which corresponds to values that $\xi_{s\_1\_exch}$ takes in the occasional power boost mode. In Fig. 10.15B, we present the probability masses that each of the operational modes separately contribute to the distri-

bution $\mathbb{P}_t(\xi_{s\_1\_exch})$. Transforming this distribution to the conditional form, we obtain $\mathbb{P}_t(\xi_{s\_1\_exch} \mid LDCQ_0)$ and $\mathbb{P}_t(\xi_{s\_1\_exch} \mid LDCQ_1)$, the conditional probability of $\xi_{s\_1\_exch}$ given each of the operational modes. These are shown in Fig. 10.15C and 10.15D. We derived similar measures for the state variable ξ_{n_exch} also at $t = 20$, which are shown in Fig 10.16A, 10.16B, 10.16C and 10.16D. Our exploratory evaluation shows that the analysis method used for this model is required to capture the random activation and termination of LDCs and also, the distinct features of the distributions $\mathbb{P}_t(\xi_{s\_1\_exch})$, $\mathbb{P}_t(\xi_{s\_2\_exch})$ and $\mathbb{P}_t(\xi_{n\_exch})$.

Performance Evaluation Questions

Given the behaviour observed in Fig. 10.13 and 10.14, we aim to investigate the following questions in our performance evaluation:

1. The probability of having active WCU connection within a femto-cell.
2. The transient probability of having active LDCs within the network.
3. The transient probability of a WCU being rejected by its local femto-cell.
4. The transient probability of a WCU being rejected by both its local femto-cell and the macro-cell.
5. The impact of the activation and termination of LDCs on near users (the state variable ξ_{n_exch}).
6. The impact of the activation and termination of LDCs on SCUs (the state variable $\xi_{s\_1\_exch}$ and $\xi_{s\_2\_exch}$).

Due to the symmetry of the femto-cells in our model, the corresponding state variables related to two different femto-cells have the same behaviours. For instance, the state variable $\xi_{WCUQ\_1\_0}$ related to the first femto-cell has the same probability distribution as $\xi_{WCUQ\_2\_0}$, related to the second femto-cell. Furthermore, the users of these femto-cells also have identical evolutions. For instance, the state variable $\xi_{s\_1\_exch}$ related to the SCUs around the first femto-cell has the same evolution as $\xi_{s\_2\_exch}$, which is related to SCUs near the second femto-cell. To save some space, we present the state variables and their analysis results with an index $i \in \{1,2\}$, which refers to the femto-cell they belong to.

We apply our analysis of conditional moments of up to the second order to this model. The results will be illustrated in the next sections.

10.4.1 Aggregation

Our model satisfies the two conditions that are required by our aggregation method. First, the probability of the femto-cell and macro-cell channels being idle is close to zero as SCUs and near users introduce continuous demand. Second, the users undertake the data exchanges passively and the rate is decided by the channel components which are within small groups.

Our aggregation process has the following steps. Using the partition $\Delta_{\mathcal{G}}$ on the group label, we form the partition $\Delta_{\mathcal{A}} = \{\overrightarrow{\mathcal{A}}^*_s(Net), \overrightarrow{\mathcal{A}}^*_{sl}(Net), \overrightarrow{\mathcal{A}}^*_l(Net)\}$ over the model's set of action types $\overrightarrow{\mathcal{A}}^*(Net)$ where we have:

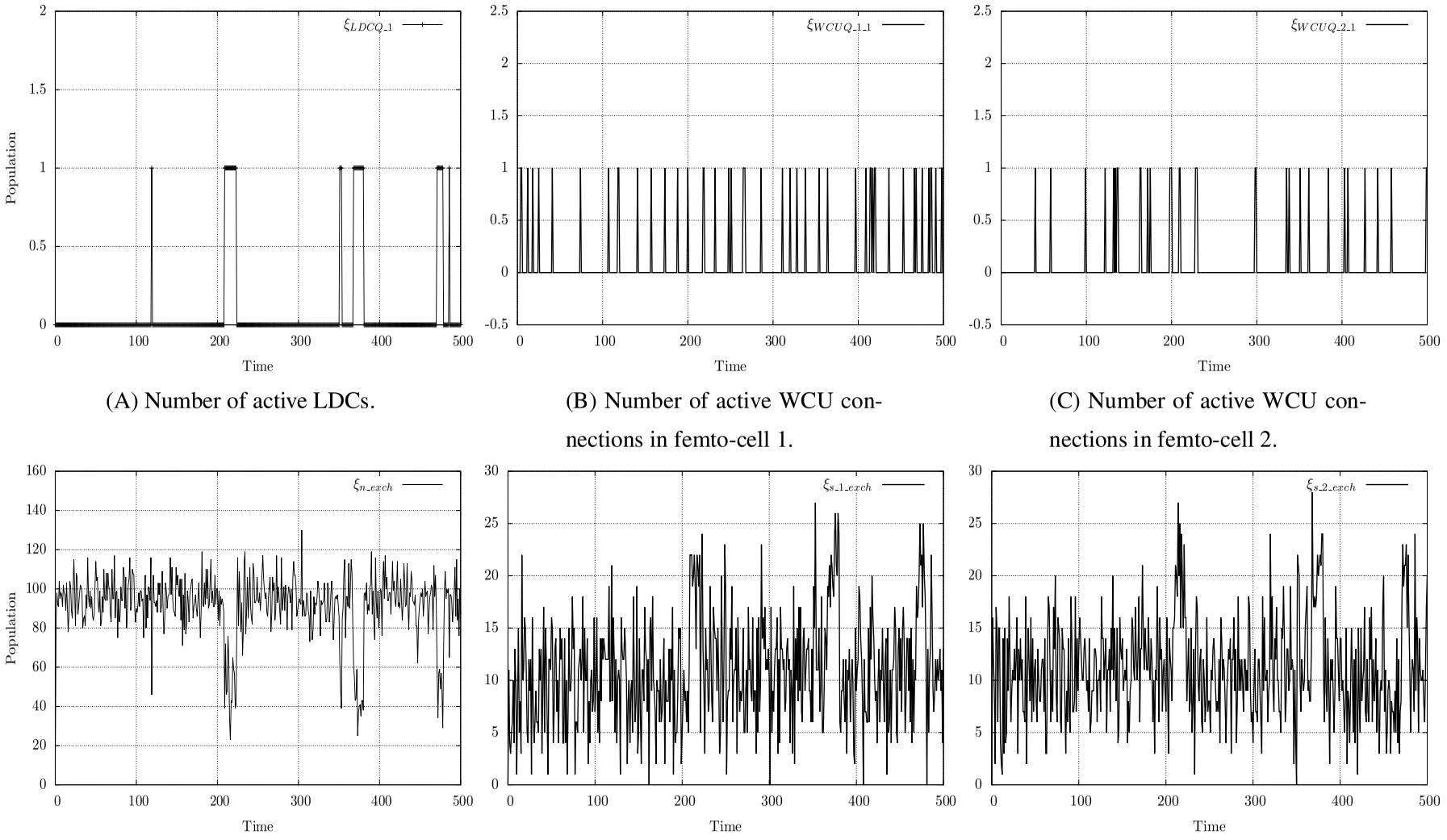
$$\begin{aligned}
\overrightarrow{\mathcal{A}}^*_s(Net) &= \{\ proc\ ,\ req_f_i\ ,\ f_i_acc\ ,\ f_i_rej\ ,\ ex_wc_f_i_po\ ,\ ex_wc_f_i_nr\ ,\\
&\qquad req_mcr\ ,\ mcr_acc\ ,\ mcr_rej\ ,\ ex_wc_mcr\ ,\ wcuq_{i_ne}\ ,\ wcuq_{i_e}\ ,\\
&\qquad wcuq_{i_full}\ ,\ ldcq_e\ ,\ ldcq_{ne}\ ,\ ldcq_{full}\ \}\\
\overrightarrow{\mathcal{A}}^*_{sl}(Net) &= \{\ ex_nr\ ,\ ex_po\ ,\ ex_sc_f_i_nr\ ,\ ex_sc_f_i_po\ \}\\
\overrightarrow{\mathcal{A}}^*_l(Net) &= \{\ proc\ \}
\end{aligned}$$

Then, we apply the aggregation reduction rules (shown in Eq. 4.3) to derive the reduced system equation of the aggregated model:

$$\begin{aligned}
Net \stackrel{def}{=}\ MC \underset{B}{\bowtie} \Bigg[&\Big(FC_1 \underset{L_1}{\bowtie} \big(U_w_1\{\, U_{w_proc}[n_{w_1}]\,\}\big)\Big) \ \ ||\\
&\Big(FC_2 \underset{L_2}{\bowtie} \big(U_w_N_f\{\, U_{w_proc}[n_{w_2}]\,\}\big)\Big)\Bigg] \qquad (10.14)
\end{aligned}$$

Next, we generate the aggregated state space D^{agg}_{Net} by applying the count-oriented semantics (Sec. 4.3.2) to the reduced system equation. Finally, the underlying C-K equations of D^{agg}_{Net} are constructed and solved to find the transient evolution of $\mathbb{P}_t(\boldsymbol{\beta})$, $\boldsymbol{\beta} \in D^{agg}_{Net}$, which denotes the model's marginal probability distribution over $\boldsymbol{\xi}^s$.

The distribution $\mathbb{P}_t(\boldsymbol{\beta})$ is sufficient for answering the first four performance questions presented on Page 242. The probability of having at least one WCU being served by femto-cell $i \in \{1,2\}$ is calculated by summing over the states where $WCUQ_i = WCUQ_{i\_1}$. In Fig 10.17A, the evolution of $\mathbb{P}_t(WCUQ\_i = WCUQ_{i\_1})$ is illustrated. In



(A) Number of active LDCs.

(B) Number of active WCU connections in femto-cell 1.

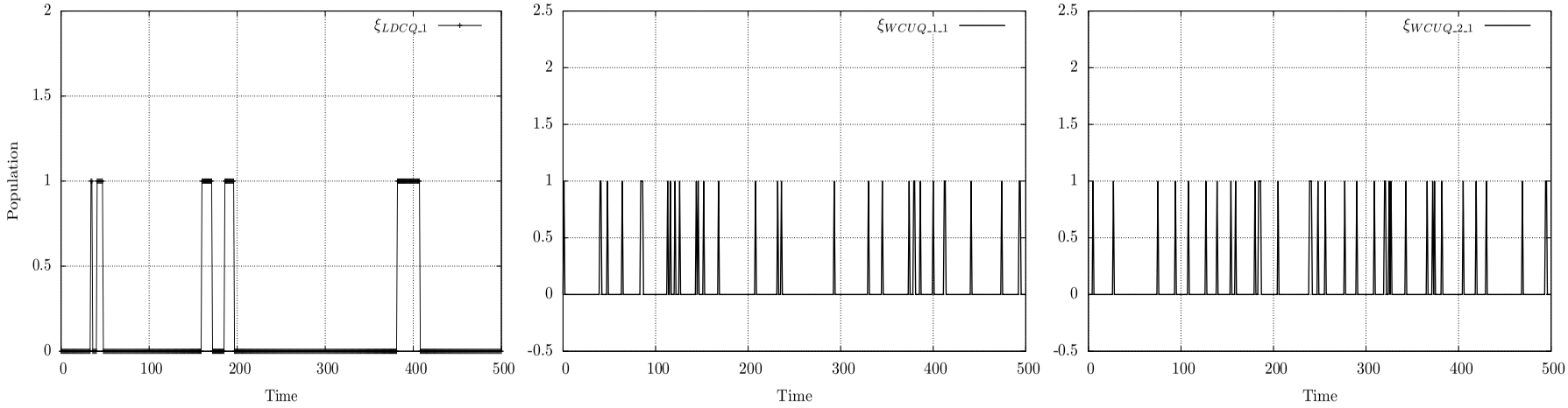
(C) Number of active WCU connections in femto-cell 2.

(D) Number of near users waiting for service from the macro-cell.

(E) Number of SCUs waiting for service from femto-cell 1.

(F) Number of SCUs waiting for service from femto-cell 2.

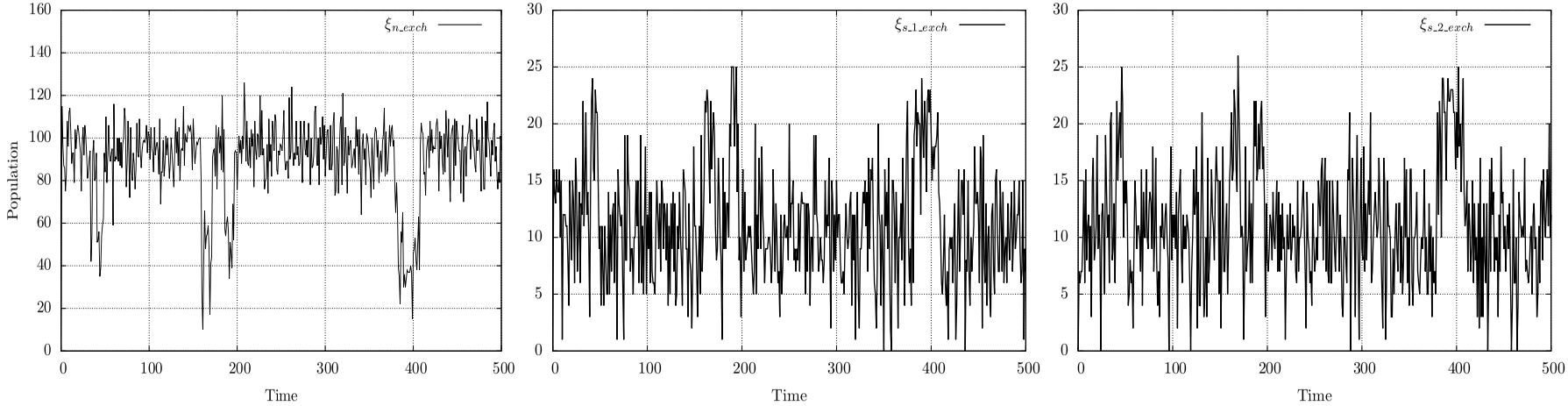
Figure 10.13: The evolution of some of the model's key variables (ξ_{LDCQ_1}, $\xi_{WCUQ\_1\_1}$, $\xi_{WCUQ\_2\_1}$, ξ_{n_exch}, $\xi_{s\_1\_exch}$, $\xi_{s\_2\_exch}$) in the first trajectory.



(A) Number of active LDCs.

(B) Number of active WCU connections in femto-cell 1.

(C) Number of active WCU connections in femto-cell 2.

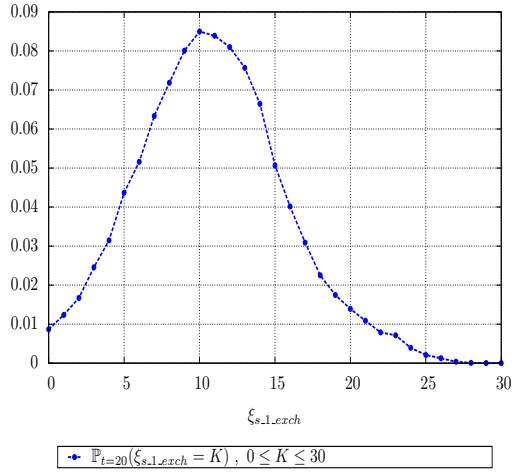


(D) Number of near users waiting for service from the macro-cell.

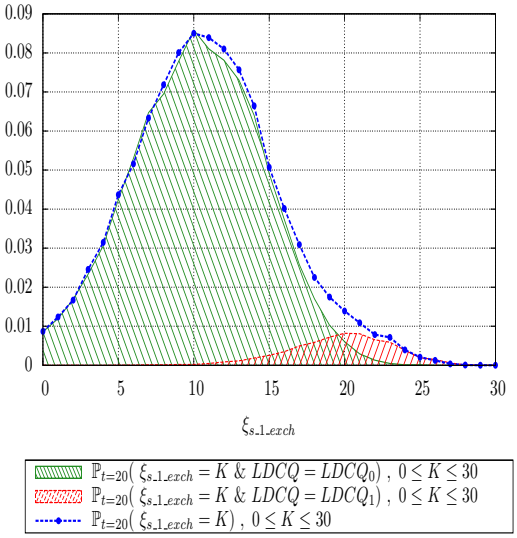
(E) Number of SCUs waiting for service from femto-cell 1.

(F) Number of SCUs waiting for service from femto-cell 2.

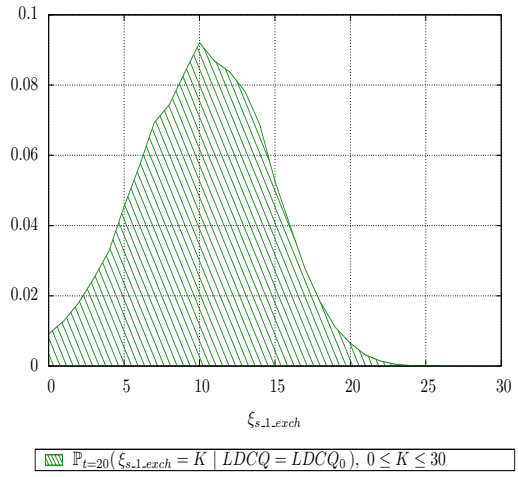
Figure 10.14: The evolution of state variables ξ_{LDCQ_1}, $\xi_{WCUQ\_1\_1}$, $\xi_{WCUQ\_2\_1}$, ξ_{n_exch}, $\xi_{s\_1\_exch}$, $\xi_{s\_2\_exch}$ in the second trajectory.



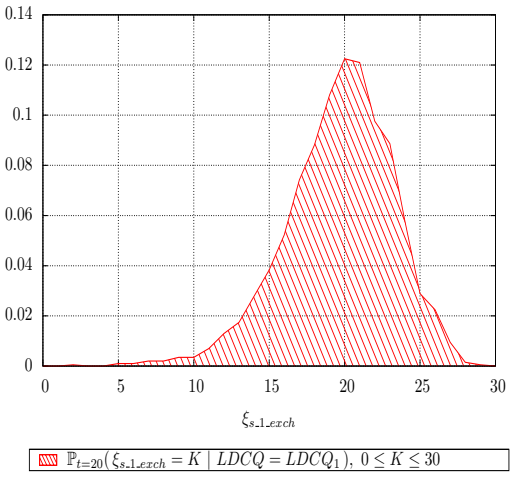
(A) Approximate probability distribution associated with $\xi_{s\_1\_exch}$ obtained using stochastic simulation.



(B) Separate probability mass contributions in the distribution of $\xi_{s\_1\_exch}$ with respect to the different operational modes.

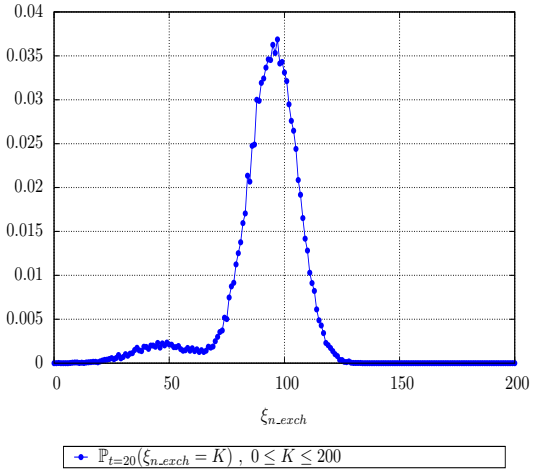


(C) The conditional distribution of $\xi_{s\_1\_exch}$ given that there is no active LDC.

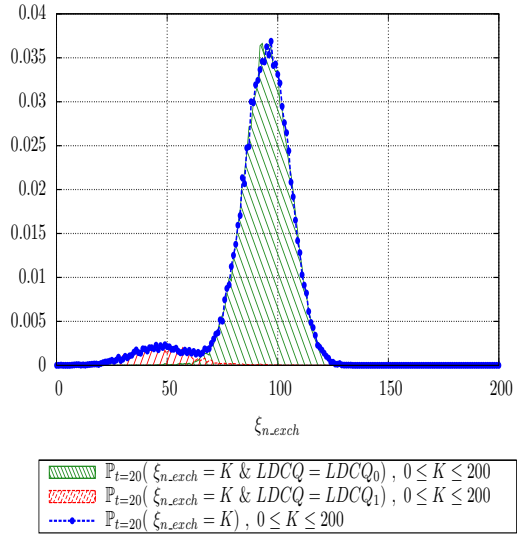


(D) The conditional distribution of $\xi_{s\_1\_exch}$ given that there is at least one active LDC.

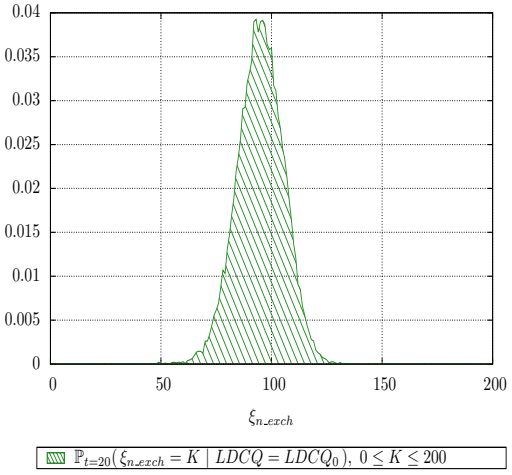
Figure 10.15: The behaviour of $\xi_{s\_1\_exch}$ (or $\xi_{s\_2\_exch}$), i.e. SCUs exchanging with their femto-cell, at $t = 20$ derived using 300,000 runs of stochastic simulation.



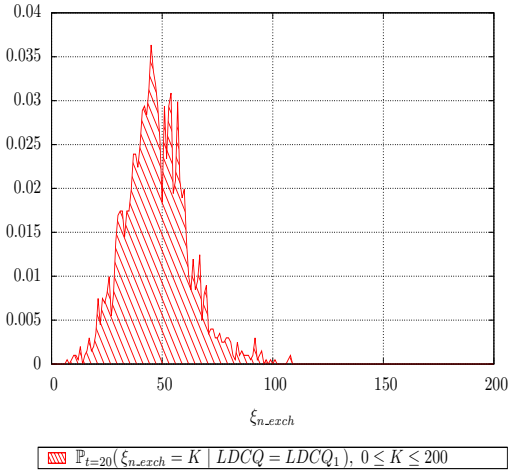
(A) Approximate probability distribution associated with ξ_{n_exch} obtained using stochastic simulation.



(B) Separate probability mass contributions in the distribution of ξ_{n_exch} with respect to the different operational modes.



(C) The conditional distribution of ξ_{n_exch} given that there is no active LDC.



(D) The conditional distribution of ξ_{n_exch} given that there is at least one active LDC.

Figure 10.16: The behaviour of ξ_{n_exch}, i.e. near users exchanging with the macro-cell, at $t = 20$ derived using 300,000 runs of stochastic simulation.

Fig 10.17B, we consider the second question; what is the probability of having active LDCs within the network? Similarly, this can be answered by summing the probabilities of states where $LDCQ = LDCQ_1$. The figure also shows the probability of being in each of the normal and power boost operational modes[1]. The third question essentially concerns the calculation of the *loss ratio* with respect to WCUs. Let E_f denote the probabilistic event of the rejection of a WCU from its femto-cell. The probability of E_f can be calculated by the ratio of the throughput observed for action f_i_acc to that of the action f_i_rej. Moreover, let E_m denote the probabilistic event of the rejection of a WCU from the macro-cell. Similarly, the probability of E_f can be calculated by the ratio of the throughput observed for action mcr_acc to that of the action mcr_rej. The probability distribution $\mathbb{P}_t(\boldsymbol{\beta})$ was sufficient to calculate the required throughputs and the evolution of $\mathbb{P}_t(E_f)$ and $\mathbb{P}_t(E_m)$ are respectively shown in Fig 10.17C and 10.17D.

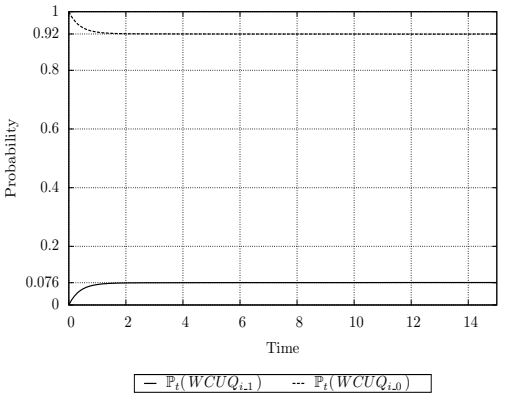
The model's complete state space has 22^6 states. This precluded the derivation of the above probability measures by the analysis of this state space. By comparison, the aggregated state space has only 112 states. This highlights the advantage of using the aggregation for the analysis of the model's small groups.

Now we proceed to consider the fourth and fifth performance questions on Page 242. In order to answer them, we apply the analysis of conditional expectation and second-order conditional moments. These will be presented in the next two sections.

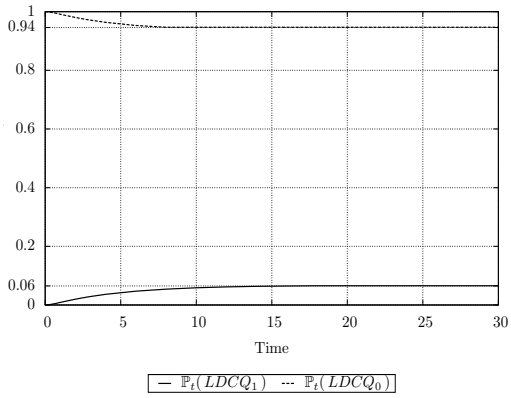
10.4.2 Analysis of the Conditional Expectations

By the analysis of conditional expectations, we find the expectations of the conditional distributions $\mathbb{P}_t(\xi \mid \boldsymbol{\beta}), \xi \in \boldsymbol{\xi}^l = \langle \xi_{n\_proc}, \xi_{n\_exch}, \xi_{s\_1\_proc}, \xi_{s\_1\_exch}, \xi_{s\_2\_proc}, \xi_{s\_2\_exch} \rangle$, $\boldsymbol{\beta} \in D_{Net}^{agg}$. The state vector $\boldsymbol{\xi}^l$ has six state variables. The aggregated state space D_{Net}^{agg} has 112 states. Therefore, the analysis of conditional expectation gives rise to $6 \times 112 = 672$ conditional expectation variables of the form $\mathbb{E}_t[\xi \mid \boldsymbol{\beta}]$. To find their evolution, a system of DAEs is constructed where, for each conditional variable, one equation is formed of the format of Eq.(6.39). The Matlab code which captures this system of equations was generated from the model and can be found in the online supplements at [77]. Due to the length of these equations, we do not present them here. Using the model's initial state described on Page 241, the initial conditional

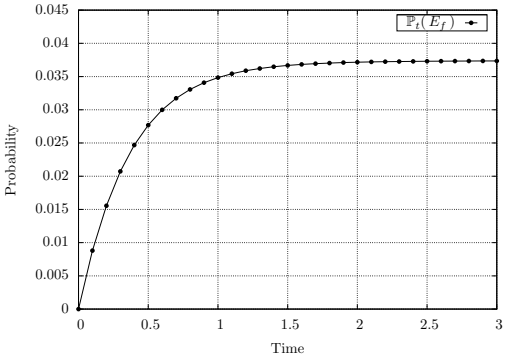
[1] Hereafter, the event that $LDCQ = LDCQ_0$ is shortly written as $LDCQ_0$. We use similar shorthand for other components.



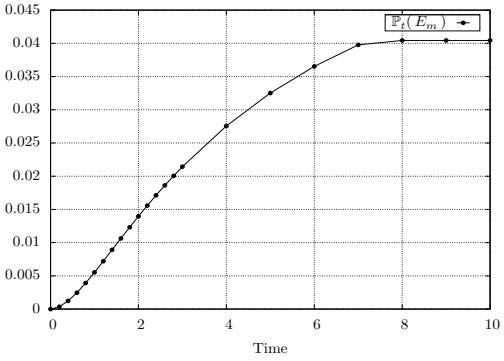
(A) The evolution of the probability of $WCUQ\_i$ being in states $WCUQ_{i\_0}$ and $WCUQ_{i\_1}$.



(B) The evolution of the probability of LDCQ being in states $LDCQ_0$ and $LDCQ_1$.



(C) The probability of a WCU being rejected from a femto-cell.



(D) The probability of a WCU being rejected from a macro-cell.

Figure 10.17: The results of the analysis of the aggregated model.

expectations (initial values of the DAEs) were obtained approximately at $t_{init} = 0.1$, using the *approximate initial value extraction method* (Sec. 8.4.3). This point of time is sufficiently long so that $\forall\, \boldsymbol{\beta} \in D_{Net}^{agg} \;:\; \mathbb{P}_{t_{init}}(\boldsymbol{\beta}) \neq 0$. The transient solution was then obtained by using the Matlab DAE solver for our time period of interest $t_{init} \leq t \leq 20$. The solution related to all 672 states is available at [77].

In Fig. 10.15 and 10.16, we showed that one key cause of $\boldsymbol{\xi}^l$ exhibiting a highly stochastic behaviour is the switching between the normal and power boost operational modes. The conditional distributions concerning these modes, $\mathbb{P}_t(\xi_{n\_exch} \mid LDCQ_0)$, $\mathbb{P}_t(\xi_{n\_exch} \mid LDCQ_1)$, $\mathbb{P}_t(\xi_{s\_i\_exch} \mid LDCQ_0)$, $\mathbb{P}_t(\xi_{s\_i\_exch} \mid LDCQ_1)$ were shown in Fig. 10.15C, 10.15D, 10.16C and 10.16D. The conditional expectations $\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\beta}\right]$: $\boldsymbol{\beta} \in D_{Net}^{agg}$ *related to the aggregated states*, and the probability distribution over the aggregated states are sufficient information to calculate the conditional expectations of these distributions *related to the modes*. Let the normal operational mode be denoted by M_{nr} and formally defined as a sub-set of states in D^{agg} where $LDCQ = LDCQ_0$: $M_{nr} = \{\, \boldsymbol{\beta}_i \in D_{Net}^{agg} \mid\; LDCQ = LDCQ_0 \text{ in } \boldsymbol{\beta}_i \,\}$. Recall that $LDCQ$ is within the small groups and therefore, M_{nr} can be efficiently found by a reachability analysis of the aggregated state space. Similarly, the power boost mode is denoted by $M_{po} = \{\, \boldsymbol{\beta}_i \in D_{Net}^{agg} \mid\; LDCQ = LDCQ_1 \text{ in } \boldsymbol{\beta}_i \,\}$. The conditional expectation $\mathbb{E}_t[\xi \mid LDCQ_0]$, $\xi \in \boldsymbol{\xi}^l$ is found by:

$$\mathbb{E}_t[\xi \mid LDCQ_0] = \sum_{\langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l \rangle \in D \;,\; \xi \in \boldsymbol{\xi}_i^l} \xi \cdot \mathbb{P}_t(\langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l \rangle \mid LDCQ_0) = \frac{\sum \xi \cdot \mathbb{P}_t(\langle \boldsymbol{\xi}_i^s, \boldsymbol{\xi}_i^l \rangle, LDCQ_0)}{\mathbb{P}_t(LDCQ_0)}$$

$$= \frac{\sum \xi \cdot \mathbb{P}_t(\langle \boldsymbol{\beta}, \boldsymbol{\xi}_i^l \rangle, LDCQ_0)}{\mathbb{P}_t(LDCQ_0)} = \frac{\sum \xi \cdot \mathbb{P}_t(\boldsymbol{\xi}_i^l \mid \boldsymbol{\beta}) \cdot \mathbb{P}_t(\boldsymbol{\beta})}{\mathbb{P}_t(LDCQ_0)}$$

$$= \frac{\sum_{\boldsymbol{\beta} \in M_{nr}} \overbrace{\mathbb{E}_t[\xi \mid \boldsymbol{\beta}]}^{\text{conditional expectation}} \times \overbrace{\mathbb{P}_t(\boldsymbol{\beta})}^{\mathbb{P}_t(\boldsymbol{\beta}) \;,\; \boldsymbol{\beta} \in D_{Net}^{agg},\; \text{derived by analysis of } D_{Net}^{agg}}}{\underbrace{\mathbb{P}_t(LDCQ_0)}_{\text{derived by analysis of } D_{Net}^{agg}}} \tag{10.15}$$

Similarly, the conditional expectation $\mathbb{E}_t[\xi \mid LDCQ_1]$, $\xi \in \boldsymbol{\xi}^l$ is found by:

$$\mathbb{E}_t[\xi \mid LDCQ_1] = \frac{\sum_{\boldsymbol{\beta} \in M_{nr}} \mathbb{E}_t[\xi \mid \boldsymbol{\beta}] \;\times\; \mathbb{P}_t(\boldsymbol{\beta})}{\mathbb{P}_t(LDCQ_1)} \tag{10.16}$$

	Conditional expectation	MCM	Simulation	Error (%)
$LDCQ = LDCQ_0$	$\mathbb{E}[\xi_{n\_exch} \mid LDCQ_0]$	97.83	95.111	1
	$\mathbb{E}[\xi_{n\_exch} \mid LDCQ_1]$	51.06	45.769	10
$LDCQ = LDCQ_1$	$\mathbb{E}[\xi_{s\_i\_exch} \mid LDCQ_0]$	10.65	10.204	4
	$\mathbb{E}[\xi_{s\_i\_exch} \mid LDCQ_1]$	19.618	18.734	4

Table 10.3: Comparison between $\mathbb{E}_t[\xi \mid LDCQ_k]$, $\xi \in \boldsymbol{\xi}^l$, $k \in \{0,1\}$ obtained by the method of conditional expectation and the same measures derived using 300000 runs of stochastic simulation. The simulations were done using PEPA Eclipse plug-in [93].

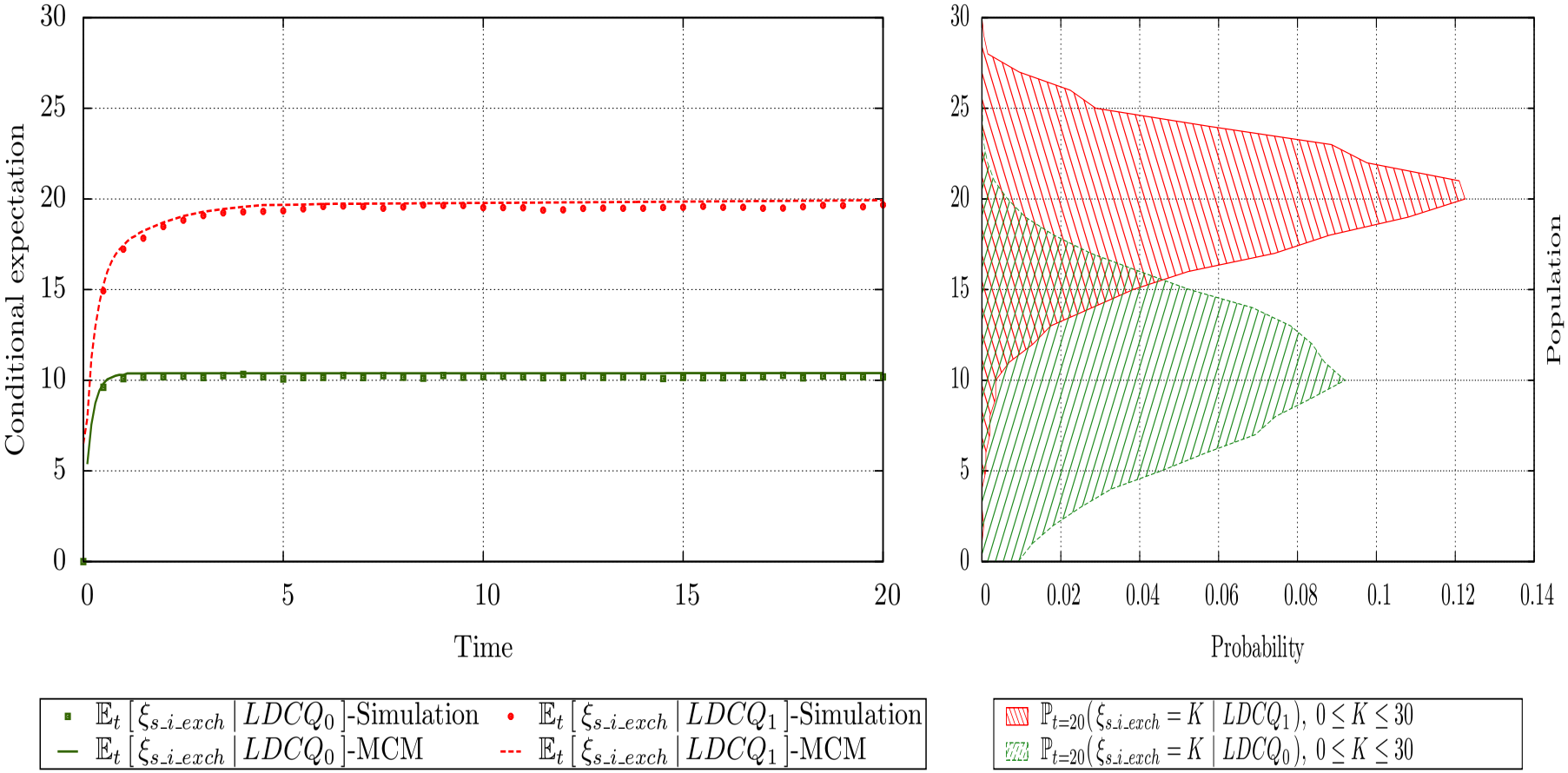
Using Eq.(10.15) and (10.16) we derived the evolution of the conditional expectations $\mathbb{E}_t[\boldsymbol{\xi} \mid LDCQ_k]$, $\xi \in \boldsymbol{\xi}^l$, $k \in \{1,2\}$. The results are shown in Fig 10.18A and 10.19A. To check the accuracy of our result, we measured the same expectations using stochastic simulation. This involved running 300,000 simulation which took nearly 3 days on a 2.8 GHz machine. The result of this comparison is shown in Table 10.3. The comparison shows that the MCM solution has acceptable accuracy; the maximum error was 10% for $\mathbb{E}_t[\xi_{n\_exch} \mid LDCQ_1]$ for $t = 1.5$. As stated, the tool we have for the derivation of MCM equations is in its early stages of development and we conjecture that the discrepancies seen in solution will be resolved in the near future. The conditional expectations were calculated significantly faster in approximately 20 minutes.

10.4.3 Analysis of Second-Order Conditional Moments

10.4.3.1 Moments

In this analysis, we derive the variances and covariances of the conditional distributions $\mathbb{P}_t(\boldsymbol{\xi}^l \mid \boldsymbol{\beta})$ for $\boldsymbol{\xi}^l = \langle \xi_{n\_proc}, \xi_{n\_exch}, \xi_{s\_1\_proc}, \xi_{s\_1\_exch}, \xi_{s\_2\_proc}, \xi_{s\_2\_exch} \rangle$ and $\boldsymbol{\beta} \in D^{agg}_{Net}$. Each second-order conditional moment $M_{\mathbf{I}}(\boldsymbol{\beta})$ is identified by a moment vector $\mathbf{I} = \langle I_1, I_2, I_3, I_4, I_5, I_6 \rangle$, where $I_i : 0 \leq i \leq 6$ are positive integers and $I_1 + I_2 + \cdots + I_6 = 2$. According to Lemma 8.3.4, the equation $I_1 + I_2 + \cdots + I_6 = 2$ has $\binom{2+6-1}{6-1} = 42$ solutions. The aggregated state space has 112 states. Therefore, the model gives rise to $42 \times 112 = 4704$ second-order conditional moments. The evolution of these moments is captured by a system of DAE where for each $M_{\mathbf{I}}$ one equation of the format of Eq.(8.34) is constructed.

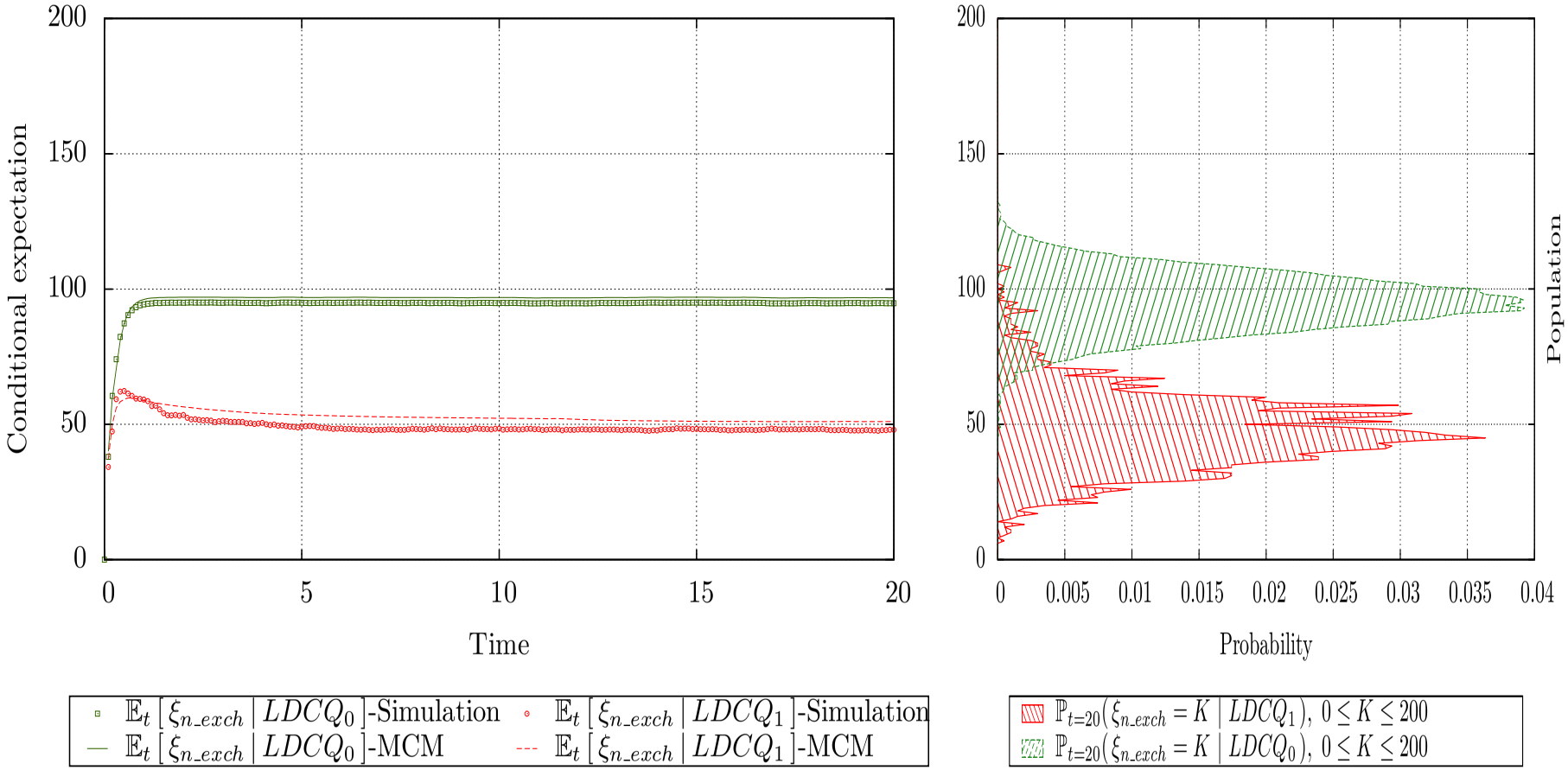
Due to the time constraints of this project, our tool which automatically gener-



(A) Conditional expectations of $\xi_{s\_i\_exch}$, $i \in \{1,2\}$ given that $LDCQ = LDCQ_0$ or $LDCQ = LDCQ_1$.

(B) The distribution of $\xi_{s\_i\_exch}$ in the conditional form at time 20.

Figure 10.18: The evolution of $\mathbb{E}_t[\xi_{s\_i\_exch} \mid LDCQ_k]$, $k = 1,2$, and comparison between $\mathbb{E}_{t=20}[\xi_{s\_i\_exch} \mid LDCQ_k]$ derived by MCM and probability distribution $\mathbb{P}_{t=20}(\xi_{s\_i\_exch} \mid LDCQ_k)$ which is obtained by computationally expensive stochastic simulation.



(A) Conditional expectations of ξ_{n_exch} given $LDCQ_0$ or $LDCQ_1$.

(B) The distribution of ξ_{n_exch} in the conditional form at time 20.

Figure 10.19: The evolution of $\mathbb{E}_t[\xi_{n\_exch} \mid LDCQ_k]$, $k = 1,2$, and comparison between $\mathbb{E}_{t=20}[\xi_{n\_exch} \mid LDCQ_k]$ derived by MCM and the distribution $\mathbb{P}_{t=20}(\xi_{n\_exch} \mid LDCQ_k)$ obtained by computationally expensive stochastic simulation.

ates the DAE system from the model could not be fully developed. Nevertheless, to show the importance of Eq.(8.34) and in particular the usefulness of second-order conditional moments, we still derive and present them using the method of stochastic simulation. We are interested in conditional variances $\mathbb{VAR}_t[\xi_{n\_exch} \,|\, LDCQ_0]$, $\mathbb{VAR}_t[\xi_{n\_exch} \,|\, LDCQ_1]$, $\mathbb{VAR}_t[\xi_{scu\_i\_exch} \,|\, LDCQ_0]$ and $\mathbb{VAR}_t[\xi_{scu\_i\_exch} \,|\, LDCQ_1]$ for $t_{init} \leq t \leq 20$.

For this analysis, we executed 300,000 simulations runs. Generating this number of trajectories is necessary as we then obtain conditional expectations and conditional variances that are stable and accurate[2]. Using the trajectories, the approximate conditional distributions $\mathbb{P}_t(\xi \mid LDCQ_0)$ and $\mathbb{P}_t(\xi \mid LDCQ_1)$ were constructed. From this distribution, we derived the conditional expectations $\mathbb{E}_t\left[\xi^2 \mid LDCQ_0\right]$ and $\mathbb{E}_t\left[\xi^2 \mid LDCQ_1\right]$, and the conditional expectations $\mathbb{E}_t[\xi \mid LDCQ_0]$ and $\mathbb{E}_t[\xi \mid LDCQ_1]$. These are substituted in:

$$\forall\, \xi \in \boldsymbol{\xi}^l,\ k \in \{1,2\}: \quad \mathbb{VAR}[\xi|LDCQ_k] \;=\; \mathbb{E}_t\left[\xi^2 \mid LDCQ_k\right] - \left(\,\mathbb{E}_t[\xi \mid LDCQ_k]\,\right)^2$$

to find the conditional variances $\mathbb{VAR}_t[\xi|LDCQ_k]$. As a final step, we also derived the conditional standard deviations $\sigma_t[\xi|LDCQ_0]$ and $\sigma_t[\xi|LDCQ_1]$, which are the square root of the conditional variances. The results of this analysis are shown in Fig. 10.20.

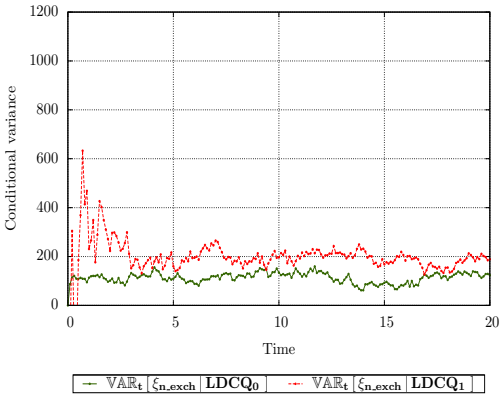
10.4.3.2 Confidence Bounds

Assuming that the distributions $\mathbb{P}_t(\xi \mid LDCQ_0)$ and $\mathbb{P}_t(\xi \mid LDCQ_1)$ are uni-modal and close to being normally distributed, we are able to use the above conditional expectations and standard deviations to calculate the distributions' 95% confidence bounds:

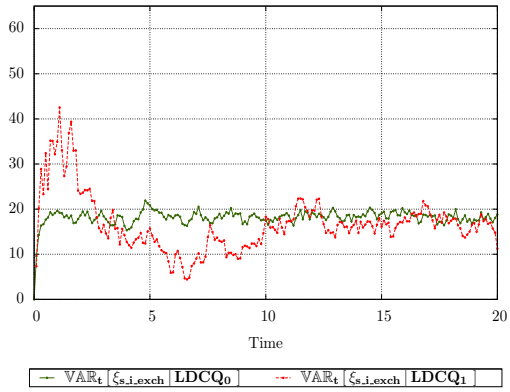
$$\mathbb{P}\Bigg(\; \mathbb{E}_t[\xi \mid LDCQ_0] - 2\sigma_t[\xi|LDCQ_0] \leq \Big(\xi \mid LDCQ_0\Big) \leq \mathbb{E}_t[\xi \mid LDCQ_0] + 2\sigma_t[\xi|LDCQ_0] \;\Bigg) \approx 95\%$$

$$\mathbb{P}\Bigg(\; \mathbb{E}_t[\xi \mid LDCQ_1] - 2\sigma_t[\xi|LDCQ_1] \leq \Big(\xi \mid LDCQ_1\Big) \leq \mathbb{E}_t[\xi \mid LDCQ_1] + 2\sigma_t[\xi|LDCQ_1] \;\Bigg) \approx 95\%$$

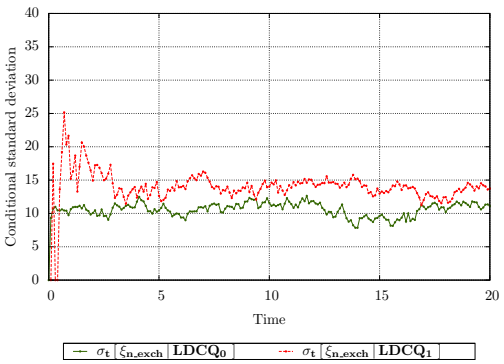
[2] Recall that the calculation of a variance depends on the expectation: $\mathbb{VAR}(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2$.



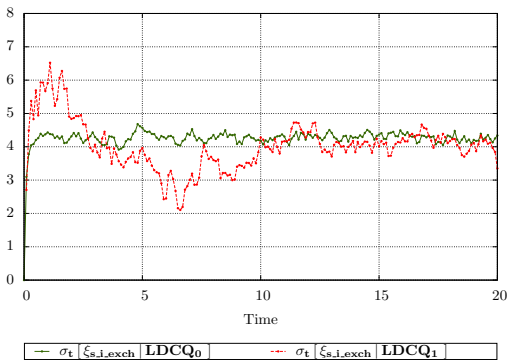
(A) Evolution of $\mathbb{VAR}[\xi_{n\_exch} | LDCQ_0]$ and $\mathbb{VAR}[\xi_{n\_exch} | LDCQ_1]$.



(B) Evolution of $\mathbb{VAR}[\xi_{s\_i\_exch} | LDCQ_0]$ and $\mathbb{VAR}[\xi_{s\_i\_exch} | LDCQ_1]$.



(C) Evolution of $\sigma_t[\xi_{n\_exch} | LDCQ_0]$ and $\sigma_t[\xi_{n\_exch} | LDCQ_1]$.



(D) Evolution of $\sigma_t[\xi_{s\_i\_exch} | LDCQ_0]$ and $\sigma_t[\xi_{s\_i\_exch} | LDCQ_1]$.

Figure 10.20: The result of the analysis of second-order moment performed using the method of stochastic simulation.

The confidence bounds related to the dynamics of near users are shown in Fig. 10.21 and 10.22. In these figures, we are also comparing the confidence bounds at $t = 20$ with the distributions $\mathbb{P}_{t=20}(\xi_{n\_exch} \mid LDCQ_k)$. Similarly, in Fig 10.23 and 10.24 we present the evolution of the bounds calculated for far SCU users and compare them with the distribution $\mathbb{P}_{t=20}(\xi_{s\_i\_exch} \mid LDCQ_k)$. In these graphs, all values shown are derived using stochastic simulation.

10.4.4 Concluding Remarks

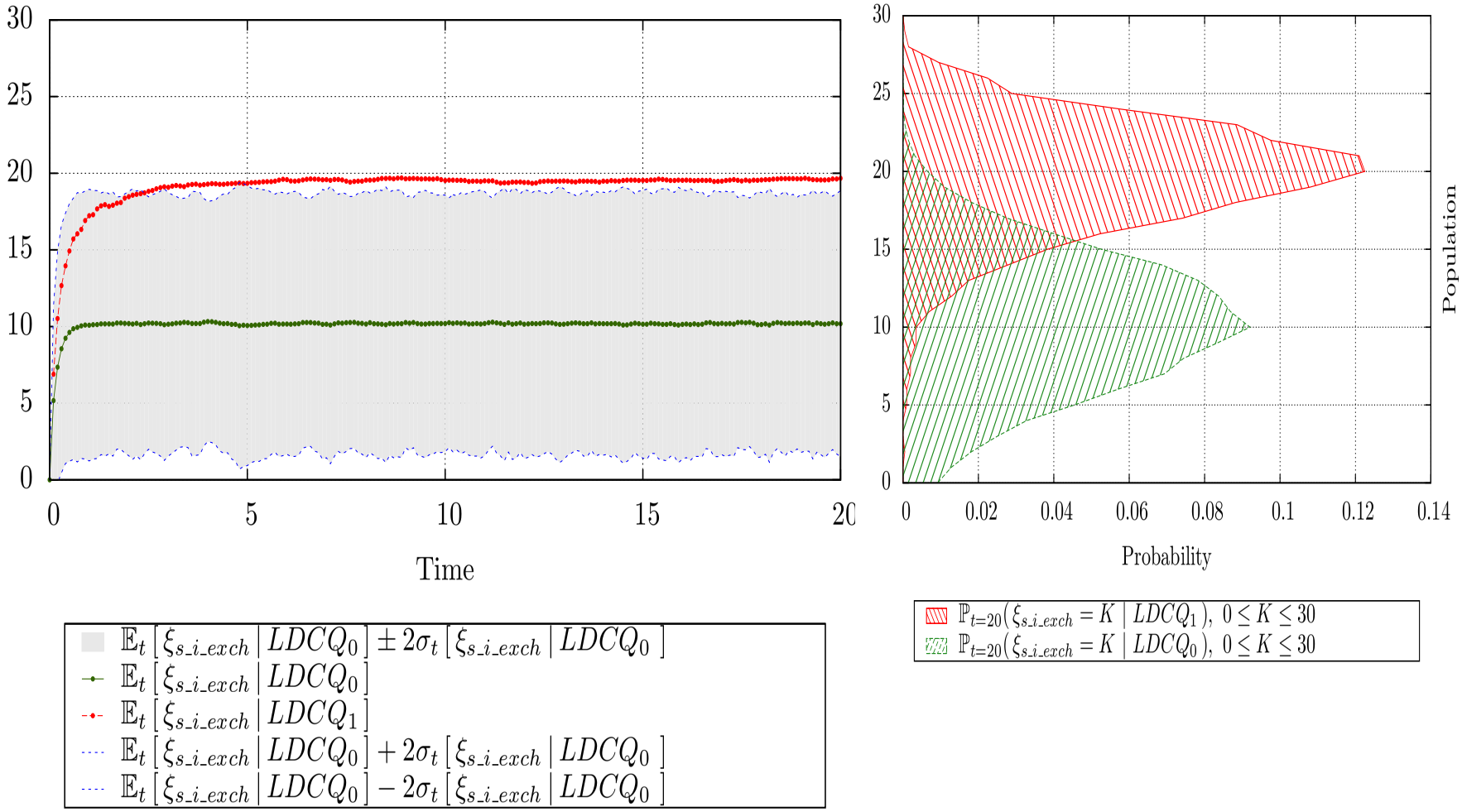
The conditional moments derived in the last two sections capture key aspects of the network's behaviour. The expectations $\mathbb{E}_t[\xi \mid LDCQ_0]$ and $\mathbb{E}_t[\xi \mid LDCQ_1]$ are considerably different, which implies that the system's performance is sensitive to the activation of LDCs. It is possible to make this judgment because in our analysis we have distinguished between the different configurations that the small groups may take and have calculated one expected behaviour for each configuration. Moreover, the combination of the conditional expectations and confidence bounds enable us to obtain a rich representation of the distributions $\mathbb{P}_t(\xi)$, $\xi \in \{\xi_{n\_exch}, \xi_{s\_i\_exch}\}$. For any ξ, we are now capturing the values within its domain where the probability mass is clustered given that the system may perform in any mode $M \in \{M_{nr}, M_{po}\}$. This highlights the fact that the MCM analysis offers a higher degree of faithfulness compared to the fluid flow approximation, where for all state variables one set of moments is derived regardless of the system's distinct operational modes.

10.5 Summary

In this chapter, we presented a case study where we applied the analysis of conditional moments to the model of a wireless network built based on the two-tier architecture.

We briefly reviewed the main features of the architecture and its benefits in terms of interference reduction and energy efficiency. We showed that the system exhibits two modes of operation with distinct performance characteristics. This gave rise to the evaluation requirement that the system's behaviour must be studied in both modes, and also that the impact of mode alternations on users should be captured.

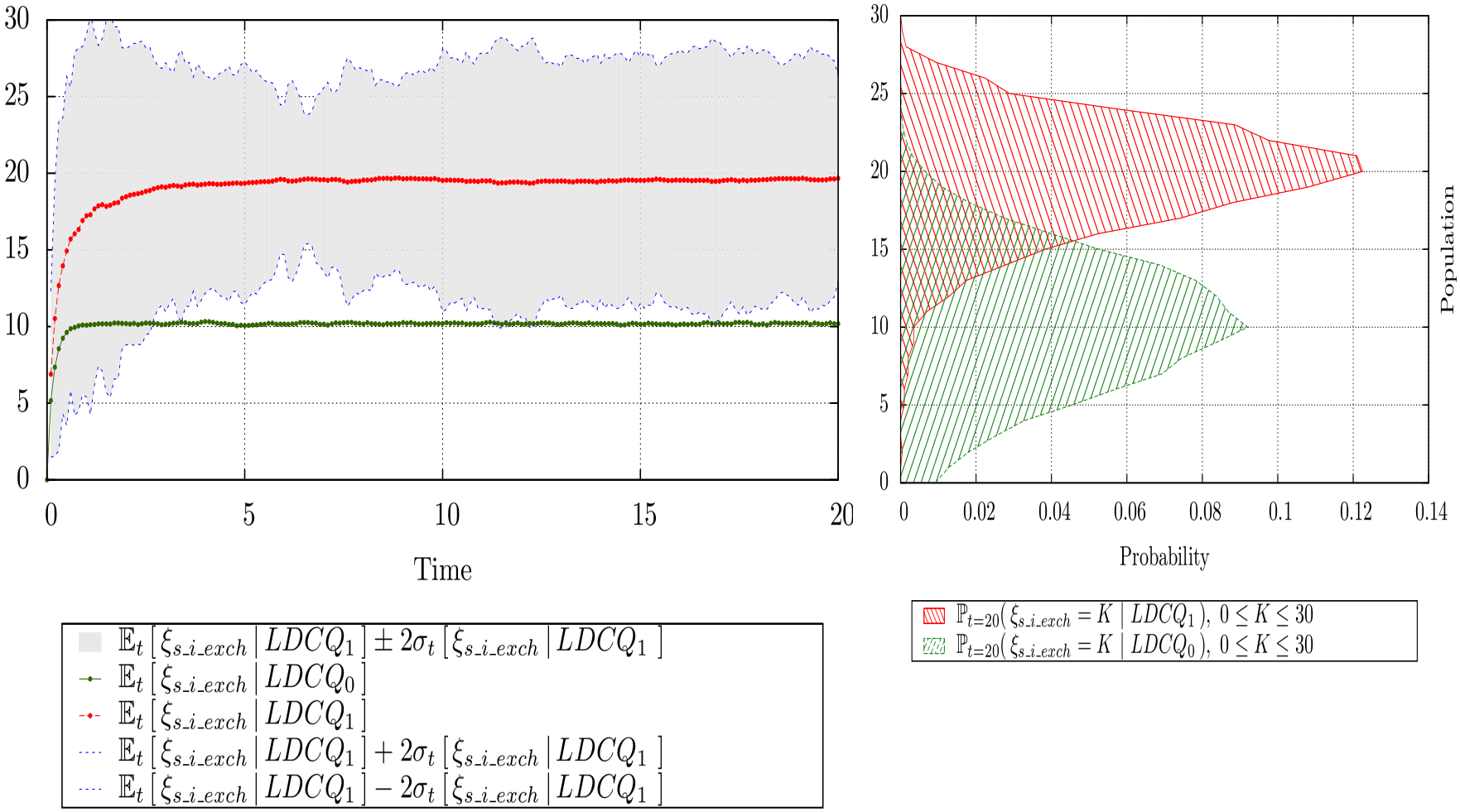
Then we presented our PEPA model. The compositionality of PEPA aided the con-



(A) The conditional expectation $\mathbb{E}_t[\xi_{s\_i\_exch} \mid LDCQ_0]$ and the 95% confidence bounds around.

(B) The distribution of $\xi_{s\_i\_exch}$ in the conditional form at time 20.

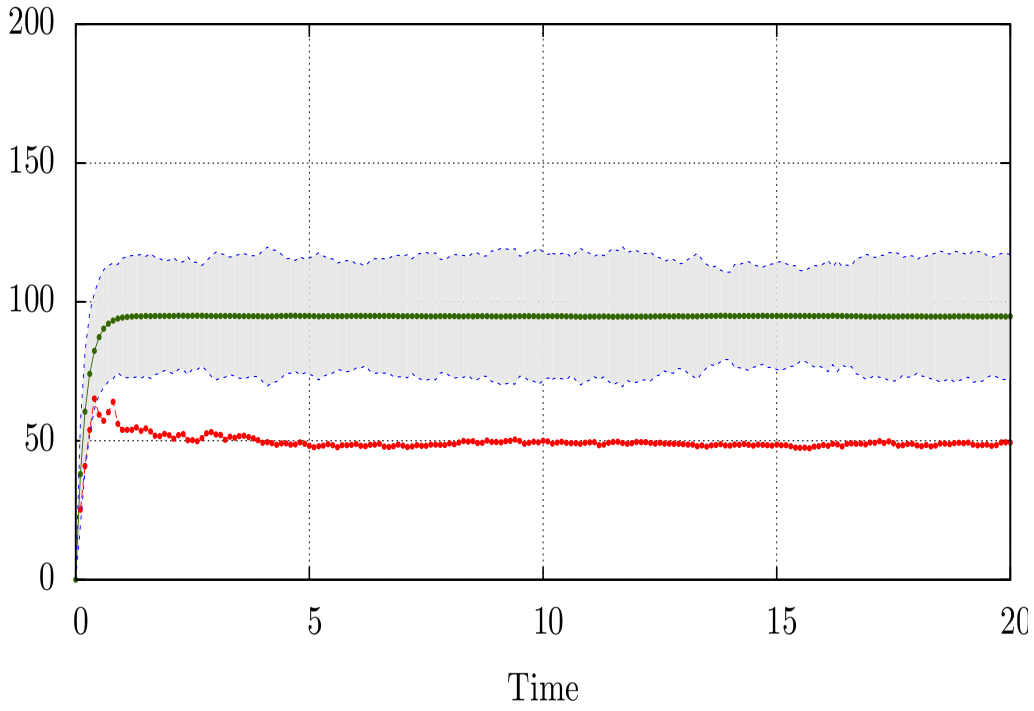
Figure 10.21: Comparison between conditional expectation $\mathbb{E}_t[\xi_{s\_i\_exch} \mid LDCQ_0]$ and 95% confidence bounds and the distribution $\mathbb{P}_t(\xi_{s\_i\_exch} \mid LDCQ_0)$.



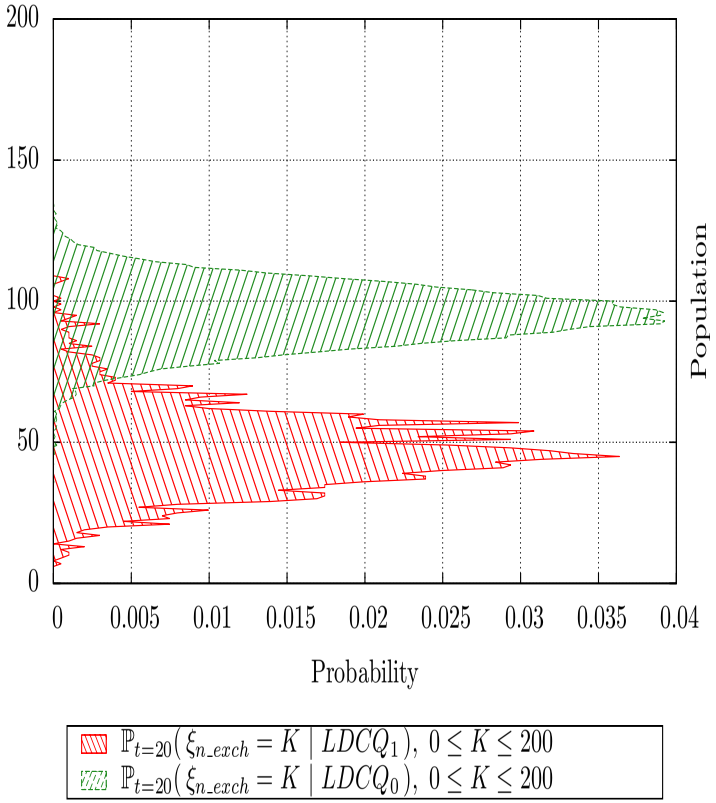
(A) The conditional expectation $\mathbb{E}_t[\xi_{s\_i\_exch} \mid LDCQ_1]$ and the 95% confidence bounds around.

(B) The distribution of $\xi_{s\_i\_exch}$ in the conditional form at time 20.

Figure 10.22: Comparison between conditional expectation $\mathbb{E}_t[\xi_{s\_i\_exch} \mid LDCQ_1]$ and 95% confidence bounds and the distribution $\mathbb{P}_t(\xi_{s\_i\_exch} \mid LDCQ_1)$.

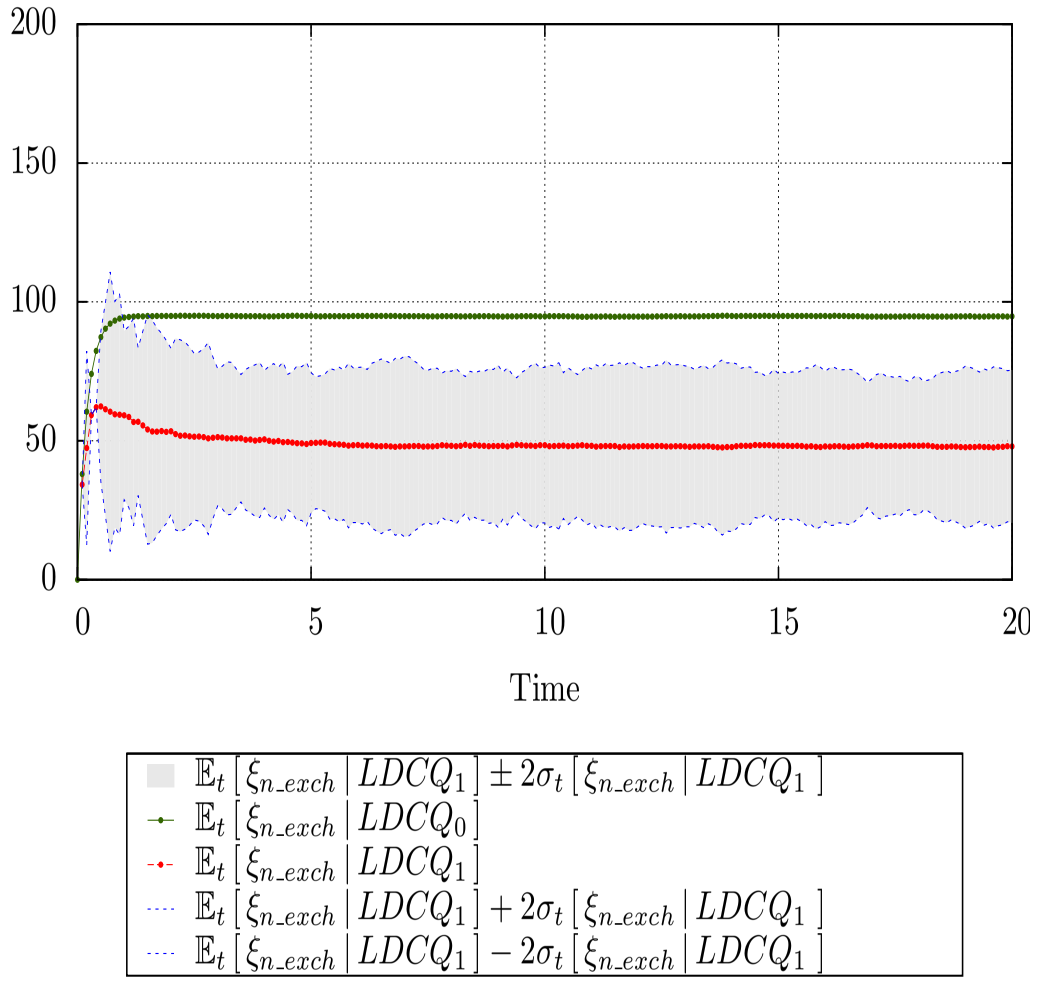


(A) The conditional expectation $\mathbb{E}_t[\xi_{n\_exch} \mid LDCQ_0]$ and the 95% confidence bounds around.

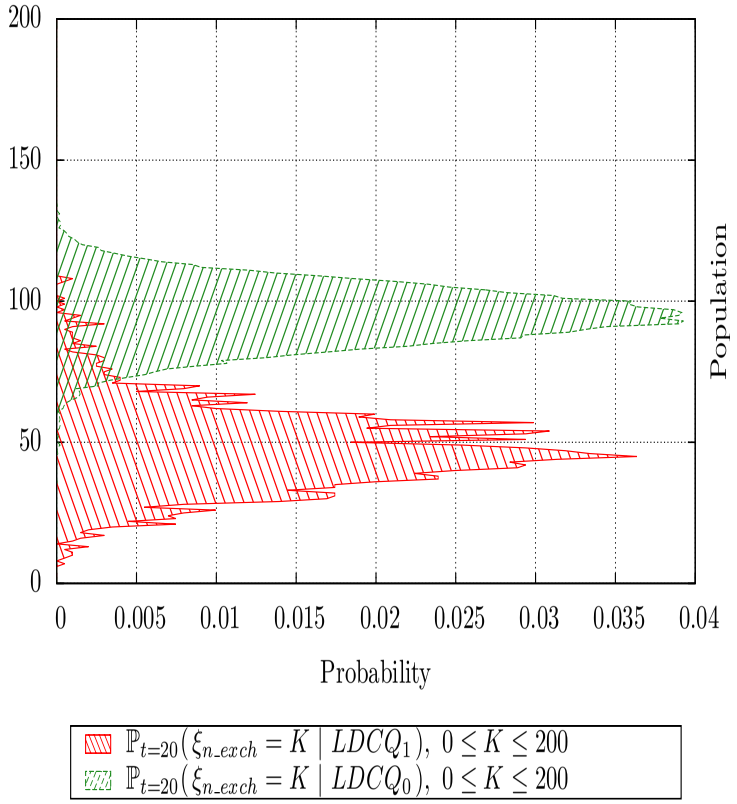


(B) The distribution of ξ_{n_exch} in the conditional form at time 20.

Figure 10.23: Comparison between conditional expectation $\mathbb{E}_t[\xi_{n\_exch} \mid LDCQ_0]$ and 95% confidence bounds and the distribution $\mathbb{P}_t(\xi_{n\_exch} \mid LDCQ_0)$.



(A) The conditional expectation $\mathbb{E}_t[\xi_{n\_exch} \mid LDCQ_1]$ and the 95% confidence bounds around.



(B) The distribution of ξ_{n_exch} in the conditional form at time 20.

Figure 10.24: Comparison between conditional expectation $\mathbb{E}_t[\xi_{n\_exch} \mid LDCQ_1]$ and 95% confidence bounds and the distribution $\mathbb{P}_t(\xi_{n\_exch} \mid LDCQ_1)$.

struction of the model from the bottom up and helped us to cope with the complexity of the components' interactions. We categorised the model's groups as being either small or large. The small groups were related to the components within the network infrastructure and users that appeared in small populations. The large groups were related to near users and those strongly coupled to the femto-cells who appeared in large populations.

Next, we analysed the model. First, we applied our aggregation method. We could answer key performance questions regarding the operation of the network (eg. the probability of having active LDCs within the network) by analysing the aggregated state space. Second, we applied the method of conditional expectations. This enabled us to study the expected behaviour of users in large groups in each of the system's operational modes. Third, we applied the method of higher-order conditional moments to obtain the conditional variances. This means that for each operational mode, in addition to the users' expected evolution, we also have a measure of the variability of their stochastic behaviour around the expectation. We showed that the conditional expectations and variances can be combined to provide conditional confidence bounds. These are particularly useful for capturing the shape of the marginal probability distribution related to the large groups.

Chapter 11

Conclusions

11.1 Thesis Summary

This thesis has explored the performance evaluation of the class of large-scale resource-bound computer and communication systems. Conceptually, LSRB systems are those in which components are categorised into resources and resource users, with dynamics where large populations of resource users are handled by a significantly smaller population of resources. Due to the conceptual generality, many realistic systems can be considered to fit in this class.

The performance evaluation methodology of interest is quantitative modelling. We argued that PEPA is an appealing option for model construction. In particular, the compositionality feature enables us to deal with systems that have many different types of components with complex interactions. LSRB systems are mapped to LSRB PEPA models.

We showed that the method used for the analysis of LSRB models must respect two requirements: efficiency and faithfulness. Efficiency means that the problem of state space explosion is avoided and the solution is obtained in a timely manner. Faithfulness means that the method's output should reveal key aspects of a system's stochastic behaviour, covering its performance within both optimal and sub-optimal situations. Although the PEPA framework is a powerful tool for the construction of LSRB models, we showed that none of the existing analysis methods, i.e. exact Markovian analysis, stochastic simulation and fluid flow analysis, simultaneously support the above requirements. Thus, we developed four new analysis techniques that provide a more

robust treatment for LSRB PEPA models.

In Chapter 4 we presented an aggregation method. The method allows us to extract the behaviour of a system's resources from the complete model and understand how those components behave, whilst abstracting away from their interactions with resource users. The aggregation is directly applied to the model and gives us the aggregated state space, which can be efficiently analysed to obtain a probability distribution over the configurations that the system's resources experience. The behaviour of resources is obtained in full detail, enabling further investigations such as detecting the resources' optimal and sub-otimal modes of operations.

In Chapter 5 we presented an algorithm useful for LSRB models that have dynamics happening on distinct slow / fast time scales. For a given model, our algorithm efficiently detects which actions belong to the slow time scale and which ones occur on the fast time scale. Our algorithm is embedded within a well-known time-scale decomposition solution method which analyses the models by taking as input the slow / fast categorisation that our algorithm provides.

In Chapter 6 we presented the method of conditional expectations. Assuming that the configurations of the resources are captured by the aggregated state space, here we derive the expected behaviour of resource users for each of the system's significant modes of operation. The motivation is to avoid a naive and fallible performance evaluation where a LSRB system's performance is only evaluated in terms of its average behaviour whilst abstracting from intrinsic noise. The conditional expectations enable us to take into account that resources usually have different operational modes and the system's behaviour needs to be studied for each of those separately. The method derives the conditional expectations by solving a system of differential equations directly derived from the model. The number of equations is independent of the population parameters of resource users. This enables the method to handle models with massively large user populations.

In Chapter 7 we applied the method of conditional expectations to a simple client-server model. The system we considered had a multi-modal behaviour and depending on the number of servers currently broken, the clients experienced different service throughputs from the servers. Using the conditional expectation, we derived the expected behaviour of the users for each of the operational modes. The impact of the server break-downs could be quantitatively captured.

In Chapter 8 we presented the method of higher-order conditional moments. The method enables us to build a richer representation of the behaviour of resource users; in addition to the conditional expectation, for each configuration of the resources we also derive measures such as conditional variance and conditional skewness. We showed that by combining the aggregation, conditional expectations and higher-order conditional moments, we can obtain faithful representations of the model's underlying probability distributions and capture important phenomena such as multi-modality, which could not be captured by the methods previously available in the PEPA framework. In Chapter 9, we showed the usefulness of higher-order moments in the case of our client-server example. Here, the moments showed us the variability of the users' behaviour in each of the servers' operational modes.

We think it is worth noting that our analysis methods support transparency. This means that they are integrated into the PEPA language, and their application requires no specific insight into the tedious underlying algebra. Thus, domain experts of various fields can readily take advantage of the analysis methods once they create their models in PEPA. This was particularly shown in the case study of Chapter 10, where we considered the evaluation of a two-tier wireless network. Once the model was ready, the aggregation and the method of conditional expectation, could be easily applied. In our future work, we need to develop a more robust version of our software so that other types of analyses are also included.

11.2 Future Work

Our future work is divided into two areas. The first is related to the theoretical work that can improve our newly devised methods. The second is related to introducing our methods to practical applications which can greatly benefit from using them.

11.2.1 Theoretical Improvements

Accuracy of the Conditional Moments. Our derivation of the conditional moments relied upon a number of approximation steps. One important one was that the expectation of the minimum of two random variables is replaced by the minimum of their expectations (Page 120). The approximation is a classic step in moment closure techniques. However, depending on the context it may actually have poor quality. In [51],

Guenther et al. proposed a heuristic that checks the quality of this approximation for fluid flow analysis by considering the evolution of state variables. When the quality is poor, corrections are introduced using higher-order moments. We partly verified their heuristic in [79]. We think that a similar accuracy detection and correction algorithm can be developed for the method of conditional moments. Developing the algorithm requires applying the MCM within some more experiments and carefully examining the factors that give rise to potential errors.

Relaxing the Restrictions of Aggregation. The analysis methods we devised rely heavily on lumpability of an input model's state space with respect to its sub-chains. In turn, the lumpability depends on a restriction imposed on the actions shared between the model's small and large groups; for such actions, the instances within large groups (resource users) must be passive and the cooperation rate must be determined by instances within small groups (resources). This way, state variables related to small groups form the model's independent variables; those related to large groups are completely dependent; and consequently, the large groups cannot affect the small ones. This restriction limits the scope of models that are amenable to our analysis methods. Nevertheless, we believe that it can be relaxed easily. Our preliminary work on Chapman-Kolmogorov equations suggests that when the restriction is relaxed, we can still derive a closed set of equations in which the impact of large groups on small ones is captured. In such cases, the rate of a shared action will be a function of the state of resources and the conditional expectations of the large groups. By introducing this extension, there would be an impact from large groups to small ones and we would be able to accommodate a wider range of LSRB models.

Reducing the Number of DAEs. When applying the MCM, for each single state of the aggregated state space we derive one set of equations concerning the conditional expectation and higher-order moments of large groups given that state. This derivation is usually too detailed, especially when the number of aggregated states is large. We are usually interested in the conditional expectations, not related to single aggregated states, but to the system's *significant modes of operation*, which are often defined as isolated collections of the aggregated states. To derive the conditional moments of a mode of operation, we form a weighted sum of the conditional moments associated with the states within. Work needs to be done to investigate if the equations related to the operational modes could be directly derived without moving to the lower level of the aggregated states. Intuitively, we might be able to achieve this by considering

the probability fluxes between the modes. In this way the application of MCM would generate fewer equations and become more scalable. This line of work originates from [95] where Tschaikowski and Tribastone proposed a method for aggregating differential equations generated by fluid flow analysis.

11.2.2 Practical Directions

Software Tool. We developed some exploratory code for checking the validity of our analysis methods and in particular, the equations related to the conditional expectations. The MCM equations have complex structures and the value of the method is achieved by a fully-fledged tool supporting all aspects of running the method (deriving the equations and their initial values). One of our topmost priorities for future work is therefore to enhance our tool and possibly integrate it with PEPA Eclipse plug-in [93]. Our tool should be interactive and provide an easy way for specifying the categorisation of groups into small and large, and the system's operational modes of interest.

Risk-Aware Capacity Planning. One interesting venue where we can explore the use of the MCM is capacity planning (CP). In a CP scenario, we aim to solve an optimisation problem: given the load from the users, specification of resources and a number of cost functions, what populations of resources are needed to meet the demand of the users? In [101] an automated CP method was proposed where the input is a PEPA model. The analysis method embedded in the tool is fluid flow analysis; for each candidate answer its suitability is verified by checking if the expected behaviour of the users is satisfactory, where the expected behaviour is derived using fluid flow equations. We think that this tool can be fruitfully extended by incorporating the conditional expectations and higher-order moments. These methods show the users' behaviour across all operational modes, including the risky or critical ones. Thus, embedding the MCM within CP would allow us to check each answer with respect to the behaviour that it creates across all operational modes. The result is a more conservative CP but one that may also run faster (as the state space search will be more restricted) and guarantee system robustness under optimal and sub-optimal situations. The validation of these advantages is the subject of our future work.

Improved Inference. The models we build in practice contain rate parameters that are supplied by the modeller. It is often the case that the knowledge about the rates is incomplete, and they are estimated using the observations made on the real sys-

tem. One of the methods useful for parameter estimation of stochastic processes is the moment closure technique [74]. Given a set of observations and a prior distribution over the parameters (our initial / prior belief about the probability of different parameter values before we see any actual data / observations), the estimation outputs a posterior distribution that captures how well different parameter values can replicate the observations. Since the MCM equations can provide a richer representation of the model's underlying dynamics, it is interesting to check if using this method will result in more accurate and realistic parameter estimation. Perhaps the posterior distribution will become more precise, as the impact of the randomness within small populations is explicitly preserved. One piece of related work in this direction is [65]. We will leave the validation of this idea as a future work.

Bibliography

[1] HP Capacity Advisor Consolidation Software. Product specification available at http://h18004.www1.hp.com/products/servers/management/capad/index.html.

[2] D. Abbaldo, M. Gribaudo, and D. Manini. Evaluation of different scheduling policies in IaaS applications by mean field analysis. In *Performance Evaluation Methodologies and Tools (VALUETOOLS), 2012 6th International Conference on*, pages 307–316, Oct 2012.

[3] H. Alla and R. David. Continuous and hybrid Petri nets. *Journal of Circuits, Systems and Computers*, 08(01):159–188, 1998.

[4] J. Allspaw. *The Art of Capacity Planning: Scaling Web Resources*. O'Reilly Media, Inc., 2008.

[5] A. Ando and F. M. Fisher. Near-decomposability, partition and aggregation, and the relevance of stability discussions. *International Economic Review*, 4(1):53–67, 1963.

[6] J.G. Andrews, H. Claussen, M. Dohler, S. Rangan, and M.C. Reed. Femtocells: Past, present, and future. *Selected Areas in Communications, IEEE Journal on*, 30(3):497–508, 2012.

[7] A. Angius, G. Balbo, M. Beccuti, E. Bibbona, A. Horvath, and Sirovich R. Approximate analysis of biological systems by hybrid switching jump diffusion. To appear in Theoretical Computer Science, 2015.

[8] R. Bakhshi, L. Cloth, W. Fokkink, and B. Haverkort. Mean-field analysis for the evaluation of gossip protocols. In *Quantitative Evaluation of Systems, 2009. QEST '09. Sixth International Conference on the*, pages 247–256, Sept 2009.

[9] G. Balbo. Introduction to generalized stochastic Petri nets. In Marco Bernardo and Jane Hillston, editors, *Formal Methods for Performance Evaluation*, volume 4486 of *Lecture Notes in Computer Science*, pages 83–131. Springer Berlin Heidelberg, 2007.

[10] AD Barbour. Density dependent Markov population processes. In *Biological growth and spread*, pages 36–49. Springer, 1980.

[11] M. Benam and J. I Le Boudec. A class of mean field interaction models for computer and communication systems. *Performance Evaluation*, 65(1112):823 – 838, 2008. Performance Evaluation Methodologies and Tools: Selected Papers from ValueTools 2007.

[12] P. Berkhin. A survey of clustering data mining techniques. In Jacob Kogan, Charles Nicholas, and Marc Teboulle, editors, *Grouping Multidimensional Data*, pages 25–71. Springer Berlin Heidelberg, 2006.

[13] M. Bernardo and R. Gorrieri. A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time”. *Theor. Comput. Sci.*, 254(1-2):691–694, March 2001.

[14] Z. Bharucha, H. Haas, A. Saul, and G. Auer. Throughput enhancement through femto-cell deployment. *European Transactions on Telecommunications*, 21(5):469–477, 2010.

[15] C. Bishop. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[16] A. Bobbio, M. Gribaudo, and M. Telek. Analysis of large scale interacting systems by mean field method. In *Quantitative Evaluation of Systems, 2008. QEST '08. Fifth International Conference on*, pages 215–224, Sept 2008.

[17] G. Bolch, S. Greiner, H. De Meer, and K. Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. Wiley-Interscience, New York, NY, USA, 1998.

[18] L. Bortolussi. Hybrid limits of continuous time Markov chains. In *Quantitative Evaluation of Systems (QEST), 2011 Eighth International Conference on*, pages 3–12, Sept 2011.

[19] L. Bortolussi, V. Galpin, J. Hillston, and M. Tribastone. Hybrid semantics for PEPA. In *Seventh International Conference on Quantitative Evaluation of Systems (QEST), 2010*, pages 181–190, Sept 2010.

[20] L. Bortolussi and A. Policriti. The importance of being (a little bit) discrete. *Electronic Notes in Theoretical Computer Science*, 229(1):75 – 92, 2009. Proceedings of the Second Workshop From Biology to Concurrency and Back (FBTC 2008).

[21] L. Bortolussi and A. Policriti. Hybrid dynamics of stochastic programs. *Theoretical Computer Science*, 411(20):2052 – 2077, 2010.

[22] J. Bradley, S. Gilmore, and J. Hillston. Analysing distributed internet worm attacks using continuous state-space approximation of process algebra models. *Journal of Computer and System Sciences*, 74(6):1013 – 1032, 2008.

[23] J. Bradley, M. Guenther, R. Hayden, and A. Stefanek. GPA - A multiformalism, multisolution approach to efficient analysis of large scale population models. In *Theory and Application of Multi-Formalism Modeling*, pages 144–169. IGI Global, September 2013.

[24] P. Brown, A. Hindmarsh, and L. Petzold. Consistent initial condition calculation for differential-algebraic systems. *SIAM J. Sci. Comput.*, 19(5):1495–1512, September 1998.

[25] P. Buchholz. Exact and ordinary lumpability in finite Markov chains. *Journal of Applied Probability*, 31(1):59–75, 1994.

[26] D. Calin, H. Claussen, and H. Uzunalioglu. On femto deployment architectures and macrocell offloading benefits in joint macro-femto deployments. *Communications Magazine, IEEE*, 48(1):26–32, January 2010.

[27] Y. Cao, D. Gillespie, and D. Petzold. Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems. *Journal of Computational Physics*, 206(2):395 – 411, 2005.

[28] Y. Cao, D. Gillespie, and L. Petzold. Adaptive explicit-implicit tau-leaping method with automatic tau selection. *The Journal of Chemical Physics*, 126(22):120–146, 2007.

[29] G. Casale and P. Harrison. AutoCAT: Automated product-form solution of stochastic models. In G. Latouche, V. Ramaswami, J. Sethuraman, K. Sigman, M. Squillante, and D. Yao, editors, *Matrix-Analytic Methods in Stochastic Models*, volume 27 of *Springer Proceedings in Mathematics and statistics*, pages 57–85. Springer New York, 2013.

[30] V. Chandrasekhar, J.G. Andrews, and Alan Gatherer. Femtocell networks: a survey. *Communications Magazine, IEEE*, 46(9):59–67, September 2008.

[31] M. Cirillo. Humanizing calculus. *Mathematics teacher*, 101(1):23–27, 2007.

[32] A.E. Conway and N.D. Georganas. Decomposition and aggregation by class in closed queueing networks. *Software Engineering*, pages 1025–1040, 1986.

[33] P.J. Courtois. *Decomposability: queueing and computer system applications*. ACM monograph series. Academic Press, 1977.

[34] W. Day and H. Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1(1):7–24, 1984.

[35] J. Ding. *Structural and fluid analysis of large scale PEPA models, with applications to content adaptation system*. PhD thesis, School of Informatics, The University of Edinburgh, 2010.

[36] A. Duncan, S. Liao, T. Vejchodsk, R. Erban, and R. Grima. Noise-induced multistability in chemical systems: Discrete versus continuum modeling. *Phys. Rev. E*, 91:042111, Apr 2015.

[37] C. Dutheillet and S. Haddad. Aggregation of states in colored stochastic Petri nets: application to a multiprocessor architecture. In *Petri Nets and Performance Models, 1989. PNPM89., Proceedings of the Third International Workshop on*, pages 40–49, Dec 1989.

[38] S. Engblom. Computing the moments of high dimensional solutions of the master equation. *Applied Mathematics and Computation*, 180(2):498 – 515, 2006.

[39] P. Feiler, R. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, L. Northrop, D. Schmidt, K. Sullivan, and K. Wallnau. *Ultra-Large-Scale Systems – The Software Challenge of the Future*. Pittsburgh, PA 15213-3890, USA, June 2006.

[40] W. Feller. *An introduction to probability theory and its applications*. Wiley series in probability and mathematical statistics: Probability and mathematical statistics. Wiley, 1971.

[41] J. Fourneau, L. Kloul, and F. Valois. Performance modelling of hierarchical cellular networks using PEPA. *Performance Evaluation*, 50(2):83–99, 2002.

[42] V. Galpin, L. Bortolussi, and J. Hillston. HYPE: Hybrid modelling by composition of flows. *Formal Aspects of Computing*, 25(4):503–541, 2013.

[43] I. Giannoulakis, K. Kontovasilis, and N. Mitrou. Performance, dimensioning and interference tradeoffs for two-tier wireless networks. In *IEEE PIMRC 2013*, 2013.

[44] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[45] Daniel T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A: Statistical Mechanics and its Applications*, 188(13):404 – 425, 1992.

[46] Daniel T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 115(4):1716–1733, 2001.

[47] S. Gilmore, J. Hillston, and M. Ribaudo. An efficient algorithm for aggregating PEPA models. *Software Engineering, IEEE Transactions on*, 27(5):449–464, May 2001.

[48] N. Gotz, U. Herzog, and M. Rettelbach. Multiprocessor and distributed system design: The integration of functional specification and performance analysis using stochastic process algebras. In *Performance Evaluation of Computer and Communication Systems, Joint Tutorial Papers of Performance '93 and Sigmetrics '93*, pages 121–146, London, UK, 1993. Springer-Verlag.

[49] R. Grima. Investigating the robustness of the classical enzyme kinetic equations in small intracellular compartments. *BMC Systems Biology*, 3(1):101–113, 2009.

[50] G. Grimmett and D. Stirzaker. *Probability and random processes*, volume 2. Clarendon Press Oxford, 1992.

[51] M. Guenther, A. Stefanek, and J. Bradley. Moment closures for performance models with highly non-linear rates. In Mirco Tribastone and Stephen Gilmore, editors, *Computer Performance Engineering*, volume 7587 of *Lecture Notes in Computer Science*, pages 32–47. Springer Berlin Heidelberg, 2013.

[52] M.C. Guenther and J. Bradley. Mean-field performance analysis of a hazard detection wireless sensor network. In *Performance Evaluation Methodologies and Tools (VALUETOOLS), 2012 6th International Conference on*, pages 220–221, Oct 2012.

[53] J. Hasenauer, V. Wolf, A. Kazeroonian, and F.J. Theis. Method of conditional moments (MCM) for the chemical master equation. *Journal of Mathematical Biology*, 69(3):687–735, 2014.

[54] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer, 2009.

[55] R. Hayden. *Scalable Performance Analysis of Massively Parallel Stochastic Systems*. PhD thesis, Imperial College London, April 2011.

[56] R. A. Hayden and J. T. Bradley. A fluid analysis framework for a Markovian process algebra. *Theor. Comput. Sci.*, 411(22-24):2260–2297, May 2010.

[57] T. Henzinger. The theory of hybrid automata. In *Logic in Computer Science, 1996. LICS '96. Proceedings., Eleventh Annual IEEE Symposium on*, pages 278–292, Jul 1996.

[58] J. Hillston. *A compositional approach to performance modelling*. CUP, New York, NY, USA, 1996.

[59] J. Hillston. Fluid flow approximation of PEPA models. In *2nd International Conference on Quantitative Evaluation of Systems (QEST)*, pages 33–42, 2005.

[60] J. Hillston, A. Marin, S. Rossi, and C. Piazza. Contextual lumpability. In *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools*, ValueTools '13, pages 194–203, Brussels, Belgium, 2013. ICST.

[61] J. Hillston and V. Mertsiotakis. A simple time scale decomposition technique for stochastic process algebras. *The Computer Journal*, 38(7):566–577, 1995.

[62] C. A. R. Hoare. Communicating sequential processes. *Commun. ACM*, 21(8):666–677, August 1978.

[63] J. Kemeny and J. Snell. *Finite Markov chains*, volume 356. van Nostrand Princeton, NJ, 1960.

[64] J.F.C. Kingman. The first Erlang century and the next. *Queueing Systems*, 63(1-4):3–12, 2009.

[65] H. Koeppl, C. Zechner, A. Ganguly, S. Pelet, and M. Peter. Accounting for extrinsic variability in the estimation of stochastic rate constants. *International Journal of Robust and Nonlinear Control*, 22(10):1103–1119, 2012.

[66] T. Kurtz. Solutions of ordinary differential equations as limits of pure jump Markov processes. *Journal of Applied Probability*, 7(1):49–58, 1970.

[67] T. Kurtz. Strong approximation theorems for density dependent Markov chains. *Stochastic Processes and Their Applications*, 6(3):223–240, 1978.

[68] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: verification of probabilistic real-time systems. In *23rd Int. Conf.*, CAV'11, pages 585–591. Springer-Verlag, 2011.

[69] B. Lauwens and B. Scheers. Stochastic hybrid simulation with applications to queuing networks. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, SIMUTools '10, pages 19–32, Brussels, Belgium, 2010. ICST.

[70] C. Lee, K. Kim, and P. Kim. A moment closure method for stochastic reaction networks. *The Journal of chemical physics*, 130(13):104–137, 2009.

[71] J. Little. A proof for the queuing formula: $L = \lambda W$. *Operations Research*, 9(3):383–387, 1961.

[72] S. Menz, J. Latorre, C. Schtte, and W. Huisinga. Hybrid stochastic–deterministic solution of the chemical master equation. *Multiscale Modeling and Simulation*, 10(4):1232–1262, 2012.

[73] V. Mertsiotakis. *Approximate Analysis Methods For Stochastic Process Algebra*. PhD thesis, Erlangen, Germany, November 1998.

[74] P. Milner, C. Gillespie, and D. Wilkinson. Moment closure based parameter inference of stochastic kinetic models. *Statistics and Computing*, 23(2):287–295, 2013.

[75] R. Milner. *A Calculus of Communicating Systems*. Springer-Verlag New York, Inc., Seracus, NJ, USA, 1982.

[76] J. Paulsson. Models of stochastic gene expression. *Physics of Life Reviews*, 2(2):157 – 175, 2005.

[77] A. Pourranjbar. Thesis online suppliments, http://tinyurl.com/ooj5uux, 2014.

[78] A. Pourranjbar and J. Hillston. An aggregation technique for large-scale PEPA models with non-uniform populations. In *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools*, ValueTools '13, pages 20–29, Brussels, Belgium, 2013. ICST.

[79] A. Pourranjbar, J. Hillston, and L. Bortolussi. Don't just go with the flow: Cautionary tales of fluid flow approximation. In Mirco Tribastone and Stephen Gilmore, editors, *Computer Performance Engineering*, volume 7587 of *Lecture Notes in Computer Science*, pages 156–171. Springer Berlin Heidelberg, 2013.

[80] W. Reisig and G. Rozenberg, editors. *Lectures on Petri Nets II: Applications, Advances in Petri Nets, the Volumes Are Based on the Advanced Course on Petri Nets*, London, UK, 1998. Springer-Verlag.

[81] M. Ribaudo. On the aggregation techniques in stochastic Petri nets and stochastic process algebras. *The Computer Journal*, 38(7):600–611, 1995.

[82] D. Romik. Stirling's approximation for $n!$: the ultimate short proof? *Amer. Math. Monthly*, 107(6):556–557, 2000.

[83] T. Rowland and E. Weisstein. Lipschitz Functin. From MathWorld - A Wolfram Web Resource. http://mathworld.wolfram.com/LipschitzFunction.html.

[84] H. Salis and Y. Kaznessis. Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. *The Journal of Chemical Physics*, 122(5):91–125, 2005.

[85] D. Schnoerr, G. Sanguinetti, and R. Grima. Validity conditions for moment closure approximations in stochastic chemical kinetics. *The Journal of Chemical Physics*, 141(8):56–93, 2014.

[86] B. Schroeder and M. Harchol-Balter. Web servers under overload: How scheduling can help. *ACM Trans. Internet Technol.*, 6(1):20–52, February 2006.

[87] H. A. Simon and A. Ando. Aggregation of variables in dynamic systems. *Econometrica*, 29(2):111–138, 1961.

[88] I. Sommerville, D. Cliff, R. Calinescu, J. Keen, T. Kelly, M. Kwiatkowska, J. Mcdermid, and R. Paige. Large-scale complex IT systems. *Commun. ACM*, 55(7):71–77, July 2012.

[89] Anton Stefanek. *A high-level framework for efficient computation of performance-energy trade-offs in Markov population models*. PhD thesis, Imperial College London, January 2014.

[90] W. Stewart. *Introduction to the numerical solution of Markov chains*, volume 41. Princeton University Press, Princeton, 1994.

[91] N.N. Taleb. *The Black Swan: Second Edition: The Impact of the Highly Improbable Fragility"*. Incerto. Random House Publishing Group, 2010.

[92] P. Thomas, N. Popovi, and R. Grima. Phenotypic switching in gene regulatory networks. *Proceedings of the National Academy of Sciences*, 111(19):6994–6999, 2014.

[93] M. Tribastone, A. Duguid, and S. Gilmore. The PEPA Eclipse plugin. *SIGMETRICS Perform. Eval. Rev.*, 36(4):28–33, March 2009.

[94] M. Tribastone, S. Gilmore, and J. Hillston. Scalable differential analysis of process algebra models. *Software Engineering, IEEE Transactions on*, 38(1):205–219, 2012.

[95] M. Tschaikowski and T. Tribastone. Exact fluid lumpability in Markovian process algebra. *Theoretical Computer Science*, 538(0):140 – 166, 2014. Special Issue on Quantitative Aspects of Programming Languages and Systems (2011-12).

[96] N.G. Van Kampen. *Stochastic Processes in Physics and Chemistry*. North-Holland Personal Library. Elsevier Science, 2011.

[97] Zuberek W. Timed Petri nets definitions, properties, and applications. *Microelectronics Reliability*, 31(4):627 – 644, 1991.

[98] E.W.J. Wallace, D. Gillespie, K. Sanft, and L. Petzold. Linear noise approximation is valid over limited times for any chemical system that is sufficiently large. *Systems Biology, IET*, 6(4):102–115, August 2012.

[99] E. Weisstein. Taylor Series. From MathWorld - A Wolfram Web Resource. http://mathworld.wolfram.com/TaylorSeries.html.

[100] D.J. Wilkinson. *Stochastic Modelling for Systems Biology, Second Edition*. Chapman & Hall/CRC Mathematical and Computational Biology. Taylor & Francis, 2011.

[101] C. Williams and J. Hillston. Automated capacity planning for PEPA models. In Andrs Horvth and Katinka Wolter, editors, *Computer Performance Engineering*, volume 8721 of *Lecture Notes in Computer Science*, pages 209–223. Springer International Publishing, 2014.

[102] K. Winfried. Finding transient solutions in Markovian event systems through randomization. *Numerical Solution of Markov Chains*, 8:357–380, 1991.

Appendix A

Chapter 5

A.1 The proof for Construction of the NCD Solution

We show that using Eq. 5.11, the solution leads to a block diagonal R^{D^*} in which each R_I^* is stochastic.

$$
\begin{aligned}
&\sum_{J=1}^{N}\sum_{j=1}^{ord(J)} r^*_{i_I j_J} \\
&\quad= \sum_{j=1}^{ord(I)} r^*_{i_I j_I} + \sum_{J=1,J\neq I}^{N}\sum_{j=1}^{ord(J)} r^*_{i_I j_J} \\
&\quad= \sum_{j=1}^{ord(I)} r^*_{i_I j_I} \\
&\quad= \sum_{j=1}^{ord(I)} \left(r_{i_I j_J} + \frac{r_{i_I j_J}}{\sum_{j=1}^{ord(I)} r_{i_I j_J}} \sum_{J=1,J\neq I}^{N}\sum_{j=1}^{ord(J)} ri_I j_J \right) = \\
&\quad= \sum_{j=1}^{ord(I)} r_{i_I j_J} + \frac{\sum_{j=1}^{ord(I)} r_{i_I j_J}}{\sum_{j=1}^{ord(I)} r_{i_I j_J}} \sum_{J=1,J\neq I}^{N}\sum_{j=1}^{ord(J)} r_{i_I j_J} = \\
&\quad= \sum_{j=1}^{ord(I)} r_{i_I j_J} + \sum_{J=1,J\neq I}^{N}\sum_{j=1}^{ord(J)} r_{i_I j_J} = \sum_{J=1}^{N}\sum_{j=1}^{ord(J)} ri_I j_J = 1
\end{aligned}
$$

A.2 Calculating the Probabilities of Cross-block Transitions

Let i be a state in B_I and j a state in B_J. Assuming that X^D is currently in i, then, after the next jump it will be in j with probability p_{ij}. Therefore, the likelihood of $X^D \subset J$ after the next jump from state i is:

$$\mathbb{P}\left(X^D \subset J\right) = \sum_{j=1}^{ord(J)} p_{ij}.$$

The joint probability of $X^D \subset I$ and after one step, $X^D \subset J$ is:

$$\mathbb{P}\left(\, X^D(t+1) \subset J \,\wedge\, X^D(t) \subset I \,\right) = \sum_{i=1}^{ord(I)} \pi_{i_I} \left(\sum_{j=1}^{ord(J)} p_{i_I j_J} \right) \tag{A.1}$$

The conditional probability of $X^D(t+1) \subset J$ given that $X^D(t) \subset I$ can be obtained by:

$$\begin{aligned} p'_{IJ} = \mathbb{P}\left(X^D(t+1) \subset J \;\middle|\; X^D(t) \subset I\right) &= \frac{\mathbb{P}\left(X^D(t+1) \subset J \wedge X^D(t) \subset I\right)}{\mathbb{P}\left(X^D(t) \subset I\right)} \\ &= \frac{\sum_{i=1}^{ord(I)} \pi_{i_I} \left(\sum_{j=1}^{ord(J)} p_{i_I j_J} \right)}{\theta_I} = \sum_{i=1}^{ord(I)} \mathbf{p}_{i_I} \times \sum_{j=1}^{ord(J)} p_{i_I j_J} \end{aligned}$$

Obtaining p'_{IJ}, $I,J = 1 \cdots N$ allows us to construct the macro DTMC which captures the evolution of X^D with respect to $[B_I]_{I=1}^N$.

Appendix B

Conditional Expectations

B.1 Expectation of a Linear Function of a Random Variable

Let X represent a random variable and $F : X \to \mathbb{R}$ be a linear function over X. We show:

$$\mathbb{E}[F(X)] = F(\mathbb{E}[X]) \tag{B.1}$$

Proof. We start by expanding $F(X)$ around $\mathbb{E}[X]$ using the Taylor expansion.

$$F(X) = F(\mathbb{E}[X]) + \frac{\partial F(\mathbb{E}[X])}{\partial X}(X - \mathbb{E}[X]) \tag{B.2}$$

Since F is a linear function, the terms above the first-order derivatives are zero. We now present the calculation for $\mathbb{E}[F(X)]$:

$$\mathbb{E}[F(X)] = \sum_{X} F(X) \cdot \mathbb{P}(X) \tag{B.3}$$

We substitute Eq.(B.3) in Eq.(B.2) to derive:

$$
\begin{aligned}
&\mathbb{E}[F(X)] \\
&= \sum_X \left(F(\mathbb{E}[X]) + \frac{\partial F(\mathbb{E}[X])}{\partial X}(X - \mathbb{E}[X]) \right) \cdot \mathbb{P}(X) \\
&= \sum_X \Big(F(\mathbb{E}[X]) \cdot \mathbb{P}(X) \Big) + \sum_X \Big(\frac{\partial F(\mathbb{E}[X])}{\partial X}(X - \mathbb{E}[X]) \cdot \mathbb{P}(X) \Big) \\
&= F(\mathbb{E}[X]) + \frac{\partial F(\mathbb{E}[X])}{\partial X} \cdot \Big(\sum_X (X - \mathbb{E}[X]) \cdot \mathbb{P}(X) \Big) \\
&= F(\mathbb{E}[X]) + \frac{\partial F(\mathbb{E}[X])}{\partial X} \left[\Big(\underbrace{\sum_X X \cdot \mathbb{P}(X)}_{\mathbb{E}[X]} \Big) - \mathbb{E}[X] \cdot \Big(\underbrace{\sum_X \mathbb{P}(X)}_{1} \Big) \right] \\
&= F(\mathbb{E}[X])
\end{aligned}
\tag{B.4}
$$

B.2 Proof of Proposition 6.4.2 - Initial Conditional Expectations

Consider an aggregated state $\boldsymbol{\gamma} \in D_z^{agg}$. We start by considering Eq.(B.5), the equation that captures the evolution of $\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$.

$$\begin{aligned}
&\mathbb{P}_t(\boldsymbol{\gamma}) \cdot \frac{d\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\right)}{d\,t} + \frac{d\left(\mathbb{P}_t(\boldsymbol{\gamma})\right)}{d\,t} \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right] \approx \\
&- \sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot r_\alpha(\boldsymbol{\gamma}) \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right] \\
&+ \sum_{\substack{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+}}} \mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma} - \mathcal{V}_\alpha^s\right] \\
&+ \sum_{\substack{\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+}}} \mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \mathcal{V}_\alpha^l \\
&+ \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \mathbb{P}_t(\boldsymbol{\gamma}) \cdot \mathcal{V}_\alpha^l \cdot r_\alpha\Big(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\Big)
\end{aligned} \tag{B.5}$$

First, using the reverse version of the product rule for differentiation, we transform the left hand side to obtain:

$$\mathbb{P}_t(\boldsymbol{\gamma}) \cdot \frac{d\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\right)}{d\,t} + \frac{d\left(\mathbb{P}_t(\boldsymbol{\gamma})\right)}{d\,t} \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right] = \frac{d}{d\,t}\Big(\mathbb{P}_t(\boldsymbol{\gamma}) \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\Big) \tag{B.6}$$

We substitute Eq.(B.6) as the left hand side in Eq.(B.5). Now, we differentiate Eq.(B.5) to the $(\nabla_{\boldsymbol{\gamma}}-1)$-th order:

$$\begin{aligned}
&\frac{d^{\nabla_{\boldsymbol{\gamma}}}\left(\mathbb{P}_t(\boldsymbol{\gamma})\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l\mid\boldsymbol{\gamma}\right]\right)}{d\,t^{\nabla_{\boldsymbol{\gamma}}}} = \\
&-\sum_{\alpha\in\vec{\mathcal{A}}^*_s(\mathbb{M})\cup\vec{\mathcal{A}}^*_{sl}(\mathbb{M})} r_\alpha(\boldsymbol{\gamma})\cdot\frac{d^{(\nabla\boldsymbol{\gamma}-1)}\left(\mathbb{P}_t(\boldsymbol{\gamma})\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l\mid\boldsymbol{\gamma}\right]\right)}{d\,t^{(\nabla\boldsymbol{\gamma}-1)}} && \text{I}\\
&+\sum_{\substack{\alpha\in\vec{\mathcal{A}}^*_s(\mathbb{M})\cup\vec{\mathcal{A}}^*_{sl}(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}^{s,+}_\alpha}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha)\cdot\frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\left(\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha)\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l\mid\boldsymbol{\gamma}-\mathcal{V}^s_\alpha\right]\right)}{d\,t^{(\nabla_{\boldsymbol{\gamma}}-1)}} && \text{II}\\
&+\sum_{\substack{\alpha\in\vec{\mathcal{A}}^*_{sl}(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}^{s,+}_\alpha}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha)\cdot\mathcal{V}^l_\alpha\cdot\frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\left(\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha)\right)}{d\,t^{(\nabla_{\boldsymbol{\gamma}}-1)}} && \text{III}\\
&+\sum_{\alpha\in\vec{\mathcal{A}}^*_l(\mathbb{M})}\mathcal{V}^l_\alpha\cdot\frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\left(\mathbb{P}_t(\boldsymbol{\gamma})\cdot r_\alpha\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l\mid\boldsymbol{\gamma}\right]\right)\right)}{d\,t^{(\nabla_{\boldsymbol{\gamma}}-1)}} && \text{IV}
\end{aligned}\tag{B.7}$$

In Eq.(B.7), we have expressions that consist of differentiations over products. The Leibniz rule [31] is used to expand these terms. According to this rule, for any two functions f and g, both with at least m derivatives:

$$\frac{d^m\,(f\times g)}{d\,t^m}=\sum_{k=0}^{m}\binom{m}{k}\frac{d^k\,(f)}{d\,t^k}\times\frac{d^{(m-k)}\,(g)}{d\,t^{(m-k)}}\tag{B.8}$$

First, we apply the rule on the left hand side to have:

$$\frac{d^{\nabla_{\boldsymbol{\gamma}}}\left(\mathbb{P}_t(\boldsymbol{\gamma})\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l\mid\boldsymbol{\gamma}\right]\right)}{d\,t^{\nabla_{\boldsymbol{\gamma}}}}=\sum_{k=0}^{\nabla\boldsymbol{\gamma}}\binom{\nabla_{\boldsymbol{\gamma}}}{k}\cdot\frac{d^k\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^k}\cdot\frac{d^k\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l\mid\boldsymbol{\gamma}\right]\right)}{d\,t^k}\tag{B.9}$$

Evaluating at $t=t_0$, given that $\boldsymbol{\gamma}$ has distance $\nabla_{\boldsymbol{\gamma}}$, for $k=0,1,\cdots(\nabla_{\boldsymbol{\gamma}}-1)$: $\frac{d^k\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^k}\Big|_{t_0}=0$ and when $k=\nabla_{\boldsymbol{\gamma}}$: $\frac{d^k\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^k}\Big|_{t_0}\neq 0$. Therefore, the left hand side is reduced to:

$$\frac{d^{\nabla_{\boldsymbol{\gamma}}}\left(\mathbb{P}_t(\boldsymbol{\gamma})\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l\mid\boldsymbol{\gamma}\right]\right)}{d\,t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}=\frac{d^{(\nabla_{\boldsymbol{\gamma}})}\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{(\nabla_{\boldsymbol{\gamma}})}}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l\mid\boldsymbol{\gamma}\right]\tag{B.10}$$

Note here that on the left hand side, the term containing the conditional expectation $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l\mid\boldsymbol{\gamma}\right]$ is preserved. Now, let us consider the right hand side and apply the rule again.

By applying the rule on the expression marked by (I) (related to the outward transitions enabled by $\boldsymbol{\gamma}$), we have:

$$\sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma}) \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)} \left(\mathbb{P}_t(\boldsymbol{\gamma}) \cdot \mathbb{E}_t \left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma} \right] \right)}{d\, t^{(\nabla_{\boldsymbol{\gamma}}-1)}} =$$

$$\sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma}) \sum_{k=0}^{(\nabla_{\boldsymbol{\gamma}}-1)} \binom{(\nabla_{\boldsymbol{\gamma}}-1)}{k} \cdot \frac{d^k \left(\mathbb{P}_t(\boldsymbol{\gamma}) \right)}{d\, t^k} \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1-k)} \left(\mathbb{E}_t \left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma} \right] \right)}{d\, t^{(\nabla_{\boldsymbol{\gamma}}-1-k)}} \quad \text{(B.11)}$$

When evaluated at $t = t_0$, for $k = 0, 1, \cdots, (\nabla_{\boldsymbol{\gamma}} - 1)$: $\frac{d^k \left(\mathbb{P}_t(\boldsymbol{\gamma}) \right)}{d\, t^k}\Big|_{t_0} = 0$. Therefore, in Eq.(B.11) and the second summation, all terms reduce to zero:

$$\sum_{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M})} r_\alpha(\boldsymbol{\gamma}) \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)} \left(\mathbb{P}_t(\boldsymbol{\gamma}) \cdot \mathbb{E}_t \left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma} \right] \right)}{d\, t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0} = 0 \quad \text{(B.12)}$$

Next, we consider the term marked by (II) (related to the inward transitions into $\boldsymbol{\gamma}$). Applying the Leibniz rule, we get:

$$\sum_{\substack{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+}}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)} \left(\mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \mathbb{E}_t \left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma} - \mathcal{V}_\alpha^s \right] \right)}{d\, t^{(\nabla_{\boldsymbol{\gamma}}-1)}} =$$

$$\sum_{\substack{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+}}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \sum_{k=0}^{(\nabla_{\boldsymbol{\gamma}}-1)} \binom{\nabla_{\boldsymbol{\gamma}}-1}{k} \frac{d^k \left(\mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \right)}{d\, t^k} \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1-k)} \left(\mathbb{E}_t \left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma} - \mathcal{V}_\alpha^s \right] \right)}{d\, t^{(\nabla_{\boldsymbol{\gamma}}-1-k)}} \quad \text{(B.13)}$$

In this equation, the first summation is over the action types α which enable the transitions $\boldsymbol{\gamma}' \rightarrow \boldsymbol{\gamma}$, $\boldsymbol{\gamma}' = \boldsymbol{\gamma} - \mathcal{V}_\alpha^s$, and the second summation is over the expressions in terms of the first $(\nabla_{\boldsymbol{\gamma}} - 1)$ derivatives of $\mathbb{P}_t(\boldsymbol{\gamma}')$. There is a one-to-one relation between the action types and the enabling states. This equation can be simplified by categorising the enabling states with respect to their distances.

If $\nabla_{\boldsymbol{\gamma}} \leq \nabla_{\boldsymbol{\gamma}'}$, then: $(\nabla_{\boldsymbol{\gamma}} - 1) \leq (\nabla_{\boldsymbol{\gamma}'} - 1)$. Consequently, for $k = 0, \cdots, (\nabla_{\boldsymbol{\gamma}} - 1)$: $\frac{d^k \left(\mathbb{P}_t(\boldsymbol{\gamma}') \right)}{d\, t^k}|_{t_0} = 0$. This means that all action types which enable inward transitions into $\boldsymbol{\gamma}$ from $\boldsymbol{\gamma}'$ with $\nabla_{\boldsymbol{\gamma}'} \geq \nabla_{\boldsymbol{\gamma}}$, contribute zero to the equation. On the other hand, when $\nabla_{\boldsymbol{\gamma}'} = (\nabla_{\boldsymbol{\gamma}} - 1)$, then for all $k = 0, 1, \cdots (\nabla_{\boldsymbol{\gamma}} - 1 - 1)$: $\frac{d^k \left(\mathbb{P}_t(\boldsymbol{\gamma}') \right)}{d\, t^k}|_{t_0} = 0$. The only

non-zero terms are related to $k = \nabla_{\boldsymbol{\gamma}'} = (\nabla_{\boldsymbol{\gamma}} - 1)$, since in this case:

$$\frac{d^{\nabla_{\boldsymbol{\gamma}'}} \left(\mathbb{P}_t(\boldsymbol{\gamma}')\right)}{d\, t^{\nabla_{\boldsymbol{\gamma}'}}}\Big|_{t_0} = \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)} \left(\mathbb{P}_t(\boldsymbol{\gamma}) - \mathcal{V}_\alpha^s\right)}{d\, t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0} \neq 0 \tag{B.14}$$

By removing the zero terms from Eq.(B.13), we obtain Eq.(B.15).

$$\sum_{\substack{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+}}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)} \left(\mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma} - \mathcal{V}_\alpha^s\right]\right)}{d\, t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0} =$$

$$\sum_{\substack{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+} \\ \underline{\nabla_{(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)} = \nabla_{\boldsymbol{\gamma}} - 1}}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)} \left(\mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)\right)}{d\, t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0} \cdot \mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma} - \mathcal{V}_\alpha^s\right] \tag{B.15}$$

where the underlined term indicates that only the terms associated with states $(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \in D^{agg}$ with initial distances $\nabla_{(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)} = \nabla_{\boldsymbol{\gamma}} - 1$ are non-zero.

Now we focus on the term marked by (III). Following a similar approach as above, the only non-zero contributions are related to inward transitions from states $\boldsymbol{\gamma}'$ where $\nabla_{\boldsymbol{\gamma}'} = \nabla_{\boldsymbol{\gamma}} - 1$:

$$\sum_{\substack{\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+}}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \mathcal{V}_\alpha^l \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)} \left(\mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)\right)}{d\, t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0} =$$

$$\sum_{\substack{\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+} \\ \nabla_{(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)} = \nabla_{\boldsymbol{\gamma}} - 1}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \mathcal{V}_\alpha^l \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)} \left(\mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)\right)}{d\, t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0} \tag{B.16}$$

Finally, we consider the term marked by (IV), which is related to the $\vec{\mathcal{A}}_l^*(\mathbb{M})$. By applying the Leibniz rule, we have:

$$\sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \mathcal{V}_\alpha^l \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)} \left(\mathbb{P}_t(\boldsymbol{\gamma}) \cdot r_\alpha\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\right)\right)}{d\, t^{(\nabla_{\boldsymbol{\gamma}}-1)}}$$

$$= \sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \mathcal{V}_\alpha^l \cdot \sum_{k=0}^{(\nabla_{\boldsymbol{\gamma}}-1)} \binom{(\nabla_{\boldsymbol{\gamma}} - 1)}{k} \cdot \frac{d^k \left(\mathbb{P}_t(\boldsymbol{\gamma})\right)}{d\, t^k} \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1-k)} \left(r_\alpha\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\right)\right)}{d\, t^{(\nabla_{\boldsymbol{\gamma}}-1-k)}} \tag{B.17}$$

At $t = t_0$, since for $k = 0, 1, \cdots, (\nabla_{\boldsymbol{\gamma}} - 1)$: $\frac{d^k\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\ t^k}\Big|_{t_0} = 0$, all the terms in the equation reduce to zero. Therefore:

$$\sum_{\alpha \in \vec{\mathcal{A}}_l^*(\mathbb{M})} \mathcal{V}_\alpha^l \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\ \left(\mathbb{P}_t(\boldsymbol{\gamma}) \cdot r_\alpha\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\right)\right)}{d\ t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0} = 0 \tag{B.18}$$

We simplified terms marked by (I-IV) in Equations (B.12),(B.15), (B.16) and (B.18). By substituting the results back into Eq.(B.7), we derive:

$$\begin{aligned}
&\frac{d^{(\nabla_{\boldsymbol{\gamma}})}\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\ t^{(\nabla_{\boldsymbol{\gamma}})}}\Big|_{t_0} \cdot \mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right] \;=\; \\
&\qquad \sum_{\substack{\alpha \in \vec{\mathcal{A}}_s^*(\mathbb{M}) \cup \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+} \\ \nabla_{(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)} = \nabla_{\boldsymbol{\gamma}} - 1}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\ (\mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s))}{d\ t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0} \cdot \mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma} - \mathcal{V}_\alpha^s\right] \\
&+ \qquad \sum_{\substack{\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+} \\ \nabla_{(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)} = \nabla_{\boldsymbol{\gamma}} - 1}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \mathcal{V}_\alpha^l \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\ (\mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s))}{d\ t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0} \\
&= \qquad \sum_{\substack{\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+} \\ \nabla_{(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)} = \nabla_{\boldsymbol{\gamma}} - 1}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\ (\mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s))}{d\ t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0} \left(\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma} - \mathcal{V}_\alpha^s\right] + \mathcal{V}_\alpha^l\right)
\end{aligned} \tag{B.19}$$

As Eq.(B.19) shows, for $\boldsymbol{\gamma}$ with distance $(\nabla_{\boldsymbol{\gamma}})$, the differentiation of its conditional expectation equation at order $(\nabla_{\boldsymbol{\gamma}} - 1)$ preserves $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$ on the left hand side. Therefore, by the reordering of the terms we have:

$$\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right] = \frac{\sum_{\substack{\alpha \in \vec{\mathcal{A}}_{sl}^*(\mathbb{M}) \\ \boldsymbol{\gamma} \geq \mathcal{V}_\alpha^{s,+}, \nabla_{(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s)} = \nabla_{\boldsymbol{\gamma}} - 1}} r_\alpha(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s) \cdot \frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\ (\mathbb{P}_t(\boldsymbol{\gamma} - \mathcal{V}_\alpha^s))}{d\ t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0} \left(\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma} - \mathcal{V}_\alpha^s\right] + \mathcal{V}_\alpha^l\right)}{\frac{d^{(\nabla_{\boldsymbol{\gamma}})}\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\ t^{(\nabla_{\boldsymbol{\gamma}})}}\Big|_{t_0}} \tag{B.20}$$

Eq.(B.20) shows that for any $\boldsymbol{\gamma} \in D^{agg}$ with distance $\nabla_{\boldsymbol{\gamma}}$, the initial value $\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$, depends on the $(\nabla_{\boldsymbol{\gamma}})$-th derivative of $\mathbb{P}_t(\boldsymbol{\gamma})\big|_{t_0}$ and the initial conditional expectations of the states $\boldsymbol{\gamma}'$ with distance $\nabla_{\boldsymbol{\gamma}} - 1$ that enable the incoming transitions into $\boldsymbol{\gamma}$. This concludes the proof of Prop. 6.4.2.

B.3 Proof of Proposition 6.4.3 - Initial Derivatives of Conditional Expectations

The outline of this proof is the following. For an aggregated state $\boldsymbol{\gamma}$, we consider the equation that captures $\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$ and differentiate it to the $\nabla_{\boldsymbol{\gamma}}$-th order. The resulting equation is evaluated at t_0 and simplified to remove the terms that are equal to zero. In our simplifications, we make extensive use of Prop. 6.4.1, using the same logic that was presented in the proof of Prop. 6.4.2. Due to this similarity, a number of steps are omitted.

Consider an aggregated state $\boldsymbol{\gamma} \in D_z^{agg}$ with distance $\nabla_{\boldsymbol{\gamma}}$. The equation capturing $\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]$ is Eq.(B.5). Let us focus on the left hand side. By forming the $\nabla_{\boldsymbol{\gamma}}$-th derivative of the equation and using the Liebniz rule, we have:

$$\frac{d^{(\nabla_{\boldsymbol{\gamma}}+1)}\left(\mathbb{P}_t(\boldsymbol{\gamma})\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\right)}{d\,t^{(\nabla_{\boldsymbol{\gamma}}+1)}} = \sum_{k=0}^{(\nabla_{\boldsymbol{\gamma}}+1)}\binom{(\nabla_{\boldsymbol{\gamma}}+1)}{k}\cdot\frac{d^k\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^k}\cdot\frac{d^k\left(\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\right)}{d\,t^k} \tag{B.21}$$

By evaluating the left hand side at t_0 and using Prop.6.4.1, we derive:

$$\begin{aligned}\frac{d^{(\nabla_{\boldsymbol{\gamma}}+1)}\left(\mathbb{P}_t(\boldsymbol{\gamma})\cdot\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\right)}{d\,t^{(\nabla_{\boldsymbol{\gamma}}+1)}}\Big|_{t_0} = (\nabla_{\boldsymbol{\gamma}}+1)\cdot\frac{d^{\nabla_{\boldsymbol{\gamma}}}\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}\cdot\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]}{d\,t}\Big|_{t_0} + \\ \frac{d^{(\nabla_{\boldsymbol{\gamma}}+1)}\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{(\nabla_{\boldsymbol{\gamma}}+1)}}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]\end{aligned} \tag{B.22}$$

In Eq.(B.22), given that $\boldsymbol{\gamma}$ has the distance $\nabla_{\boldsymbol{\gamma}}$, we have $\frac{d^{(\nabla_{\boldsymbol{\gamma}})}\,(\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{(\nabla_{\boldsymbol{\gamma}})}}\Big|_{t_0} \neq 0$ and therefore, the initial derivative $\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l \mid \boldsymbol{\gamma}\right]}{d\,t}\Big|_{t_0}$ is preserved.

We substitute Eq.(B.22) in Eq.(B.5). By performing the same steps on the right

hand side, Eq.(B.5) is transformed into:

$$
\begin{aligned}
&(\nabla_{\boldsymbol{\gamma}}+1)\cdot\frac{d^{\nabla_{\boldsymbol{\gamma}}}\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}\cdot\underline{\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}\right]}{d\,t}\Big|_{t_0}}+\frac{d^{(\nabla_{\boldsymbol{\gamma}}+1)}\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{(\nabla_{\boldsymbol{\gamma}}+1)}}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}\right] \;= \\
&-\sum_{\alpha\in\vec{\mathcal{A}}^*_s(\mathbb{M})\cup\vec{\mathcal{A}}^*_{sl}(\mathbb{M})} r_\alpha(\boldsymbol{\gamma})\cdot\frac{d^{\nabla_{\boldsymbol{\gamma}}}\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}\right] \qquad \text{I} \\
&+\sum_{\substack{\alpha\in\vec{\mathcal{A}}^*_s(\mathbb{M})\cup\vec{\mathcal{A}}^*_{sl}(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}^{s,+}_\alpha\\ \underline{\nabla_{(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha)}<\nabla_{\boldsymbol{\gamma}}}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha)\cdot\left(\frac{d^{(\nabla_{\boldsymbol{\gamma}})}\ (\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha))}{d\,t^{(\nabla_{\boldsymbol{\gamma}})}}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}-\mathcal{V}^s_\alpha\right]+\right. \qquad \text{II} \\
&\qquad\left.\nabla_{\boldsymbol{\gamma}}\cdot\frac{d^{(\nabla_{\boldsymbol{\gamma}}-1)}\ (\mathbb{P}(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha))}{d\,t^{(\nabla_{\boldsymbol{\gamma}}-1)}}\Big|_{t_0}\cdot\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}-\mathcal{V}^s_\alpha\right]}{d\,t}\Big|_{t_0}\right) \qquad \text{III} \\
&+\sum_{\substack{\alpha\in\vec{\mathcal{A}}^*_s(\mathbb{M})\cup\vec{\mathcal{A}}^*_{sl}(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}^{s,+}_\alpha\\ \underline{\nabla_{(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha)}=\nabla_{\boldsymbol{\gamma}}}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha)\cdot\left(\frac{d^{(\nabla_{\boldsymbol{\gamma}})}\ (\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha))}{d\,t^{(\nabla_{\boldsymbol{\gamma}})}}\Big|_{t_0}\cdot\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}-\mathcal{V}^s_\alpha\right]\right. \qquad \text{IV} \\
&+\sum_{\substack{\alpha\in\vec{\mathcal{A}}^*_{sl}(\mathbb{M})\\ \boldsymbol{\gamma}\geq\mathcal{V}^{s,+}_\alpha\\ \underline{\nabla_{(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha)}\leq\nabla_{\boldsymbol{\gamma}}-1}}} r_\alpha(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha)\ \cdot\mathcal{V}^l_\alpha\cdot\frac{d^{(\nabla_{\boldsymbol{\gamma}})}\ (\mathbb{P}_t(\boldsymbol{\gamma}-\mathcal{V}^s_\alpha))}{d\,t^{(\nabla_{\boldsymbol{\gamma}})}}\Big|_{t_0} \\
&+\sum_{\alpha\in\vec{\mathcal{A}}^*_l(\mathbb{M})}\mathcal{V}^l_\alpha\cdot\frac{d^{\nabla_{\boldsymbol{\gamma}}}\ (\mathbb{P}_t(\boldsymbol{\gamma}))}{d\,t^{\nabla_{\boldsymbol{\gamma}}}}\Big|_{t_0}\cdot r_\alpha\Big(\mathbb{E}_{t_0}\left[\boldsymbol{\xi}^l\,|\,\boldsymbol{\gamma}\right]\Big) \qquad \text{V} \qquad \text{(B.23)}
\end{aligned}
$$

Let us consider the terms in this equation. Here, the expressions marked by (I), (II), (IV) and (V) depend on the initial conditional expectations, which are obtained using Prop. 6.4.2. Moreover, the expression (III) consists of the derivative of initial conditional expectations for the states $\boldsymbol{\gamma}'$ which enable transitions into $\boldsymbol{\gamma}$ and have strictly less distance than $\boldsymbol{\gamma}$. We can assume that these initial derivatives have been obtained using a similar version of Eq.(B.23), constructed at a reduced scale. Having considered the expressions, we conclude that in Eq.(B.23) the only variable unknown is $\frac{d\,\mathbb{E}_t\left[\boldsymbol{\xi}^l|\right]}{d\,t}\Big|_{t_0}$, which can be derived by reordering of the terms. This ends the proof of Prop. 6.4.3.

Appendix C

Application of Conditional Expectation to Client-Server System

C.1 Equations for the Conditional Expectations of the Client-Server System

The aggregated state space has 6 states. First, let us consider the state $\langle 2,0,0\rangle$:

$$\begin{aligned}
&\mathbb{P}_t(\langle 2,0,0\rangle)\cdot\frac{\partial\left(\,\mathbb{E}_t\left[\,\langle C_t,C_r\rangle \mid \langle 2,0,0\rangle\,\right]\,\right)}{\partial\, t}+\frac{\partial\,\mathbb{P}_t(\langle 2,0,0\rangle)}{\partial\, t}\cdot\mathbb{E}_t\left[\,\langle C_t,C_r\rangle \mid \langle 2,0,0\rangle\,\right]\approx\\
&\quad-\ \mathbb{P}_t(\langle 2,0,0\rangle)\cdot\mathbb{E}_t\left[\,\langle C_t,C_r\rangle \mid \langle 2,0,0\rangle\,\right]\cdot\Big[r_{brk}(\langle 2,0,0\rangle)+r_{req}(\langle 2,0,0\rangle)\Big]\\
&\quad+\ \mathbb{P}_t(\langle 1,0,1\rangle)\cdot r_{fix}(\langle 1,0,1\rangle)\cdot\mathbb{E}_t\left[\,\langle C_t,C_r\rangle \mid \langle 1,0,1\rangle\,\right]\\
&\quad+\ \mathbb{P}_t(\langle 1,1,0\rangle)\cdot r_{log}(\langle 1,1,0\rangle)\cdot\mathbb{E}_t\left[\,\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle\,\right]\\
&\quad+\ \mathbb{P}_t(\langle 2,0,0\rangle)\cdot\langle -1,+1\rangle\cdot r_{think}\Big(\mathbb{E}_t\left[\,\langle C_t,C_r\rangle \mid \langle 2,0,0\rangle\,\right]\Big)
\end{aligned}\tag{C.1}$$

For the aggregated state $\langle 1,1,0\rangle$:

$$\mathbb{P}_t(\langle 1,1,0\rangle)\cdot\frac{\partial\left(\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle\right]\right)}{\partial t}+\frac{\partial\,\mathbb{P}_t(\langle 1,1,0\rangle)}{\partial t}\cdot\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle\right]\approx$$

$$\begin{aligned}
&-\ \mathbb{P}_t(\langle 1,1,0\rangle)\cdot\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle\right]\cdot\Big[r_{log}(\langle 1,1,0\rangle)+r_{brk}(\langle 1,1,0\rangle)+r_{req}(\langle 1,1,0\rangle)\Big]\\
&+\ \mathbb{P}_t(\langle 2,0,0\rangle)\cdot r_{req}(\langle 2,0,0\rangle)\cdot\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 2,0,0\rangle\right]\\
&+\ \mathbb{P}_t(\langle 0,1,1\rangle)\cdot r_{fix}(\langle 0,1,1\rangle)\cdot\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 0,1,1\rangle\right]\\
&+\ \mathbb{P}_t(\langle 0,2,0\rangle)\cdot r_{log}(\langle 0,2,0\rangle)\cdot\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 0,2,0\rangle\right]\\
&+\ \mathbb{P}_t(\langle 2,0,0\rangle)\cdot r_{req}(\langle 2,0,0\rangle)\cdot\langle +1,-1\rangle\\
&+\ \mathbb{P}_t(\langle 1,1,0\rangle)\cdot\mathcal{V}^{l}_{think}\cdot r_{think}\Big(\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle\right]\Big)
\end{aligned}\tag{C.2}$$

For the aggregated states $\langle 0,2,0\rangle$:

$$\mathbb{P}_t(\langle 0,2,0\rangle)\cdot\frac{\partial\left(\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 0,2,0\rangle\right]\right)}{\partial t}+\frac{\partial\,\mathbb{P}_t(\langle 0,2,0\rangle)}{\partial t}\cdot\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 0,2,0\rangle\right]\approx$$

$$\begin{aligned}
&-\ \mathbb{P}_t(\langle 0,2,0\rangle)\cdot\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 0,2,0\rangle\right]\cdot r_{log}(\langle 0,2,0\rangle)\\
&+\ \mathbb{P}_t(\langle 1,1,0\rangle)\cdot r_{req}(\langle 1,1,0\rangle)\cdot\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle\right]\\
&+\ \mathbb{P}_t(\langle 1,1,0\rangle)\cdot r_{req}(\langle 1,1,0\rangle)\cdot\langle +1,-1\rangle\\
&+\ \mathbb{P}_t(\langle 0,2,0\rangle)\cdot\langle -1,+1\rangle\cdot r_{think}\Big(\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 0,2,0\rangle\right]\Big)
\end{aligned}$$

For $\boldsymbol{\beta}=\langle 1,0,1\rangle$, the evolution of $\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 0,1,1\rangle\right]$, is captured by:

$$\mathbb{P}_t(\langle 1,0,1\rangle)\cdot\frac{\partial\left(\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 1,0,1\rangle\right]\right)}{\partial t}+\frac{\partial\,\mathbb{P}_t(\langle 1,0,1\rangle)}{\partial t}\cdot\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 1,0,1\rangle\right]\approx$$

$$\begin{aligned}
&-\ \mathbb{P}_t(\langle 1,0,1\rangle)\cdot\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 1,0,1\rangle\right]\cdot\Big[r_{brk}(\langle 1,0,1\rangle)+r_{fix}(\langle 1,0,1\rangle)+r_{req}(\langle 1,0,1\rangle)\Big]\\
&+\ \mathbb{P}_t(\langle 2,0,0\rangle)\cdot r_{brk}(\langle 2,0,0\rangle)\cdot\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 2,0,0\rangle\right]\\
&+\ \mathbb{P}_t(\langle 0,0,2\rangle)\cdot r_{fix}(\langle 0,0,2\rangle)\cdot\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 0,0,2\rangle\right]\\
&+\ \mathbb{P}_t(\langle 0,1,1\rangle)\cdot r_{log}(\langle 0,1,1\rangle)\cdot\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 0,1,1\rangle\right]\\
&+\ \mathbb{P}_t(\langle 1,0,1\rangle)\cdot\langle -1,+1\rangle\cdot r_{think}(\mathbb{E}_t\left[\langle C_t,C_r\rangle \mid \langle 1,0,1\rangle\right])
\end{aligned}\tag{C.3}$$

For state $\langle 0,1,1\rangle$, we have:

$$\mathbb{P}_t(\langle 0,1,1\rangle)\cdot\frac{\partial\left(\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 0,1,1\rangle]\right)}{\partial t}+\frac{\partial\,\mathbb{P}_t(\langle 0,1,1\rangle)}{\partial t}\cdot\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 0,1,1\rangle] \approx$$

$$\begin{aligned}
&- \mathbb{P}_t(\langle 0,1,1\rangle)\cdot\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 0,1,1\rangle]\cdot\Big[r_{log}(\langle 0,1,1\rangle)+r_{fix}(\langle 0,1,1\rangle)\Big]\\
&+ \mathbb{P}_t(\langle 1,0,1\rangle)\cdot r_{req}(\langle 1,0,1\rangle)\cdot\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 1,0,1\rangle]\\
&+ \mathbb{P}_t(\langle 1,1,0\rangle)\cdot r_{brk}(\langle 1,1,0\rangle)\cdot\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle]\\
&+ \mathbb{P}_t(\langle 1,0,1\rangle)\cdot r_{req}(\langle 1,1,0\rangle)\cdot\langle +1,-1\rangle\\
&+ \mathbb{P}_t(\langle 0,1,1\rangle)\cdot\langle -1,+1\rangle\cdot r_{think}\Big(\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 0,1,1\rangle]\Big)
\end{aligned}\tag{C.4}$$

For the state $\langle 0,0,2\rangle$ we have:

$$\mathbb{P}_t(\langle 0,0,2\rangle)\cdot\frac{\partial\left(\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 0,0,2\rangle]\right)}{\partial t}+\frac{\partial\,\mathbb{P}_t(\langle 0,0,2\rangle)}{\partial t}\cdot\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 0,0,2\rangle] \approx$$

$$\begin{aligned}
&- \mathbb{P}_t(\langle 0,0,2\rangle)\cdot\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 0,0,2\rangle]\cdot\Big[r_{fix}(\langle 0,0,2\rangle)\Big]\\
&+ \mathbb{P}_t(\langle 1,0,1\rangle)\cdot r_{brk}(\langle 1,0,1\rangle)\cdot\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 1,0,1\rangle]\\
&+ \mathbb{P}_t(\langle 0,0,2\rangle)\cdot\langle -1,+1\rangle\cdot r_{think}\Big(\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 0,0,2\rangle]\Big)
\end{aligned}\tag{C.5}$$

C.2 Derivations of the Initial Values

From the states in D_z^{agg}, we choose the states $\langle 1,1,0\rangle$, $\langle 1,0,1\rangle$ and $\langle 0,1,1\rangle$ and illustrate how their initial values are obtained. The derivations for the rest of the states is similar.

The state $\langle 1,1,0\rangle$ has the distance $\nabla_{\langle 1,1,0\rangle}=1$. According to Prop. 6.4.2, we can evaluate the equation capturing $\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle]$ at t_0 and derive $\mathbb{E}_{t_0}[\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle]$. By evaluating Eq.(C.2) at t_0, we get:

$$2r_s\cdot\mathbb{E}_{t_0}[\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle] \approx 2r_s\cdot\mathbb{E}_{t_0}[\langle C_t,C_r\rangle \mid \langle 2,0,0\rangle]+2r_s\cdot\langle +1,-1\rangle$$

By reordering of the terms we have:

$$\mathbb{E}_{t_0}[\langle C_t,C_r\rangle \mid \langle 1,1,0\rangle]=\mathbb{E}_{t_0}[\langle C_t,C_r\rangle \mid \langle 2,0,0\rangle]+\langle +1,-1\rangle=\langle 151,-1\rangle \tag{C.6}$$

	Distance	Probability	First derivative	Second derivative
$\langle 2,0,0\rangle$	$\nabla_{\langle 2,0,0\rangle}=0$	$\mathbb{P}_{t_0}(\langle 2,0,0\rangle)\neq 0$	-	-
$\langle 1,1,0\rangle$	$\nabla_{\langle 1,1,0\rangle}=1$	$\mathbb{P}_{t_0}(\langle 1,1,0\rangle)=0$	$\frac{d\ \mathbb{P}_t(\langle 1,1,0\rangle)}{d\ t}\Big\vert_{t_0}=2r_s$	-
$\langle 0,2,0\rangle$	$\nabla_{\langle 0,2,0\rangle}=2$	$\mathbb{P}_{t_0}(\langle 0,2,0\rangle)=0$	$\frac{d\ \mathbb{P}_t(\langle 0,2,0\rangle)}{d\ t}\Big\vert_{t_0}=0$	$\frac{d^2\ (\mathbb{P}_t(\langle 0,2,0\rangle))}{d\ t^2}\Big\vert_{t_0}=2r_s^2$
$\langle 1,0,1\rangle$	$\nabla_{\langle 1,0,1\rangle}=1$	$\mathbb{P}_{t_0}(\langle 1,0,1\rangle)=0$	$\frac{d\ \mathbb{P}_t(\langle 1,0,1\rangle)}{d\ t}\Big\vert_{t_0}=2r_b$	-
$\langle 0,1,1\rangle$	$\nabla_{\langle 0,1,1\rangle}=2$	$\mathbb{P}_{t_0}(\langle 0,1,1\rangle)=0$	$\frac{d\ \mathbb{P}_t(\langle 0,1,1\rangle)}{d\ t}\Big\vert_{t_0}=0$	$\frac{d^2\ (\mathbb{P}_t(\langle 0,1,1\rangle))}{d\ t^2}\Big\vert_{t_0}=4r_r r_b$
$\langle 0,0,2\rangle$	$\nabla_{\langle 0,0,2\rangle}=2$	$\mathbb{P}_{t_0}(\langle 0,0,2\rangle)=0$	$\frac{d\ \mathbb{P}_t(\langle 0,0,2\rangle)}{d\ t}\Big\vert_{t_0}=0$	$\frac{d^2\ (\mathbb{P}_t(\langle 0,0,2\rangle))}{d\ t^2}\Big\vert_{t_0}=2r_b^2$

Table C.1: The derivatives of $\mathbb{P}_t(\boldsymbol{\gamma}),\boldsymbol{\gamma}\in D^{agg}$ at $t=t_0$

The state $\langle 1,0,1\rangle$ has the same distance as $\langle 1,1,0\rangle$. By evaluating the equation which captures the evolution of $\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 1,0,1\rangle]$ at t_0 and simplifying the terms, we derive:

$$2r_b\cdot\mathbb{E}_{t_0}[\langle C_t,C_r\rangle\,|\,\langle 1,0,1\rangle] \approx 2r_b\cdot\mathbb{E}_{t_0}[\langle C_t,C_r\rangle\,|\,\langle 2,0,0\rangle] \Longrightarrow$$
$$\mathbb{E}_{t_0}[\langle C_t,C_r\rangle\,|\,\langle 1,0,1\rangle] \approx \mathbb{E}_{t_0}[\langle C_t,C_r\rangle\,|\,\langle 2,0,0\rangle] \quad \text{(C.7)}$$

For the state $\langle 0,1,1\rangle$, we have the distance$\nabla_{\langle 0,1,1\rangle}=2$. Using Prop. 6.4.2, in order to find $\mathbb{E}_{t_0}[\langle C_t,C_r\rangle\,|\,\langle 0,1,1\rangle]$ we need to differentiate the equation capturing the evolution of $\mathbb{E}_t[\langle C_t,C_r\rangle \mid \langle 0,1,1\rangle]$ once and then evaluate it at t_0. By following these steps, we derive:

$$(4r_sr_b)\cdot\mathbb{E}_{t_0}[\langle C_t,C_r\rangle\,|\,\langle 0,1,1\rangle] = 2r_sr_b\cdot\mathbb{E}_{t_0}[\langle C_t,C_r\rangle\,|\,\langle 1,1,0\rangle] +$$
$$2r_sr_b\cdot\mathbb{E}_{t_0}[\langle C_t,C_r\rangle\,|\,\langle 1,0,1\rangle] + 2r_sr_b\cdot\langle +1,-1\rangle$$
$$\Longrightarrow \mathbb{E}_{t_0}[\langle C_t,C_r\rangle\,|\,\langle 0,1,1\rangle] = \frac{\langle 151,-1\rangle+\langle 150,0\rangle+\langle +1,-1\rangle}{2} = \langle 151,-1\rangle \quad \text{(C.8)}$$

Initial Derivatives

For the state $\langle 2,0,0\rangle$, we calculated $\mathbb{E}_{t_0}[\langle C_t,C_r\rangle\,|\,\langle 2,0,0\rangle]$ in the previous section. The initial value $\frac{d\ \mathbb{E}_t[\langle C_t,C_r\rangle|\langle 2,0,0\rangle]}{d\ t}\Big\vert_{t_0}$ is derived by substituting $\mathbb{E}_{t_0}[\langle C_t,C_r\rangle\,|\,\langle 2,0,0\rangle]$ in its

conditional expectation equation and evaluating the equation at t_0:

$$\frac{d\,\mathbb{E}_t\,[\,\langle C_t, C_r\rangle \mid \langle 2,0,0\rangle\,]}{d\,t}\Big|_{t_0} + (-2r_s - 2r_b)\cdot \mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \langle 2,0,0\rangle\,] =$$

$$(-2r_s - 2r_b)\cdot \mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \langle 2,0,0\rangle\,] \;+\; \langle -1,+1\rangle \cdot r_t \cdot \mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \langle 2,0,0\rangle\,]$$

$$\Longrightarrow \frac{d\,\mathbb{E}_t\,[\,\langle C_t, C_r\rangle \mid \langle 2,0,0\rangle\,]}{d\,t}\Big|_{t_0} = \langle -1,+1\rangle \cdot r_{think} \cdot \langle 150,0\rangle = \langle -r_{think}\cdot 150, 0\rangle \tag{C.9}$$

For the states $\boldsymbol{\beta} \in D_z^{agg}$, we cannot use the above approach. Here, we use Prop. 6.4.3. According to this, for any state $\boldsymbol{\beta}$ with distance $\nabla_{\boldsymbol{\beta}}$, we need to differentiate the equation capturing $\mathbb{E}_t\,[\,\langle C_t, C_r\rangle \mid \boldsymbol{\beta}\,]$ up to order $\nabla_{\boldsymbol{\beta}}$ and then evaluate at t_0. We illustrate the evaluations for the states $\langle 1,1,0\rangle$. The calculations for other states are similar.

Given that the state $\langle 1,1,0\rangle$ has distance$\nabla_{\langle 1,1,0\rangle} = 1$, the initial value $\frac{d\,\mathbb{E}_t[\,\langle C_t, C_r\rangle | \langle 1,1,0\rangle\,]}{d\,t}\Big|_{t_0}$ is derived by differentiating Eq.(C.2) once with respect to time and then evaluating it at t_0. By performing these steps, we get:

$$\begin{aligned}
&(-2rs)\cdot(3r_s + r_b + r_l)\cdot \mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \langle 1,1,0\rangle\,] + (2r_s)\cdot \frac{d\,\mathbb{E}_t\,[\,\langle C_t, C_r\rangle \mid \langle 1,1,0\rangle\,]}{d\,t}\Big|_{t_0} = \\
&\quad +(-2r_s)\cdot(r_l + r_b + r_s)\cdot \mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \langle 1,1,0\rangle\,] \\
&\quad +(2r_s)\cdot(-2r_s)\cdot \mathbb{E}_{t_0}[\,\langle C_t, C_r\rangle \,|\, \langle 2,0,0\rangle\,] \\
&\quad +(2r_s)\cdot \frac{d\,\mathbb{E}_t\,[\,\langle C_t, C_r\rangle \mid \langle 2,0,0\rangle\,]}{d\,t}\Big|_{t_0} \\
&\quad +(-2r_s)\cdot(2r_s)\cdot\langle +1,-1\rangle \\
&\quad +(2r_s)\cdot r_t \cdot \langle -1,+1\rangle \cdot \mathbb{E}_{t_0}[C_t \,|\, \langle 1,1,0\rangle\,]
\end{aligned}$$

By simplifying this equation, we derive the values shown in Table 7.6.

Appendix D

Higher-Order Conditional Moments

D.1 Proof of Proposition 8.3.1

The following transformation is used within the derivation of the conditional expectations and higher-order moments:

$$
\begin{aligned}
&\frac{d}{d\,t}\Big(\ \mathbb{E}_t\left[T(\boldsymbol{\xi}^l,t)\mid\boldsymbol{\gamma}\right]\cdot\mathbb{P}(\boldsymbol{\gamma})\ \Big)\\
&=\ \frac{d}{dl\,t}\Big(\sum_{\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}} T(\boldsymbol{\xi}_i^l,t)\cdot\underbrace{\mathbb{P}_t(\boldsymbol{\xi}_i^l\mid\boldsymbol{\gamma})\cdot\mathbb{P}_t(\boldsymbol{\gamma})}_{\mathbb{P}_t(\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle)}\Big)\\
&=\ \frac{d}{\partial\,t}\Big(\sum_{\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}} T(\boldsymbol{\xi}_i^l,t)\cdot\mathbb{P}_t(\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle)\Big)\\
&=\ \sum_{\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}}\frac{d}{\partial\,t}\Big(T(\boldsymbol{\xi}_i^l,t)\cdot\mathbb{P}_t(\langle\boldsymbol{\xi}_i^s,\boldsymbol{\xi}_i^l\rangle)\Big)=\\
&=\ \sum_{\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle\in\mathbb{Y}_{\boldsymbol{\gamma}}}\left(\frac{d}{\partial\,t}\Big(T(\boldsymbol{\xi}_i^l,t)\Big)\cdot\mathbb{P}_t(\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle)+T(\boldsymbol{\xi}_i^l,t)\cdot\frac{d}{\partial\,t}\Big(\mathbb{P}_t(\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle)\Big)\ \right)\\
&=\ \mathbb{E}_t\left[\frac{d\,T(\boldsymbol{\xi}^l,t)}{dt}\mid\boldsymbol{\gamma}\right]\cdot\mathbb{P}(\boldsymbol{\gamma})\ +\sum_{S_i=\langle\boldsymbol{\gamma},\boldsymbol{\xi}_i^l\rangle} T(\boldsymbol{\xi}_i^l,t)\ \ \cdot\ \ \frac{d\,\mathbb{P}(S_i)}{d\,t}
\end{aligned}
\tag{D.1}
$$

D.2 Proof of Lemma 8.3.3

$$\frac{d\ \left(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}}}{d\ t} = \prod_{(H,C),H\in\mathcal{G}_l(\mathbb{M}),C\in ds^*(H)} \left(\xi(H,C) - \mu_{(H,C)}(\boldsymbol{\gamma},t)\right)^{I_{(H,C)}}$$

Using the rule for the differentiation over products, we derive:

$$\begin{aligned}
\frac{d\ \left(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}}}{d\ t} = &\Bigg[\left(-I_1 \cdot \left(\frac{d\ \mu_{(H_1,C_1)}(\boldsymbol{\gamma},t)}{d\ t}\right) \cdot \left(\xi(H_1,C_1) - \mu_{(H_1,C_1)}(\boldsymbol{\gamma},t)\right)^{I_1-1}\right) \\
&\left(\xi(H_2,C_2) - \mu_{(H_2,C_2)}(\boldsymbol{\gamma},t)\right)^{I_2} \\
&\vdots \\
&\left(\xi(H_n,C_n) - \mu_{(H_n,C_n)}(\boldsymbol{\gamma},t)\right)^{I_n} \Bigg] \\
+ &\Bigg[\left(\xi(H_1,C_1) - \mu_{(H_1,C_1)}(\boldsymbol{\gamma},t)\right)^{I_1} \\
&\left(-I_2 \cdot \left(\frac{d\ \mu_{(H_2,C_2)}(\boldsymbol{\gamma},t)}{d\ t}\right) \cdot \left(\xi(H_2,C_2) - \mu_{(H_2,C_2)}(\boldsymbol{\gamma},t)\right)^{I_2-1}\right) \\
&\vdots \\
&\left(\xi(H_n,C_n) - \mu_{(H_n,C_n)}(\boldsymbol{\gamma},t)\right)^{I_n} \Bigg] \\
+ &\vdots \\
+ &\Bigg[\left(\xi(H_1,C_1) - \mu_{(H_1,C_1)}(\boldsymbol{\gamma},t)\right) \\
&\left(\xi(H_2,C_2) - \mu_{(H_2,C_2)}(\boldsymbol{\gamma},t)\right) \\
&\vdots \\
&\left(-I_n \cdot \left(\frac{d\ \mu_{(H_n,C_n)}(\boldsymbol{\gamma},t)}{d\ t}\right) \cdot \left(\xi(H_n,C_n) - \mu_{(H_n,C_n)}(\boldsymbol{\gamma},t)\right)^{I_n-1}\right) \Bigg] \\
= &- \sum_{\substack{\xi(H,C)\in\boldsymbol{\xi}^l \\ H\in\mathcal{G}_l(\mathbb{M}) \\ C\in ds^*(H)}} I_{(H,C)} \cdot \frac{d\ \mu_{(H,C)}(\boldsymbol{\gamma},t)}{d\ t} \left(\boldsymbol{\xi}^l - \mu(\boldsymbol{\gamma},t)\right)^{\mathbf{I}-\mathbf{e}_{H,C}} \qquad \text{(D.2)}
\end{aligned}$$

D.3 Proof of Lemma 8.3.2

In this proof, we express the multiplication of distance functions and apparent rate functions in terms of the former and the conditional expectations using Taylor expansion:

$$
\begin{aligned}
&\sum_{S_k=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_k^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\mathbf{I}} \cdot r_\alpha(\boldsymbol{\xi}_k^l) \cdot \mathbb{P}_t(\boldsymbol{\xi}_k^l \mid \boldsymbol{\gamma}) = \\
&\sum_{S_k=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_k^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\mathbf{I}} \Bigg(r_\alpha(\, \mu(\boldsymbol{\gamma}, t)\,) \\
&+ \sum_{(H, C_x) \in \xi^l} \frac{\partial\, r_\alpha(\, \mu(\boldsymbol{\gamma}, t)\,)}{\partial\, \xi(H, C_x)} \,(\, \xi(H, C_x) - \mu_{(H, C_x)}(\boldsymbol{\gamma}, t)\,) \Bigg) \cdot \mathbb{P}_t(\boldsymbol{\xi}_k^l \mid \boldsymbol{\gamma}) \mathbb{P}_t(\boldsymbol{\xi}_k^l \mid \boldsymbol{\gamma}) \\
&= \sum_{S_k=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_k^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\mathbf{I}} r_\alpha(\, \mu(\boldsymbol{\gamma}, t)\,) \cdot \mathbb{P}_t(\boldsymbol{\xi}_k^l \mid \boldsymbol{\gamma}) \\
&+ \sum_{S_k=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_k^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\mathbf{I}} \sum_{(H, C_x) \in \xi^l} \frac{\partial\, r_\alpha(\, \mu(\boldsymbol{\gamma}, t)\,)}{\partial\, \xi(H, C_x)} \,(\, \boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma}, t)\,)^{e_{(H, C_x)}} \cdot \mathbb{P}_t(\boldsymbol{\xi}_k^l \mid \boldsymbol{\gamma}) \\
&= \sum_{S_k=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_k^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma}, t) \Big)^{\mathbf{I}} r_\alpha(\, \mu(\boldsymbol{\gamma}, t)\,) \cdot \mathbb{P}_t(\boldsymbol{\xi}_k^l \mid \boldsymbol{\gamma}) \\
&+ \sum_{S_k=\langle \boldsymbol{\gamma}, \boldsymbol{\xi}_k^l \rangle \in \mathbb{Y}_{\boldsymbol{\gamma}}} \sum_{(H, C_x) \in \xi^l} \frac{\partial\, r_\alpha(\, \mu(\boldsymbol{\gamma}, t)\,)}{\partial\, \xi(H, C_x)} \Big(\boldsymbol{\xi}_k^l - \mu(\boldsymbol{\gamma}, t) \Big)^{I + e_{(H, C_x)}} \cdot \mathbb{P}_t(\boldsymbol{\xi}_k^l \mid \boldsymbol{\gamma})
\end{aligned}
\tag{D.3}
$$

Appendix E

Analysis of Client-Server System Using Higher-Order Moments

E.1 Covariance of the State Variables Related to Clients

In this model, the variables C_t and C_r are strongly correlated. At any given time t: $C_t + C_r = 150$. Therefore, when calculating the conditional expectations, we have:

$$\forall\, \boldsymbol{\beta} \in D_{CS}^{agg} \;:\; \mathbb{E}_t\left[C_r \mid \boldsymbol{\beta}\right] \;=\; \mathbb{E}_t\left[150 - C_t \mid \boldsymbol{\beta}\right] \;= 150 - \mathbb{E}_t\left[C_t \mid \boldsymbol{\beta}\right] \tag{E.1}$$

This is used to identify the relationship between variances related to the variables C_r and C_t:

$$\begin{aligned}
\mathbb{VAR}_t\left[C_r \mid \boldsymbol{\beta}\right] &= \mathbb{E}_t\left[\left(C_r - \mu_{(C_r)}(\boldsymbol{\beta},t)\right)^2 \mid \boldsymbol{\beta}\right] \\
&= \mathbb{E}_t\left[\left((150 - C_t) - (150 - \mu_{(C_t)}(\boldsymbol{\beta},t))\right)^2 \mid \boldsymbol{\beta}\right] \\
&= \mathbb{E}_t\left[\left(C_t - \mu_{(C_t)}(\boldsymbol{\beta},t)\right)^2 \mid \boldsymbol{\beta}\right] \;=\; \mathbb{VAR}_t\left[C_t \mid \boldsymbol{\beta}\right]
\end{aligned}$$

Furthermore, the relationship between the variances of C_t and C_r and their covariance is identified as:

$$\begin{aligned}
\boldsymbol{M}_{\langle 1,1\rangle}(\boldsymbol{\beta},t) &= \mathbb{E}_t\left[(C_t - \mu_{(C_t)}(\boldsymbol{\beta},t))(C_r - \mu_{(C_r)}(\boldsymbol{\beta},t)) \mid \boldsymbol{\beta}\right] \\
&= \mathbb{E}_t\left[((150 - C_r) - (150 - \mu_{(C_r)}(\boldsymbol{\beta},t)))(C_r - \mu_{(C_r)}(\boldsymbol{\beta},t)) \mid \boldsymbol{\beta}\right] \\
&= -\,\mathbb{E}_t\left[(C_r - \mu_{(C_r)}(\boldsymbol{\beta},t))^2 \mid \boldsymbol{\beta}\right] \\
&= -\,\mathbb{VAR}_t\left[C_r \mid \boldsymbol{\beta}\right] = -\mathbb{VAR}\left[C_t \mid \boldsymbol{\beta}\right]
\end{aligned}$$