Robust Decision-Making with Model Uncertainty in Aerospace Systems

by

Luca Francesco Bertuccelli
Masters of Science in Aeronautical and Astronautical Engineering
Massachusetts Institute of Technology, 2004
Bachelor of Science in Aeronautical and Astronautical Engineering
Purdue University, 2002

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
September 2008

Author .....
Department of Aeronautics and Astronautics
August 11, 2008

Certified by .....
Jonathan P. How
Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by .....
Nicholas Roy
Assistant Professor of Aeronautics and Astronautics

Certified by .....
Francis Carr
The Charles Stark Draper Laboratory Inc.
Technical Supervisor

Accepted by .....
David L. Darmofal
Associate Department Head
Chair, Committee on Graduate Students

Robust Decision-Making with Model Uncertainty in Aerospace Systems

by

Luca Francesco Bertuccelli

Submitted to the Department of Aeronautics and Astronautics
on August 11, 2008, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics

Abstract

Actual performance of sequential decision-making problems can be extremely sensitive to errors in the models, and this research addressed the role of robustness in coping with this uncertainty. The first part of this thesis presents a computationally efficient sampling methodology, Dirichlet Sigma Points, for solving robust Markov Decision Processes with transition probability uncertainty. A Dirichlet prior is used to model the uncertainty in the transition probabilities. This approach uses the first two moments of the Dirichlet to generates samples of the uncertain probabilities and uses these samples to find the optimal robust policy. The Dirichlet Sigma Point method requires a much smaller number of samples than conventional Monte Carlo approaches, and is empirically demonstrated to be a very good approximation to the robust solution obtained with a very large number of samples.

The second part of this thesis discusses the area of robust hybrid estimation. Model uncertainty in hybrid estimation can result in significant covariance mismatches and inefficient estimates. The specific problem of covariance underestimation is addressed, and a new robust estimator is developed that finds the largest covariance admissible within a prescribed uncertainty set. The robust estimator can be found by solving a small convex optimization problem in conjunction with Monte Carlo sampling, and reduces estimation errors in the presence of transition probability uncertainty. The Dirichlet Sigma Points are extended to this problem to reduce the computational requirements of the estimator.

In the final part of the thesis, the Dirichlet Sigma Points are extended for real-time adaptation. Using insight from estimation theory, a modified version of the Dirichlet Sigma Points is presented that significantly improves the response time of classical estimators. The thesis is concluded with hardware implementation of these robust and adaptive algorithms on the RAVEN testbed, demonstrating their applicability to real-life UAV missions.

Thesis Supervisor: Jonathan P. How
Title: Professor of Aeronautics and Astronautics

Acknowledgments

Firstly, I'd like to thank my advisor Prof. Jonathan How for allowing me to explore a new and interesting series of problems that have formed the core part of this thesis. Many thanks go to Dr. Francis Carr for listening to my ideas and brainstorming with me, as well as providing many interesting thoughts and neat questions throughout the course of this thesis. My great thanks go to Prof. Nicholas Roy for having provided a good source of inputs, particularly in the motivation and understanding of the robustness work of this thesis. My readers, Prof. Hamsa Balakrishnan and Dr. Louis Breger, have provided me with a good array of thoughts and questions that have improved the presentation of the work in this thesis.

My research experience was greatly shaped at Purdue University by my early work with Prof. James Garrison, and I am deeply appreciative of this formal introduction to research. I am grateful for my conversations with Dr. Thomas Buter on my decision to pursue my professional and personal decisions.

Many of my fellow labmates the Aerospace Control Laboratory have been a good source of ideas, and in particular for the work in the past year, I would like to especially acknowledge Han-Lim Choi and Brett Bethke. Han-Lim provided very useful comments in the development of the robust hybrid estimation of Chapter 3. I am grateful to Brett for having provided a sounding board for some of my ideas, being a wealth of knowledge of a lot of much of the material in this work, and just being a great person to work with. The help of Kathryn Fischer in trip arrangements and logistics is also greatly appreciated.

I am grateful to my "grandparents from Purdue", Alfonso and Fernanda Corazza for lending supportive phone calls, postcards and letters. Since finishing my MS, I have had the great fortune of meeting Jennifer Gala, who has unfailingly provided support and encouragement every step of the way. I especially want to thank my family, my father Claudio, my mother Alessandra, my brothers Iacopo Matteo and Marco Antonio, and my grandmother Fernanda for their constant love and support.

To the (too many) friends and colleagues not singled out in these acknowledg-

ments, please know that your thoughts, conversations, encouragement have always helped provide a source of motivation for me to do my best, and wish to say a heartfelt thank you for all your support.

To my family, in particular to my grandfather Carlo Milea (1920-2002)

This research was funded in part under Air Force Grant # F49620-01-1-0453. The testbed were funded by DURIP Grant # F49620-02-1-0216. The author also gratefully acknowledges funding from the ASEE NDSEG and one year of support as a Draper Fellow from Draper Labs.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Decision-Making Systems

Many modern day aerospace systems, such as Unmanned Aerial Vehicles (UAVs), require an increasing level of autonomy. While UAVs are currently primarily piloted by human operators, future systems are expected to autonomously (or at least semi-autonomously) acquire information, process the observations, and come up with optimal decisions.

In the context of UAV applications, autonomous decision-making is a very challenging problem [6, 7, 14, 17, 20, 28, 83]. Autonomous agents will be flying over fairly unstructured and dynamic environments, whose true state can only be indirectly inferred from noisy observations. A hierarchy of decisions systems comes into play, ranging from low-level path planning algorithms (that, for example, control the altitude or airspeed of a particular vehicle), to more high-level task allocation algorithms, that decide which vehicle should be allocated to which region of the environment, or what optimal strategy should be used to accomplish a particular objective.

There are numerous active areas of interest in decision systems for autonomous systems. For example, computational issues have been, and still are, a very important area of research. Real-time implementations of task assignment algorithms developed in the Operations Research community are seeing applications in the UAV community [1, 2, 6, 11]. While smaller problems can typically be solved in adequate time for

real-time implementation, scaling up the size of these problems results in potential computational difficulties that limits the implementation of these decision algorithms in real systems. Adding the complexities of target motion makes this an increasingly challenging problem. Furthermore, higher-level control algorithms (such as those formulated as Markov Decision Processes) are confronted with the so-called "curse of dimensionality", which results in significant computational difficulty as the problem size increases [8].

Another important area of research is that of the role of communication of the agents in the decision-making process [1, 22, 70, 71]. For example, given a group of distributed agents connected by an intermittent communication network, the issues of which information needs to be passed to which agent and when, is still an active topic of research [22]. Including the additional realities of noisy transmissions, coupled with noisy observations, make this a particularly challenging problem. Furthermore, questions such as "what is the minimal information that should be transmitted in order to satisfactorily achieve a desired objective?", are still an open problem that attracts the attention of many researchers.

Yet another important area of research is that of the role of uncertainty in the modeling assumptions of more complex decision systems such as Markov Decision Processes. Decision systems generally rely on accurate modeling of the uncertainty in order to achieve the desired optimal performance, and minor deviations in these model parameters can lead to suboptimal performance. Due to importance of performance loss, model uncertainty in decision systems forms the key theme of this thesis where the emphasis is on a family of decision systems driven by a Markov Chain.

This class of systems is of general interest, as it forms the core of popular Markov Decision Process- (MDP-) based control approaches extensively used in a large variety of planning problems. MDP-based approaches have increasingly been applied in the aerospace community, and active research is being done in the computational challenges associated with these control problems [14, 83]. Within this class of Markov Chain-based decision systems, we also include a family of estimation problems known as stochastic hybrid estimation, where the Markov Chain is one of the model parame-

ters. Hybrid estimation provides a framework for estimating the state of a broad class of systems, and the outputs of the estimators such as state estimates and covariances are used as inputs to decision systems (such as the afore-mentioned task allocation).

While the transition probabilities of the Markov Chain model an intrinsic uncertainty in the state dynamics of the system, the probabilities themselves are the outcome of a separate estimation process, and are likely to be uncertain. It has been previously shown by other authors that the poor knowledge of the transition probabilities of the Markov Chain can degrade optimal performance of decision systems systems [3, 43, 69]. In some of the systems that we will investigate in this thesis, performance loss of 50% may not be uncommon in the presence of uncertainty in these model parameters.

An important item to note is that the problems of interest in this thesis are known to suffer the so-called "curse of dimensionality" [8], making real-time solutions for large systems an active topic of current research. In the estimation of multiple model systems, the optimal multiple model estimator cannot be implemented in real-life due to memory storage requirements. Thus, one resorts to suboptimal approximations for the estimation of these models. As a result, in accounting for the model uncertainty, one must take great care to not additionally increase the solution time of the robust counterpart of these problems.

1.2 Summary of Contributions

The emphasis of the first two chapters is to properly account for, and hedge against, errors in the transition probabilities in control and estimation frameworks. While the commonality between these frameworks is the uncertainty in the transition probabilities of the Markov Chain, the systems are fundamentally different in how the uncertainty impacts the overall performance. The final chapter discusses a technique for adapting to the Markov Chain via online measurements and has a slightly different objective from the first two chapters. The goal is to learn the transition probabilities efficiently, rather than solely being robust to model uncertainty.

One way of accounting for the uncertainty in the transition probabilities is to take a Bayesian approach, and generate samples (or scenarios) from the prior distribution on these probabilities, and use these samples to find the robust policy. As this is generally a computationally intensive task, we extend the work in robust Markov Decision Processes by presenting a new sampling-based algorithm that requires far fewer scenarios than conventional algorithms by exploiting the first two moments of the distribution.

We also show that transition probability uncertainty can degrade performance in estimation problems. In particular, this thesis demonstrates that transition probability uncertainty can generate mismatched covariances, which in turn leads to significant estimation errors. One key problem is covariance underestimation, where the estimator is overly confident of its estimates. We formulate a robust counterpart to classical multiple model estimation algorithms, and mitigate the overconfidence phenomenon.

A complementary strategy to incorporating the uncertainty using robust techniques, is to account for model uncertainty reduction by incorporating real-time observations [76]. We conclude this thesis with some insight into adaptation mechanism for the uncertain transition probabilities.

The individual contributions are as follows. In the area of robust decision-making, this thesis presents:

- An algorithm that precisely defines the model uncertainty in terms of *credibility regions*, using the Dirichlet prior to model the uncertain transition probabilities. This bisection algorithm is used in conjunction with Monte Carlo sampling, and can efficiently find the credibility region used in the robust MDP;

- A new sampling-based algorithm using Dirichlet Sigma Points for finding approximate solutions to robust MDPs in a computationally tractable manner. We prove that the Dirichlet Sigma Points are proper samples of a probability vector (summing to unity, and between 0 and 1) and can therefore be used in general sampling-based algorithms. By using Dirichlet Sigma Points, we significantly reduce the total number of samples required to find the robust solution,

while achieving near optimal performance;

- Guidelines for choosing the tuning parameter used in the Dirichlet Sigma Points, and provides numerical results demonstrating the reduction in samples required for the robust solution. In particular we show results in a machine repair problem, and autonomous agent planning.

In the area on multiple model estimation, this thesis:

- Addresses the issue of uncertain transition probabilities in multiple model estimators. In particular, we extend the work of Refs. [27, 46] and identify the problem of covariance mismatch due to the uncertain Markov Chain;

- Provides a robustness framework for generating robust estimates and covariances. In tracking applications, one of the main problems of covariance mismatch is the problem of covariance underestimation, in which the estimator is more confident about its state estimates than it should be, and can result in an increased estimation error. Our robustness framework ensures that the covariance is not underestimated, and is able to maintain a low estimation error;

- Shows reduction in estimation error in two aerospace tracking problems: the first one is a UAV multi-target tracking problem, and the second an agile target tracking problem.

The section on Markov Chain adaptation discusses a method of learning the transition probabilities of the Markov Chain. In particular:

- An explicit recursion is derived for the mean and variance of the transition probabilities under a Dirichlet prior, and uses this formulation to identify the cause of the slow learning of the Markov Chain;

- A new estimator is derived that introduces the notion of an effective process noise to speed up the transition probability identification problem, and has links to measurement fading techniques;

- Numerical examples are presented that demonstrate the faster adaptation of the transition probabilities using the new estimator. This new estimator is also demonstrated in the context of real-time MDP re-planning where the optimal reward is collected almost twice as quickly as conventional adaptation algorithms.

Finally, we implement the robust and adaptive group of algorithms in our lab's multi-vehicle testbed. In particular, we demonstrate that our algorithms can significantly extend a mission's lifetime by allowing vehicles to perform robust missions, and quickly adapt to changes in the environmental conditions. More concretely, these new algorithms reduce the number of vehicle crashes that occurred in the presence of transition probability uncertainty, thereby extending overall mission effectiveness.

Chapter 2

Decision Processes with Model Uncertainty

This first chapter addresses the impact of model uncertainty in a general class of decision-making problems known as Markov Decision Processes (MDPs). It has been previously shown that MDPs are sensitive to uncertainty in the transition probabilities of the underlying Markov Chain and that this uncertainty can significantly degrade optimal performance.

This chapter presents several contributions that build on the work of other authors in the field of robust MDPs. Previous work has primarily presented uncertainty sets described by ellipsoidal models or polytopic descriptions of the uncertainty in the transition probabilities. In some cases it might not be possible to construct a priori bounds on the transition probabilities (as in the case of polytopic uncertainty), and we therefore use a Dirichlet prior distribution on the transition probabilities. Importantly, the Dirichlet prior can be described compactly with a small number of parameters. Using the Dirichlet prior, the analogue of the uncertainty region becomes a *credibility region.* Unfortunately, the credibility region for the Dirichlet cannot be found in closed form and we present an efficient bisection algorithm that, in conjunction with Monte Carlo sampling, can successfully identify this region. These samples amount to realizations of the uncertain transition probabilities, and the samples within this credibility region are then used in a scenario-based optimization to

find robust MDP policies. The key benefit of using a sample-based robust MDP approach is that it only requires minimal modification of standard solution methods for nominal MDPs, and hence many systems can be easily modified to account for robustness.

Using the samples from the credibility region to find the robust MDP policy is computationally expensive as this approach requires a very large number of samples. Little work has been done in the context of robust MDPs to address this computational issue. The main contribution of this chapter is a new algorithm that reduces the total number of samples by introducing Dirichlet Sigma Points. The Dirichlet Sigma Points are deterministically chosen samples that are selected by using the first two moments of the Dirichlet, and are used to approximate the uncertainty in the transition probabilities. We present some numerical results demonstrating the implementation of the Dirichlet Sigma Points, and highlight the reduction in total scenarios required to obtain the robust solution. Guidance is also provided on the selection of the tuning parameter for the Dirichlet Sigma Points.

2.1 Introduction

2.1.1 Previous work

Markov Decision Processes can be quite sensitive to the transition probabilities of the underlying Markov Chain, and there has been a lot of work that has addressed this issue [3, 43, 51–53, 55, 69, 79]. In particular, this body of literature has identified the sensitivity of the MDP to the transition probabilities, and researchers have developed optimal solutions robust to errors in the transition probabilities.

The work of Satia [79] considered the on-line identification of the state transition matrix by observing the system's transitions across the states and updating the model for the transition matrix with these observations. The work of Kumar et al. [51–53] considered the problem of controlled Markov Chains, when the state transition matrix governing the chain was unknown. An additional term in the objective function was

added to account for the identification of the transition probabilities.

More recent work (e.g., [3, 43, 55, 69, 86]) incorporates the uncertainty in the state transition matrix directly in the MDP formulation and finds policies that are both optimal in minimizing the cost and robust to errors in the optimization parameters. In particular, Nilim [69] considers both finite and infinite horizon problems, and derives a robust counterpart to the well-known Value Iteration (VI) algorithm. Nilim and El Ghaoui [69] also present numerous uncertainty formulations that can be used very efficiently with Robust VI. Other approaches have also proposed techniques for adaptively identifying the state transition matrix online [18, 42, 80], but were not concerned with the robust problem. Poupart [74] has shown that Bayesian reinforcement learning can be expressed as a Partially Observable MDP (POMDP), and have presented their Beetle algorithm that can very efficiently adapt to online observations.

Recent work by Jaulmes et al. [44, 45], Mannor et al. [59] and Delage and Mannor [23] has also addressed the impact of uncertainty in multi-stage decision problems. The work by Jaulmes has addressed the uncertainty in the parameters of Partially Observable Markov Decision Processes (POMDPs). The solution method uses a direct sampling of the uncertain parameters in the MEDUSA (Markovian Exploration with Decision based on the Use of Sampled models Algorithm). Additional recent work by Mannor has investigated the issue of bias and variance in MDPs with poorly known transition probabilities. In particular, Mannor [59] discusses an analytical approximation to the mean and variance of the objective function of an infinite horizon MDP with uncertain parameters.

Delage [23] presents a percentile optimization approach as an attempt to mitigate the potential conservatism of robust solutions. The percentile optimization formulation addresses the variability in the optimal cost, and they show that solving a percentile optimization problem for an MDP with uncertain rewards results in a second order cone, while the more generic percentile optimization with general, uncertain transition model is shown to be NP-hard. Delage and Mannor approximate the (uncertain) value function using a second order approximation introduced in Mannor [59],

Table 2.1: Comparison of Some Uncertain MDP Formulations

Optimization	*Robust*	*Bayesian Optimization*	*Certainty Equivalence*
Formulation	Min-max	Probabilistic	Substitute best estimate
Used by	Nilim [69], Iyengar [43] Bagnell [3], Satia [79] White [86]	Delage, Mannor [23] Li, Paschalidis [58] Doshi, Roy [25] Jaulmes et al. [44, 45]	Doshi, Roy [26] Mannor [59]
Assumptions	Uncertainty set of transition model	Prior on transition model	Mean of transition model

but these results are only valid for a fixed policy. A summary of these formulations is shown in Table 2.1.

Nilim and El Ghaoui [69] presented an alternative approach to solving the robust MDP that used scenarios, but did not provide an explicit method for how these scenarios were generated. This motivates the following work, as we provide an explicit method for generating these scenarios, as well as formalizing the regions in a Bayesian sense by using credibility regions.

2.1.2 Outline

This chapter discusses the general decision making process formulation in Section 2.2, and the reliance on an accurate model. Model uncertainty is described in detail in Section 2.3, where we present an algorithm that precisely defines the model uncertainty in a Bayesian sense in terms of credibility regions. We then discuss the robustness approach to mitigate sensitivity to errors in Section 2.4 and use the results from the credibility region to develop a new scenario-based approach to robustness. In seeing that this scenario-based approach can be fairly computationally expensive, a new sampling algorithm with lower computational requirements is presented in Section 2.5. This new algorithm achieves the robust performance of other sampling algorithms, but requires much fewer samples to find the robust solution. We then apply this new algorithm to illustrative machine repair and robot planning problems in Sections 2.6 and 2.7.

2.2 Background

2.2.1 Markov Decision Processes

Finite state, finite action, discrete time Markov Decision Processes are defined in the following manner (see for example, Bertsekas and Puterman [10, 75]):

- **State:** The system state, i, is an element of all possible states $i \in \mathcal{X}$. The cardinality of the state space, N is denoted as $|\mathcal{X}|$

- **Action:** The decision maker at each decision epoch (time at which a decision is made) can choose a control input (action) $a_k \in \mathcal{A}$. The cardinality of the action space is denoted as $N_a = |\mathcal{A}|$. An optimal policy is defined as $u^* = [a_1^*, a_2^*, \ldots, a_T^*]$, where $a_k^* \in \Re^{N_a}$ is the optimal control action, and $a_k(i)$ is the optimal control in state i at time k

- **Transition model:** The transition model describes the (probabilistic) system dynamics Π^a, where π_{ij}^a describes the probability that the system will be in state j at the next time given that it was in state i at the previous time step, and action a was implemented[1]

- **Reward model:** The reward $g_k(i, a)$ is the value of being in state i at some time under action a at time k. The reward model can also be defined as $g_k(i, a, j)$ where this is the value of being in state i at the current time step, implementing action u, and transitioning to the next state j

- **Optimality criterion:** The optimality criterion is the desired objective, and can include maximizing the expected reward over a finite time horizon, or minimizing an expected cost. The optimization can be performed over a finite time T which constitutes the *time horizon* or an infinite time horizon

[1]Note that to maintain consistency in notation with the subsequent chapters, the transition model is denoted by Π^a. In many other texts, see for example Bertsekas [10], the set of admissible policies is denoted by Π.

- **Discount factor:** A discount factor $0 < \phi < 1$ is usually introduced to account for the fact that current costs or rewards have a higher weight than costs or rewards in the future

The transition model is more precisely defined as $\Pi^a \in \mathcal{R}^{N \times N} \quad \forall a$, given by

$$\Pi^a = \begin{bmatrix} \pi^a_{1,1} & \pi^a_{1,2} & \cdots & \pi^a_{1,N} \\ \pi^a_{2,1} & \pi^a_{2,2} & \cdots & \pi^a_{2,N} \\ \cdots & & & \\ \pi^a_{N,1} & \pi^a_{N,2} & \cdots & \pi^a_{N,N} \end{bmatrix}$$

whose (i, j)th entry describes the probability of being in state j at time $k + 1$, given the preceding state was i at the previous time step

$$\pi^v_{i,j} = \Pr[x_{k+1} = j \mid x_k = i, a_k = v] \tag{2.1}$$

Throughout this chapter, we consider the well-studied linear, additive utility of the form

$$J_T = g(i_T, a_T) + \sum_{k=0}^{T-1} g_k(i_k, a_k) \tag{2.2}$$

where $g_k(i_k, a_k)$ denotes the cost at time k for being in state i_k under action a_k, and $g(i_T, a_T)$ is the terminal cos. Our objective will be that of minimizing the expected cost as

$$\min_u \mathbf{E}\left[J_T\right] \tag{2.3}$$

Note that maximizing an expected reward in this stochastic setting is also fairly standard and can be solved using Dynamic Programming (DP). Alternative formulations of a more general nature are presented in the next section.

2.2.2 Alternative Formulations

While the linear, additive utility is a common objective, it does not take into account a user's risk aversion, and alternative formulations have been studied that do take into account this important criterion. For example, a user might be generally interested in finding the optimal policy that maximizes an expected reward, but the reward also has low variability. This gives rise to so-called *risk-sensitive* policies [21, 61]. The optimal policy in this case is a policy with lower expected reward, but much lower variance, than the optimal policy of Eq. 2.3. An example of such a risk-sensitive framework is shown below

$$\min_{u} \frac{1}{\gamma} \log \mathbf{E}\left[\exp^{\gamma J_T}\right] \tag{2.4}$$

where $\gamma > 0$ is a tuning parameter that reflects a user's risk aversion and by taking a Taylor series expansion for small values of γ, this formulation approximates a "mean-variance"-like expression of the form

$$\min_{u} \frac{1}{\gamma} \log \mathbf{E}\left[\exp^{\gamma J_T}\right] \approx \min_{u} \left[\mathbf{E}J_T + \gamma/2\Sigma_J\right] \tag{2.5}$$

where Σ_J indicates the variance of the cost J_T. Mannor [59] calls this the "internal variance" of the MDP. Note that when $\gamma \to 0$, this formulation results in the familiar linear additive utility of Eq. 2.2.

Coraluppi et al. [21] have shown that finite horizon risk-sensitive formulations satisfy a Dynamic Programming-like recursion and that Markov policies are optimal. However, the infinite horizon formulations in general may give rise to non-stationary policies [61], which may not be practical to implement. This issue is addressed by extending the horizon of the finite horizon problem and taking the limit to an infinite horizon. Coraluppi and Marcus [21, 61] also considered MDPs with partial observability.

An alternative optimization is the worst-case approach, where the optimization is

of the form

$$\min_{u} \max_{\mathcal{X}} [J_T] \tag{2.6}$$

This alternative formulation looks at the worst-case trajectory (or "sample-path" from the Markov Chain) that can occur with non-zero probability and that results in the worst possible reward. This model does not weigh the probability of this worst-case trajectory, and bears a close resemblance to the conservatism of classical robust control [90]. Coraluppi [21] showed some relationships between the risk sensitive formulations of Eq. 2.4 and Eq. 2.6.

In closing, these important formulations present more general optimizations to that of the linear additive utility, but like the linear additive utility formulation, assume that the transition probabilities of the Markov Chain are well known.[2] The issue of model uncertainty is addressed in the next section.

2.3 Model Uncertainty

2.3.1 Transition model uncertainty

In practice, the transition model Π^a is usually inferred from previous observations and the transition probabilities are themselves the outputs of a separate estimation process. For example, in a financial applications [42], the generators of the Markov Chain are derived from empirical observations of the state transition matrix. In an estimation context, Jilkov [46] and Doucet [27] identify the transition probabilities by observing online transitions. In the machine learning and Hidden Markov Model (HMM) community, learning the transition model through noisy observations is a common objective [76]. There are many models for describing the uncertainty in the transition model, and the more common ones are described in the next section.

[2]The worst-case approach actually only relies on the knowledge of the *structure* of the Markov Chain.

Polytopic Model

A classical approach for describing uncertainty in the transition probability is a polytopic model, that provides upper and lower bounds on the transition probability, where

$$\tilde{\Pi} = \left\{ \pi_{ij} \mid \pi_{ij}^{-} \leq \pi_{ij} \leq \pi_{ij}^{+} \right\} \tag{2.7}$$

and the lower and upper bounds (π_{ij}^{-} and π_{ij}^{+} respectively) are used to provide information on the admissible range of the probability. In addition, the appropriate row or column sum constraint of the transition probabilities is enforced.

Likelihood Model

An alternative description is a likelihood model, where

$$\tilde{\Pi} = \left\{ \pi_{ij} \mid \sum_{i,j} f_{ij} \log \pi_{ij} \geq \psi \right\} \tag{2.8}$$

where f_{ij} are the empirical frequencies of the state transitions, and ψ is a tuning parameter constrained such that $\psi < \sum_{i,j} f_{ij} \log f_{ij} \doteq \psi_{\max}$. ψ can be found via resampling methods [69], and is related to the Bayesian credibility regions we will discuss in the next sections.

A second order approximation to the log likelihood model is the ellipsoidal model, defined as

$$\tilde{\Pi} = \left\{ \pi \mid \sum_{j=1}^{N} \frac{(\pi_{ij} - f_{ij})^2}{f_{ij}} \leq \kappa^2, \quad \forall i \right\} \tag{2.9}$$

and κ is a constant that needs to be found. Again, for both example, the appropriate constraints for the probability must be enforced for π_{ij}.

Bayesian Approach

The approaches introduced previously, such as the polytopic approach, require knowledge of the constraints on the transition probabilities, and it may be unclear how to derive these constraints. An alternative approach is to provide a prior distribution on the transition probabilities. This approach is useful in that it does not require *hard* constraints (such as knowing a priori the bounds π_{ij}^- and π_{ij}^+). Also, depending on the choice of prior, this methods provides a rich class of follow on algorithms that learn, or adapt to, the transition probability.

In following this Bayesian approach, one assigns a prior f_D to the uncertain transition probabilities $\pi \sim f_D(\mathbf{p} \mid \alpha)$, where α is a vector of *hyperparameters*, or parameters that characterize the probability density f_D. This density is introduced next.

2.3.2 Dirichlet density

This thesis primarily uses the Dirichlet density to represent transition probability uncertainty.[3] The primary reasons for using the Dirichlet is that this choice of density implicitly accounts for the unit sum constraint on the rows of the probability transition matrix Π, and positivity constraints. Furthermore, the Dirichlet distribution is defined by hyper-parameters α_i that can be interpreted as counts, or times that a particular state transition was observed. By exploiting conjugacy [4] with the multinomial, this makes any measurement updates available in closed form. The Dirichlet prior has been applied frequently in the Artificial Intelligence literature [25, 26, 44, 45].

The Dirichlet f_D is a prior for the *row* of the transition matrix. That is, by defining $\mathbf{p} = \pi_{i,\cdot}$, we have $\mathbf{p} = [p_1, p_2, \ldots, p_N]^T$ and parameter (or prior counts) $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T$, is defined as

$$f_D(\mathbf{p}|\alpha) = K \prod_{i=1}^{N} p_i^{\alpha_i - 1}, \quad \sum_{i}^{N} p_i = 1, \quad 0 \leq p_i \leq 1 \tag{2.10}$$

[3] The Dirichlet density is the multi-dimensional extension to the Beta distribution [72].

[4] The conjugacy property ensure that if a Bayesian update is performed with a Dirichlet prior, and a multinomially distributed sequence of observations, the updated prior is a Dirichlet.

where $K = \frac{\Gamma(\sum_i \alpha_i)}{\prod \Gamma(\alpha_i)}$ is a normalizing factor that ensures the probability density integrates to unity. Two examples of the Dirichlet with different choices of hyperparameters are given in Figure 2-1.

2.3.3 Uncertainty set using the Dirichlet

While the Dirichlet density is characterized by the hyperparameters α_i for each row of the transition model, the density itself does not completely provide a precise notion of *uncertainty* in the row of the transition matrix. A more precise notion of the uncertainty is the idea of the *credibility region* [9, 19]. The credibility region is the Bayesian equivalent of a confidence region, and is formally defined as follows: a $100 \times \eta\%$ credibility region for parameter p is a subset $\mathcal{P}$ of P of the form

$$\mathcal{P} = \{p \in P \mid f_D(p \mid \alpha) \geq k(\eta)\}$$

and $k(\eta)$ is the largest constant such that

$$\int_{\mathcal{P}} f_D(p \mid \alpha) dp \geq \eta \tag{2.11}$$

In other words, given a prior $f_D(\alpha)$, the output is a credibility region $\mathcal{P}$, such that the overall mass of the density covers a $100 \times \eta\%$ region, such that the likelihood of the density achieves at least the threshold $k(\eta)$.

Two examples of the credibility regions are shown in Figure 2-2, for two different values of η, $\eta = 0.50$ and $\eta = 0.95$. The red line indicates the credibility region for a level of 50% and 95%. Note that as expected, as the credibility region increases, the area covered by the density fills a larger portion of the probability simplex.

The integration problem for the credibility region, unfortunately, cannot be solved in closed form for the Dirichlet density. Even for the simpler Beta density (a one-dimensional Dirichlet), it turns out that the credibility region $\mathcal{P}$ is a line segment

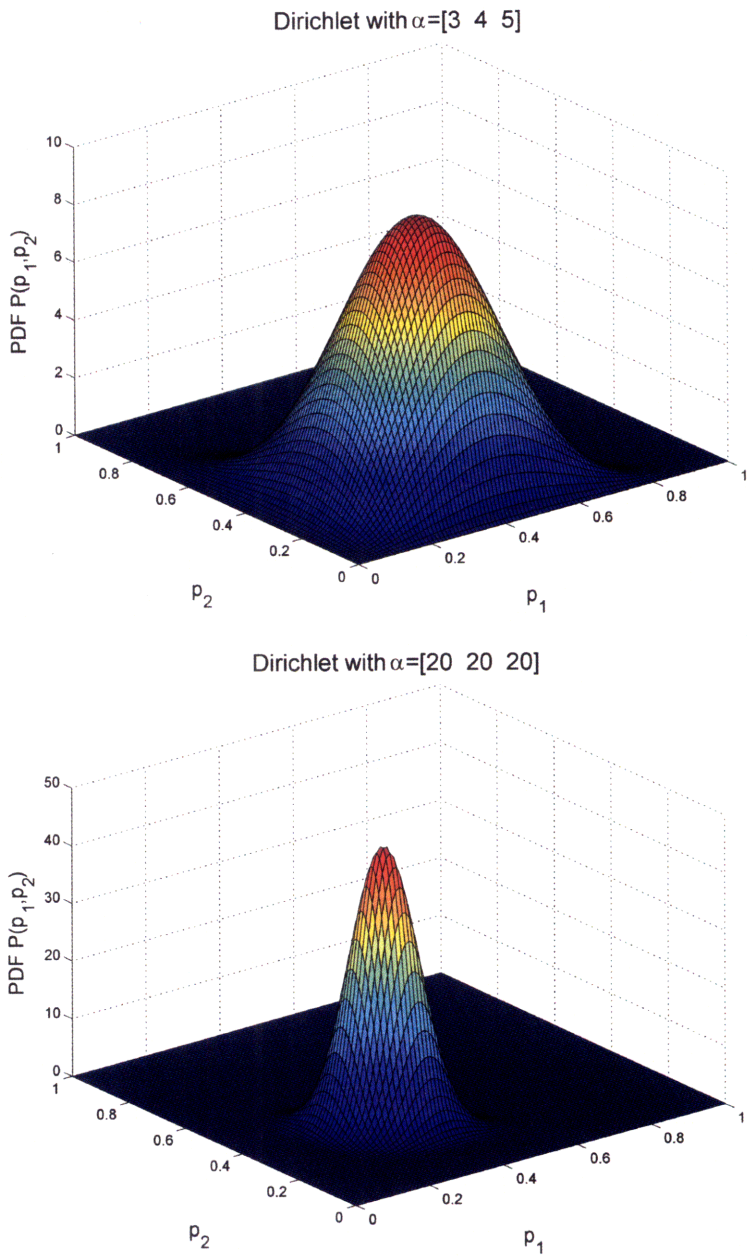


Fig. 2-1: Two different Dirichlet examples for (top) $\alpha = [3,\ 4\ ,5]$ and (bottom) $\alpha = [20,\ 20,\ 20]$

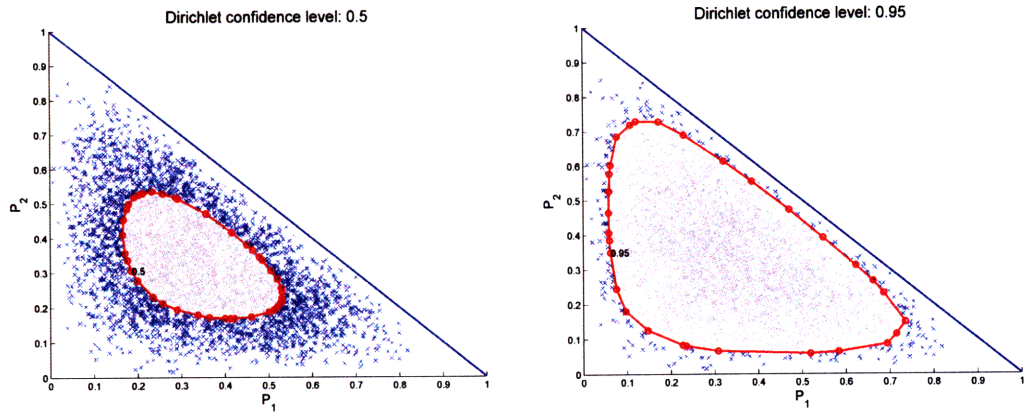


Fig. 2-2: Two different Dirichlet credibility regions for $\alpha = [3,\ 3,\ 3]$: (left) $\eta = 0.50$ and (right) $\eta = 0.95$. These regions were obtained by Monte Carlo sampling (see Section 2.3.5).

over p, and that the integration

$$\int_{\mathcal{P}} Kp^{\alpha_1}(1-p)^{\alpha_2}dp = \int_{p^-}^{p^+} Kp^{\alpha_1}(1-p)^{\alpha_2}dp \geq k(\eta) \tag{2.12}$$

can only be evaluated numerically. This is in fact the definition of the *incomplete* Beta function. Matlab for example, evaluates this by numerical gridding.[5] Even though the numerical gridding approach is fairly efficient for the Beta density, extending to higher dimensions like the Dirichlet becomes highly impractical, and computationally expensive. Common alternative techniques for bypassing the computational complexity of numerical grids is the use of Monte Carlo methods [19, 29]. We introduce these next, and discuss how to incorporate them in finding the credibility region for the Dirichlet.

2.3.4 Monte Carlo Methods

Monte Carlo methods can be used to efficiently approximate difficult integration problems. Our approach for finding the credibility regions relies crucially on the simplicity of generating samples from the Dirichlet. Since the Dirichlet is in the

[5] http://www.mathworks.com/

exponential family of distributions, it can be sampled very effectively with commercial software by normalizing samples obtained from the Gamma distribution. To generate Dirichlet samples from a density $f_D(\mathbf{p}|\alpha) = K \prod_{i=1}^{N} p_i^{\alpha_i - 1}$, one samples from the Gamma distribution with shape parameter α_i and scale factor of unity,

$$q_i \sim \text{Gamma}(\alpha_i, 1) \tag{2.13}$$

The Dirichlet sample is then given by $y_i = q_i / \sum_i q_i$. This corresponds to a single realization of the probability vector described the Dirichlet density.

2.3.5 Dirichlet Credibility Region Using Monte Carlo

Monte Carlo integration still does not provide any insight into how to ultimately find this region $\mathcal{P}$, as we still need to evaluate the difficult integral of Eq. 2.11. The basic idea of the approach is to use Monte Carlo sampling to generate realizations from the Dirichlet distribution, and approximate the integral over the entire credibility region, with a finite sum of Dirichlet samples y_i in the credibility region

$$\int_{\mathcal{P}} f_D(p \mid \alpha) dp \geq \eta \approx \sum_i \delta_i (f_D(y_i \mid \alpha) \geq \eta) \tag{2.14}$$

where $\delta_i(x)$ is an indicator function that is 1 if the argument x is true, and 0 otherwise. The additional requirement is that

$$\mathcal{P} = \{p \in P \mid f_D(y_i \mid \alpha) \geq k(\eta)\}$$

is satisfied for each of the samples. Unlike Chen [19], we will be using the samples to ultimately seek a robust solution in the next sections, and we do not know *a priori* what the value for $k(\eta)$ is. In order to find the value for $k(\eta)$, we employ a bisection algorithm to find the actual threshold $k(\eta)$. Our approach relies on the unimodal property of the Dirichlet density to find this credibility region using a bisection scheme.

Algorithm 2 Selecting samples within the Credibility Region

1: Provide an initial guess for lower bound $k^-(\eta)$, and upper bound $k^+(\eta)$ on the threshold
2: Define $k(\eta) = \frac{1}{2}(k^-(\eta) + k^+(\eta))$
3: Generate N_s samples $y_i, \quad \forall i = 1, \ldots, N_s$ for a Dirichlet prior $f_D(\mathbf{p} \mid \alpha)$
4: For all samples i, evaluate the density, and update the indicator function δ_i

$$\delta_i = \begin{cases} 1 & \text{If } f_D(y_i \mid \alpha) \geq k(\eta) \\ 0, & \text{else} \end{cases} \tag{2.15}$$

5: **if** $\frac{\sum_i \delta_i}{N_s} \geq \eta$ **then**
6: $\quad k^-(\eta) := (k^-(\eta) + k^+(\eta))/2$
7: **else**
8: $\quad k^+(\eta) := (k^-(\eta) + k^+(\eta))/2$
9: **end if**
10: **if** $|\frac{\sum_i \delta_i}{N_s}| - \eta \leq \epsilon$ **then**
11: $\quad$ Return $k(\eta)$ and δ_i
12: **end if**

The algorithm is initialized with an upper and lower bound on $k(\eta)$, and uses Monte Carlo simulation to generate a large number N_s of random samples of the Dirichlet density. Each of these samples is then checked to see whether or not they exceed the density $k(\eta)$ at the current time step. All the samples that exceed this threshold are then summed up, and if their total fraction exceeds the threshold, then there are too many samples, and the threshold $k(\eta)$ is reduced. If there are too few samples, the threshold is increased. The algorithm converges since the Dirichlet is unimodal, and the solution is unique.

These iterations are shown in Figures 2-3 and 2-4. The red line indicates the credibility region, the blue x denote the samples of the Dirichlet, and the gray x are all the samples that fall within the credibility region, which means that at convergence, 95% of the samples fall within this region.

2.4 Robustness in MDP

Now that we have an efficient method for calculating the uncertainty region given a Dirichlet density, we can move on to the problem of making robust decisions. The

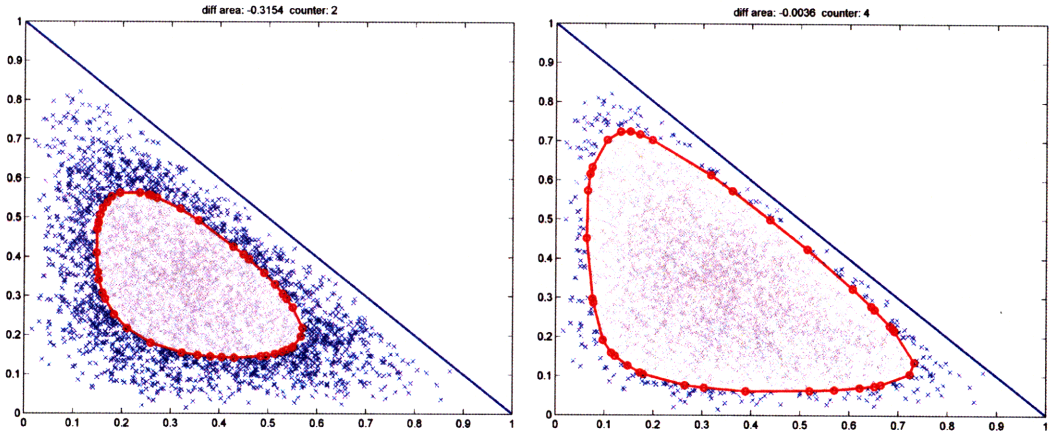


Fig. 2-3: Iterations 2 (left) and 4 (right) for finding the Dirichlet credibility regions (shown in red) for $\alpha = [3,\ 3,\ 3]$ using 1000 samples. The samples that fall within the credibility region are shown in gray, while the remaining samples are shown in blue.

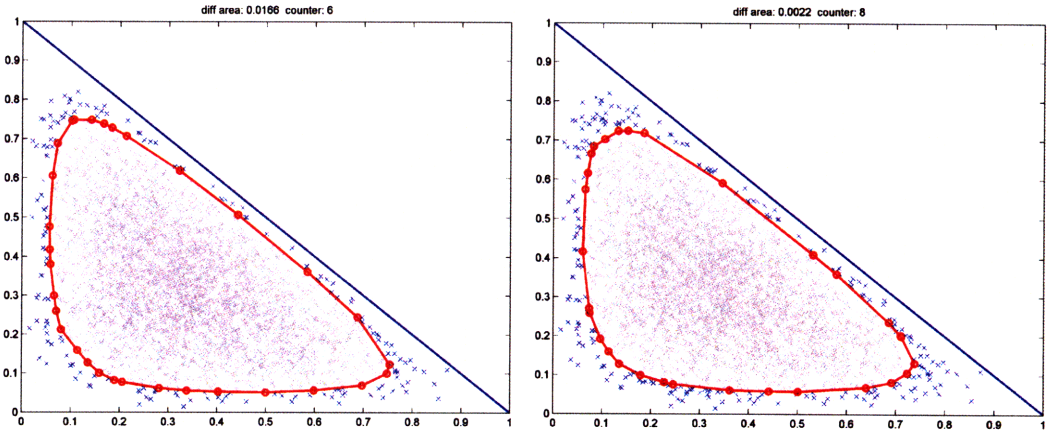


Fig. 2-4: Iterations 6 (left) and 8 (right) for finding the Dirichlet credibility regions for $\alpha = [3,\ 3,\ 3]$

previous section has discussed methods to quantify the level of uncertainty in the transition probabilities of the Markov Chain by using the Dirichlet density. The method of choice was a Monte Carlo method that samples from the uncertain transition probabilities. The idea in this section is to use this uncertainty set (and these samples) to find solution to robust Markov Decision Processes. First, the robust MDP is introduced, following from the precursor work of Nilim [69] and Iyengar [43].

2.4.1 Robustness

As we have stated earlier, in the presence of uncertainty in the optimization parameters, the optimal control policy u^* generated from incorrect parameters may no longer be optimal. Even if one had access to an estimator that could report the best estimates $\hat{\Pi}$ (in some maximum likelihood sense for example), simply replacing the uncertain parameters Π with their best estimates $\hat{\Pi}$ may lead to biased results [59]. Thus we introduce a robust counterpart to the nominal problem. The robust counterpart of Eq. (2.3) is defined as

$$\min_{u} \max_{\Pi \in \tilde{\Pi}} \mathbf{E}\left[J_u\right] \tag{2.16}$$

Like the nominal problem, the objective function is maximized with respect to the control policy; however, for the robust counterpart, the *uncertainty set* $\tilde{\Pi}$ for the transition matrix is given, rather than the actual state transition matrix Π for the nominal problem. The objective is then minimized with respect to the worst case realization of the transition matrix Π belonging to the uncertainty set $\tilde{\Pi}$. The robust policy is found from

$$u_R^* = \arg\min_{u} \max_{\Pi \in \tilde{\Pi}} \mathbf{E}\left[J_u\right] \tag{2.17}$$

Nilim and Iyengar show that robust Dynamic Programming [43, 69] can be used to solve for this robust policy. They also present robust dynamic programming and robust value/policy iteration counterparts to the classical (error-free) MDP formula-

tions for both finite and infinite horizon MDPs. These theorems are repeated below for convenience, and the proofs are in the references provided.

For the finite horizon problem, the following robust Dynamic Programming theory is provided.

Proposition 1 *(Robust Dynamic Programming [43, 69]) The robust control problem*

$$\min_{u} \max_{\Pi \in \tilde{\Pi}} \mathbf{E} \left(g(i_T, a_T) + \sum_{k=0}^{T-1} g_k(i_k, a_k) \right) \tag{2.18}$$

can be solved via the recursion

$$J_k(i) = \min_{a_k} \left(g(i_k, a_k) + \sigma_i^a(J_{k+1}) \right), \quad \forall i, \forall k \tag{2.19}$$

where $\sigma_i^a = \sup_{\pi \in \tilde{\Pi}} \pi J$ *is the support function over the uncertainty set* $\tilde{\Pi}$. *A corresponding optimal control policy is obtained by setting*

$$a_k^*(i) \in \arg_a \min \left(g(i_k, a_k) + \sigma_i^a(J_{k+1}) \right) \tag{2.20}$$

For the case of an infinite horizon, discounted cost objective, Nilim and Iyengar show that Value Iteration can be generalized to Robust Value Iteration in the case of an uncertain transition model, and is shown in the next algorithm

Proposition 2 *(Robust Value Iteration [43, 69]) The infinite horizon problem's value function with stationary uncertainty on the transition matrices, stationary control policies, and discounted cost function*

$$\min_{u} \max_{\Pi \in \tilde{\Pi}} \mathbf{E} \lim_{T \to \infty} \left(\sum_{k=0}^{T} \phi^k g_k(i_k, a_k) \right) \tag{2.21}$$

satisfies the optimality conditions

$$J(i) = \min_{a_k} \left(g(i, a) + \phi \sigma_i^a(J) \right) \tag{2.22}$$

where the value function $J(i)$ *is the unique limit value of the convergent vector se-*

quence defined by

$$J_k(i) = \min_{a_k} \left(g(i,a) + \phi \sigma_i^a(J_{k+1}) \right), \quad \forall i, \forall k \tag{2.23}$$

and the control action is found as

$$a^*(i) \in \arg_a \min \left(g(i,a) + \sigma_i^a(J) \right) \tag{2.24}$$

2.4.2 Computational Tractability

The solution times for the robust optimization of Eq. (2.16) are of course dependent on the number of stages in the problem, the dimension of the state, and the number of control actions. However, for the robust MDPs, these solution times also depend on the choice of the uncertainty model for the parameters.

Nilim [69] shows that uncertainty models such as the likelihood and polytopic models lead to computationally tractable optimizations. Scenario-based methods were also introduced in Nilim [69] as an alternate uncertainty model for the transition probabilities. In this uncertainty set, the decision-maker has access to, or can generate scenarios that form a scenario set Π_s that can then be used in performing the robust optimization of Eq. (2.16). This is similar to the random sampling from the MEDUSA approach [45]. Nilim shows that such a scenario-based optimization can also be solved with Robust Value Iteration, and requires only a marginal modification of standard value iteration.

Scenario-based approaches generally require tradeoff studies to determine the total number of simulations actually required to accurately represent the uncertainty in the transition probabilities. For example, in determining the credibility region alone, one must generate a large number of scenarios, but it is impractical to include all these scenarios in the robust optimization. There are as yet no known results that can a priori determine how many samples are sufficient. Some current results in a particle filtering framework [31] that rely on the Kullback-Leibler divergence as a distance measure to the underlying distribution indicate that these samples could be on the

order of 10^3. Thus, alternative sampling strategy must be investigated.

2.5 Sigma Point Sampling

The key problem in using scenario-based approaches is that there is no clear method for selecting *how many* scenarios are sufficient to obtain the robust solution; furthermore, this number tends to be quite large. As a consequence, one needs to either pick a *sufficiently* large number of samples, or come up with an algorithm to reduce the total number of scenarios required. In this section, we present a heuristic method to reduce the total number of scenarios, whose origins are in nonlinear estimation.

Julier et al. [47] developed Sigma Points as a deterministic sampling technique that selects *statistically relevant* samples to approximate a Gaussian distribution for nonlinear filtering problems. The Sigma Point algorithm is defined as follows for a Gaussian random vector $x \in \Re^N$. If the random vector $\mathbf{x}$ is normally distributed with mean $\bar{\mathbf{x}}_\mathbf{G}$ and covariance $\mathbf{R}_G \in \Re^{N \times N}$, $x \sim N(\bar{\mathbf{x}}_\mathbf{G}, \mathbf{R}_\mathbf{G})$, then the Sigma Points $\mathcal{M}_i$ are formed deterministically as follows

$$\begin{aligned}
\mathcal{M}_0 &= \bar{\mathbf{x}}_\mathbf{G}, \qquad w_0 = \kappa/(N+\kappa) \\
\mathcal{M}_i &= \bar{\mathbf{x}}_\mathbf{G} + \left(\sqrt{(N+\kappa)\mathbf{R}_G}\right)_i, \quad \forall i = 1, \ldots, N \\
\mathcal{M}_i &= \bar{\mathbf{x}}_\mathbf{G} - \left(\sqrt{(N+\kappa)\mathbf{R}_G}\right)_i, \quad \forall i = N+1, \ldots, 2N
\end{aligned}$$

The notation $(\sqrt{\mathbf{R}})_i$ denotes the i^{th} row of the square root matrix of $\mathbf{R}$. Each of the samples carries a weight $w_i = 1/(2(N+\kappa))$ and a tuning parameter κ is used to modify the level of uncertainty desired in the distribution.[6] For example, in the Gaussian case, a good heuristic [47] choice is $\kappa = 3 - N$. After these samples are

[6]The only requirement on the weights w_i is that they sum to 1, $\sum_i w_i$, but can otherwise be positive or negative.

propagated through a dynamic model, the posterior distribution can be recovered as

$$\begin{aligned}\bar{\mathbf{x}}^+ &= \sum_i w_i \mathcal{M}_i^+ \\ \mathbf{R}^+ &= \sum_i w_i (\mathcal{M}_i^+ - \bar{\mathbf{x}}^+)(\mathcal{M}_i^+ - \bar{\mathbf{x}}^+)^T \end{aligned} \tag{2.25}$$

where $\mathcal{M}_i^+$ are the Sigma Points propagated through the dynamic model.

2.5.1 Dirichlet Sigma Points

While the Sigma Points were originally developed in a Gaussian setting to reduce estimator divergence issues associated with linearization of the nonlinearity (hence, a completely different problem), our application is slightly different. Our objective is to *approximate* the Dirichlet with these Sigma Points, and in so doing, find a subset of statistically relevant scenarios that can capture the fundamental uncertainty in the transition probabilities. In other words, by using the first two moments $(\bar{\mathbf{p}}, \boldsymbol{\Sigma})$ of the Dirichlet, we have an expression for finding these *Dirichlet* Sigma Points as follows

$$\begin{aligned}\mathcal{Y}_0 &= \bar{\mathbf{p}} \\ \mathcal{Y}_i &= \bar{\mathbf{p}} + \beta_i \left(\sqrt{\boldsymbol{\Sigma}}\right)_i, \quad \forall i = 1, \ldots, N \\ \mathcal{Y}_i &= \bar{\mathbf{p}} - \beta_i \left(\sqrt{\boldsymbol{\Sigma}}\right)_i, \quad \forall i = N+1, \ldots, 2N \end{aligned}$$

where β_i is a tuning parameter we will discuss at length later, but has the same effect of the w_i for the Gaussian Sigma Points. $\bar{\mathbf{p}}$ and Σ are the mean and covariance of the Dirichlet.

In contrast to the Gaussian Sigma Points, the fact that the support of the Dirichlet is over the probability simplex requires that the following two statements must apply in order to use the Dirichlet Sigma Points:

1. The Dirichlet distribution must be well-approximated by a mean and a covariance.

2. The samples $\mathcal{Y}_i$ must satisfy the requirements of a probability vector, namely [72]:

$\sum \mathcal{Y}_i = 1$, and $0 \leq \mathcal{Y}_i \leq 1$

The first point is satisfied since the parameters α_i can be recovered from a set of Dirichlet-distributed random variables only using first and second moment information [88]. Furthermore, the mean and the variance of the Dirichlet are

$$\text{Mean:} \quad \bar{p} = \alpha_i/\alpha_0, \qquad \alpha_0 = \sum_i \alpha_i$$

$$\text{Variance:} \quad \Sigma(i,j) = \begin{cases} -\frac{\alpha_i \alpha_j}{\alpha_0^2(\alpha_0+1)}, & \text{if } i = j \\ \frac{\alpha_i(\alpha_0-\alpha_i)}{\alpha_0^2(\alpha_0+1)} \end{cases}$$

and Appendix B shows two approaches to recover the original Dirichlet from the first two moments.

The two-moment approximation is a very good approximation. We can show this by obtaining the mean and variance of the Dirichlet, and recover an estimate $\hat{\alpha}$ of the original parameters α_i from these moments using the technique of Appendix B. The absolute error, e, was normalized for each parameter

$$e = \frac{1}{N}\|\alpha_i - \hat{\alpha}_i\|$$

and the results are shown in Figure 2-5. Here 200 random parameters α_i were chosen for an increasing number of state dimensions: for the top figure, the parameters were chosen in the uniform interval $\alpha \in [2,\ 5]$, while in the bottom figure, the parameters were chosen in the uniform interval $\alpha \in [2,\ 50]$. The plots show the mean value (and 1-standard deviation) of the accuracy on the left axis, while they show the logarithm (base 10) of the error on the right axis. Even for small values for α, where the two-moment approximation may be less valid, the two-moment approximation still achieves a 4% error for the parameters, and in fact achieves less than a 1% error for state dimensions $N > 10$.

Thus, it remains to show that the Sigma Point samples in the case of a Dirichlet satisfy a probability vector subject to an appropriate choice of the weights w_i. The following propositions (whose proofs are in the Appendix) show that the Sigma Points

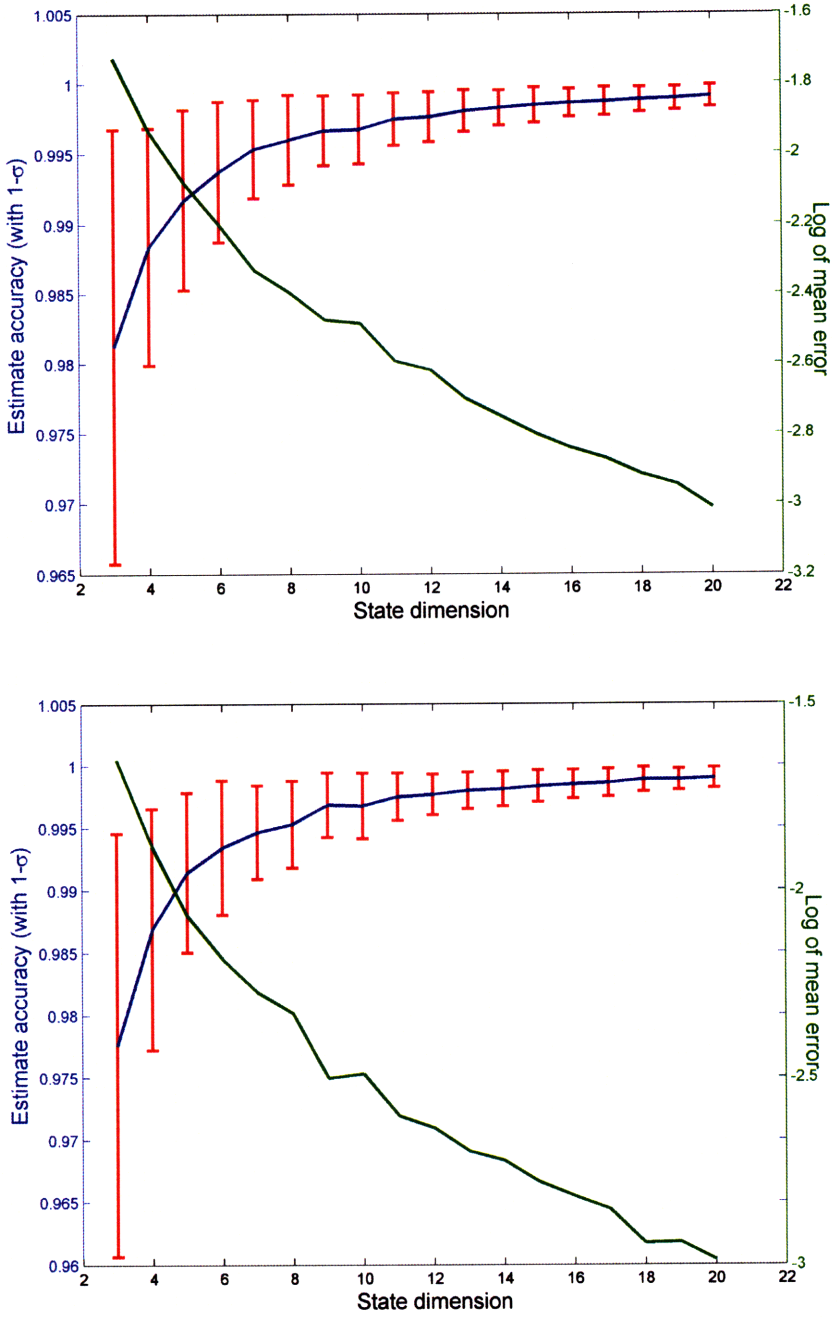


Fig. 2-5: Two moment approximation for the Dirichlet for distribution with (top) small values of $\alpha \in [2,\ 5]$, and (bottom) high values of $\alpha \in [2,\ 50]$

generated for a probability distribution in fact satisfy the assumptions of a probability vector, subject to an appropriate choice of weights.

Proposition 3 *If* $\mathbf{E}[\mathbf{p}]$ *and* $\boldsymbol{\Sigma}$ *are the mean and covariance of a Dirichlet distribution, then each Sigma Point satisfies a probability vector; namely, each* $\mathcal{Y}_i$ *satisfies* $\mathbf{1}^T \mathcal{Y}_i = 1, \quad \forall i$, *where* $\boldsymbol{\Sigma}_i^{1/2}$ *is the* i^{th} *column of the square root of the covariance matrix* $\boldsymbol{\Sigma}$

Proof: See Appendix B. ∎

The following additional proposition constrains the choice of the parameter β to ensure that the Sigma Points generated completely satisfies the requirement that $0 \leq \mathcal{Y}_i \leq 1$.

Proposition 4 *If* $\mathbf{E}[\mathbf{p}]$ *and* $\boldsymbol{\Sigma}$ *are the mean and covariance of a Dirichlet distribution, the maximum positive value for the parameter* β, $\beta_{\max}$, *which guarantees that each Sigma Point* $\mathcal{Y}_i = \mathbf{E}[\mathbf{p}] \pm \beta_{\max,i} \boldsymbol{\Sigma}_i^{1/2}$ *is a probability vector, is given by*

$$\beta_{\max,i} = \max \left(\frac{\mathbf{E}[\mathbf{p}]_i}{\boldsymbol{\Sigma}_{ii}^{1/2}}, \frac{1 - \mathbf{E}[\mathbf{p}]_i}{\boldsymbol{\Sigma}_{ii}^{1/2}}, \frac{-\mathbf{E}[\mathbf{p}]_i}{\boldsymbol{\Sigma}_{ij}^{1/2}}, \frac{-1 + \mathbf{E}[\mathbf{p}]_i}{\boldsymbol{\Sigma}_{ij}^{1/2}} \right) \tag{2.26}$$

where $\boldsymbol{\Sigma}_{ij}^{1/2}$ *is the* $(i,j)^{th}$ *entry of the square root of the covariance matrix* $\boldsymbol{\Sigma}$, *and* $\mathbf{E}[\mathbf{p}]_i$ *is the* i^{th} *row of the mean probability vector.*

Proof: See Appendix B. ∎

Based on this statistical description of the uncertainty in the transition probabilities, the Sigma Point sampling algorithm applied to uncertain MDPs selects the following Dirichlet Sigma Points

$$\begin{aligned} \mathcal{Y}_0 &= \mathbf{E}[\mathbf{p}] \\ \mathcal{Y}_i &= \mathbf{E}[\mathbf{p}] + \beta_{\max} \left(\boldsymbol{\Sigma}^{1/2} \right)_i \; \forall i = 1, \ldots, N \\ \mathcal{Y}_i &= \mathbf{E}[\mathbf{p}] - \beta_{\max} \left(\boldsymbol{\Sigma}^{1/2} \right)_i \; \forall i = N+1, \ldots, 2N \end{aligned} \tag{2.27}$$

2.5.2 Dirichlet Sigma Point Discussion

Remark 1 (Relation to Sigma Points): The Dirichlet Sigma Points can be interpreted as modified and constrained versions of the Sigma Points originally developed for a Gaussian density, since they sum to unity, and each Dirichlet Sigma Point must be between 0 and 1. A visualization of the Dirichlet Sigma Points is in Figure 2-6, where the Dirichlet Sigma Points (blue) are shown for different values of the credibility region (shown in red). While the credibility region increases (e.g., from a 95% to a 99% region), the Sigma Points are expanded outwards and thus cover a greater region of the probability simplex, while for smaller values of the credibility region, the Sigma Points are tightly clustered. Recall that this is in fact a visualization of the Dirichlet Sigma Points for a *row* of the transition matrix.

Remark 2 (Sampling requirement): The Sigma Point algorithm for an N_S dimensional vector requires $2N_S + 1$ total samples. Hence, even for a 100-state system, only 201 total samples are generated. Random sampling methods like MEDUSA [45] often use a heuristic number of samples, or need large-scale Monte Carlo investigation of the total number of simulations required to achieve a desired confidence level since the sampling is done in a completely random fashion. The Sigma Point algorithm however, explores along the principal components of the probability simplex identifying samples that have a β deviation along those components, and so captures the statistically relevant regions of uncertainty. Furthermore, since the number of samples scales linearly with the number of dimensions, the uncertainty can be absorbed readily in more sophisticated problems. For each transition probability matrix row, only a total of $2N + 1$ Sigma Points are required.

Remark 3 (Two-moment approximation): The two-moment approximation of the Dirichlet distribution implies that there might be inaccuracies in the third and higher moments of a reconstructed Dirichlet distribution. However, the higher moments of the Dirichlet decay to zero very rapidly (see for example Mannor [59]), and experience has shown that the two-moment approximation is quite accurate.

Figure 2-7 shows the result of solving a 2-dimensional infinite horizon Machine

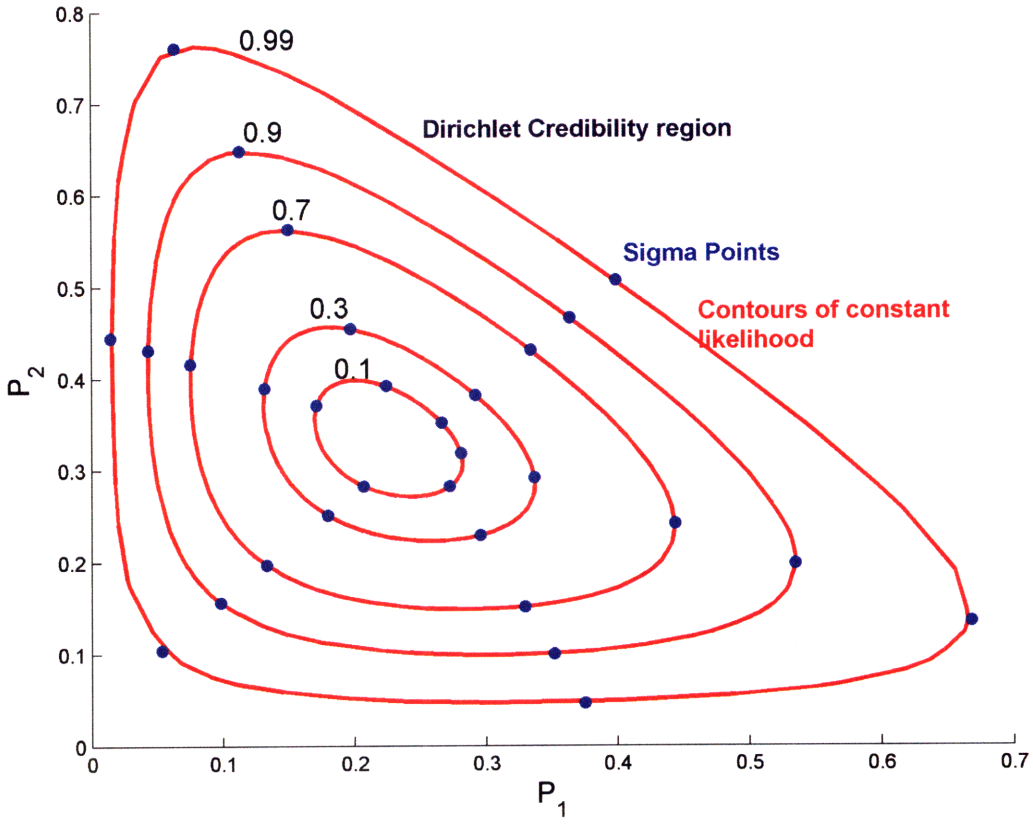


Fig. 2-6: Dirichlet Sigma Points (blue) shown to approximate the contours of constant likelihood (red) for different sizes of the credibility region.

Repair problem we will revisit in the numerical examples, using Monte Carlo realizations of the transition probabilities (red), and evaluating the cost $J = (J(1),\ J(2))$ associated with this optimal policy. That is, an optimal policy and cost were calculated for each realization of the transition probability matrix. The Dirichlet Sigma Points (blue) were also used to find the optimal policy and cost. The distribution of the costs obtained with these two methods are shown in Figure 2-7 and the Dirichlet Sigma Points approximate the cost distribution extremely well. Furthermore, the worst case cost of $J = (9.5, 10)$ is found by evaluating only 9 Dirichlet Sigma Points instead of evaluating all 500 Monte Carlo realizations.

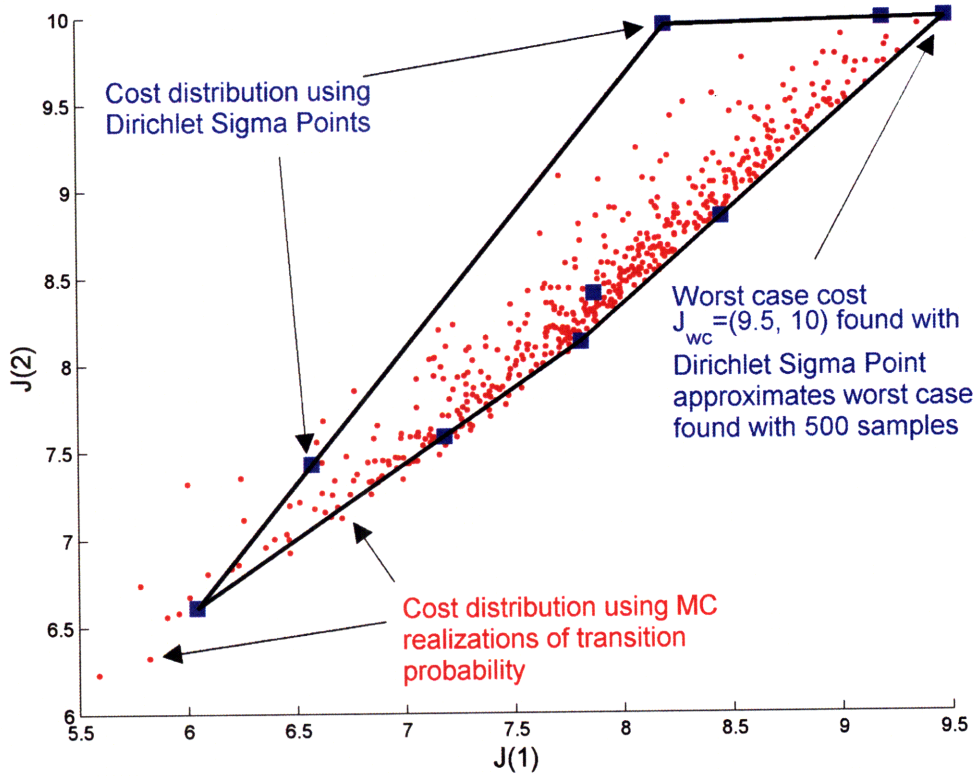


Fig. 2-7: Cost distribution approximation using Dirichlet Sigma Points (blue)

2.5.3 Robust MDP Using Sigma Point Sampling

The new robustness objective of Eq. (2.16) can now be specified in terms of the finite number of Sigma Point samples. Rather than solving the optimization problem

$$J_R^* = \min_u \max_{\Pi \in \Pi_s} \mathbf{E}\ [J_u(x_0)] \tag{2.28}$$

over the uncertainty set $\Pi \in \Pi_s$ containing the scenarios, the robust optimization is solved over the smaller set of scenarios generated by the Dirichlet Sigma Points $\mathcal{Y} \subseteq \Pi$,

$$J_{SP}^* = \min_u \max_{\Pi \in \mathcal{Y}} \mathbf{E}\ [J_u(x_0)] \tag{2.29}$$

Algorithm 4 Sigma Point Sampling for Uncertain MDP

1: Select $\beta = [0, \beta_{\max}]$ using Proposition 4
2: Select uncertainty model for i^{th} row of transition matrix by choosing appropriate parameters α for the Dirichlet distribution, $\Pi_{i,\cdot} \sim f_D(\mathbf{p} \mid \alpha)$
3: Calculate the mean and covariance

$$\mathbf{E}[\mathbf{p}] = \mathbf{E}[\Pi_{i,\cdot}] = \alpha_i / \sum_i \alpha_i$$

$$\mathbf{\Sigma} = \mathbf{E}[(\Pi_{i,\cdot} - \mathbf{E}[\mathbf{p}])(\Pi_{i,\cdot} - \mathbf{E}[\mathbf{p}])^T]$$

4: Generate the samples using the Sigma Point algorithm according to Eq. (2.40)
5: Solve the robust problem using the Sigma Points and Robust Dynamic Programming

$$J^*_{SP} = \min_u \max_{\mathcal{Y}_i} \mathbf{E}\left[J_u\right]$$

The full implementation of the Sigma Point sampling approach for an uncertain MDP is shown in Algorithm 4. The choice of β and the selection of the Dirichlet distribution $f_D(\mathbf{p} \mid \alpha)$ are made prior to running the algorithm. Using the uncertainty description given by $f_D(\mathbf{p} \mid \alpha)$, the mean and covariance are used to generate the Sigma Points $\mathcal{Y}_i$, which are the realizations for each of the models of the uncertain MDP. Robust Dynamic Programming [69] is used to find the optimal robust policy.

2.5.4 Choice of β

The selection of β is a critical choice for the algorithm, and any decision-maker that is extremely concerned with the worst case would obviously choose $\beta = \beta_{\max}$. However, in this section we provide insight into choosing other values for β to trade off this worst-case approach by using the notion of the credibility region introduced earlier.

The Sigma Points, $\mathcal{Y}_i \in \mathcal{R}^N$ are defined as follows for a distribution with mean

$\mathbf{E}[\mathbf{p}] \in \mathcal{R}^N$ and covariance $\Sigma \in \mathcal{R}^{N \times N}$, where N is the state dimension.

$$\begin{aligned} \mathcal{Y}_0 &= \mathbf{E}[\mathbf{p}] \\ \mathcal{Y}_i &= \mathbf{E}[\mathbf{p}] + \beta \left(\mathbf{\Sigma}^{1/2}\right)_i \; \forall i = 1, \ldots, N \\ \mathcal{Y}_i &= \mathbf{E}[\mathbf{p}] - \beta \left(\mathbf{\Sigma}^{1/2}\right)_i \; \forall i = N+1, \ldots, 2N \end{aligned} \tag{2.30}$$

The choice of β captures the amount of uncertainty the user is willing to embed in the problem, and is related to a percentile criterion of the credibility region.

Choice of β for a Beta distribution

First, we address the issue of choosing β for a 2-parameter Dirichlet distribution known as the Beta distribution. Suppose that a user is interested in accounting for a credibility region with $\eta = 95\%$ for a Beta distribution with parameters a and b. For completeness, the Beta distribution is defined as [33]

$$f_B(p|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} p^{a-1}(1-p)^{b-1} \tag{2.31}$$

Finding the η percentile is equivalent to finding an interval on $[l^-,\ l^+]$ such that

$$\eta = \int_{\mathbf{E}[p]-\beta\Sigma^{1/2}}^{\mathbf{E}[p]+\beta\Sigma^{1/2}} f_B(p|a,b)dp \tag{2.32}$$

where η is the desired percentile (e.g., $\eta = 0.95$ for a 95% percentile), $\mathbf{E}[p]$ is the mean of the variable, and Σ is the variance. The parameters of the Beta distribution are a and b. Since this is a single equation with two unknowns, we make the additional simplification that the interval is symmetric, thus making the optimization well posed[7]

$$\begin{aligned} l^- &= \mathbf{E}[p] - \beta\sqrt{\Sigma} \\ l^+ &= \mathbf{E}[p] + \beta\sqrt{\Sigma} \end{aligned} \tag{2.33}$$

[7]Conditions under which this may not be warranted are those where a lower and upper quantile are provided, and thus the optimization is over two equations and two unknowns and is thus well posed.

where $\bar{p}$ is the mean value of the Beta distribution, Σ is the variance, and now β is the optimization variable that needs to be found. The optimization is therefore that of finding the β such that

$$\mathbf{OB}: \left\{ \min_{\beta} |\gamma - \int_{\mathbf{E}[p]-\beta\Sigma^{1/2}}^{\mathbf{E}[p]+\beta\Sigma^{1/2}} f_B(p|a,b)dp| \right. \tag{2.34}$$

While the integral is known as the incomplete Beta function, and does not admit a closed form solution, this can be solved via a bisection algorithm over β (a related problem of finding the Beta parameters from quantile information is provided in vanDorp [85], where a bisection scheme is also used to find the upper and lower quantiles of a Beta density). The justification for using the bisection algorithm is that the optimization is over the Beta cumulative distribution which is a smooth (and continuous) function over $[0, 1]$. Hence, there exists a unique solution β^* for the optimization **OB**. Such a bisection algorithm is shown in Algorithm 5.

Figures 2-8 and 2-9 show an increase in β for a Beta distribution with an increased credibility region, which implies a higher degree of conservatism. Figures 2-8 and 2-9 show the equivalence of finding the tuning parameter β for a zero-mean, unit variance Gaussian distribution (red) and a Beta distribution (blue). For a Gaussian, a 95% percentile implies $\beta = 2$, while for a Beta distribution, a 95% percentile implies $\beta \approx 2$.

Choice of β for a Dirichlet distribution

The tuning parameter β can be obtained for the Dirichlet using the results obtained earlier with the credibility region. In fact, one can first sample to find the credibility region approximately with the samples q_i (from Monte Carlo sampling)

$$\int_{\mathcal{P}} f_D(p \mid \alpha)dp \geq \eta \approx \sum_i \delta_i(f_D(q_i \mid \alpha) \geq \eta) \tag{2.36}$$

Algorithm 5 Bisection Algorithm for optimal β

1: Inputs:

- Beta parameters a, b, termination threshold ϵ
- Threshold η

2: Output:

- Tuning parameter β

3: Initialize lower and upper bounds $l = 0$, $u = 1$, and $d = 1/2$
4: Beta distribution mean $= a/(a+b)$
5: Compute the incomplete Beta function

$$J(d) = \int_{\bar{p}-d}^{\bar{p}+d} f_B(p|a,b)dp \tag{2.35}$$

6: **if** $J(d) - \eta < \epsilon$ **then**
7: $\quad l = d$
8: **else if** $J(d) - \eta \geq \epsilon$ **then**
9: $\quad u = d$
10: **end if**
11: **if** $|J(d) - \eta| < \epsilon$ **then**
12: $\quad \beta = d/\Sigma^{1/2}$
13: **end if**

and once the credibility region $\mathcal{P}$ is found, the optimal β can be found by equating the Dirichlet Sigma Point $\mathcal{Y}_i$ with the likelihood

$$f_D(\mathcal{Y}_i \mid \alpha) = k(\eta) \tag{2.37}$$

Note that for each $\mathcal{Y}_i$, there is only a single β, and so this is a single equation with one unknown. By the unimodality of the Dirichlet, we can take log-likelihoods of both sides to obtain

$$\begin{aligned} \log(k(\eta)) - \log(K) &= \sum_i \alpha_i \log(\mathcal{Y}_i) \\ &= \sum_i \alpha_i \log(\bar{p}_i + \beta\Sigma_i^{1/2}) \end{aligned} \tag{2.38}$$

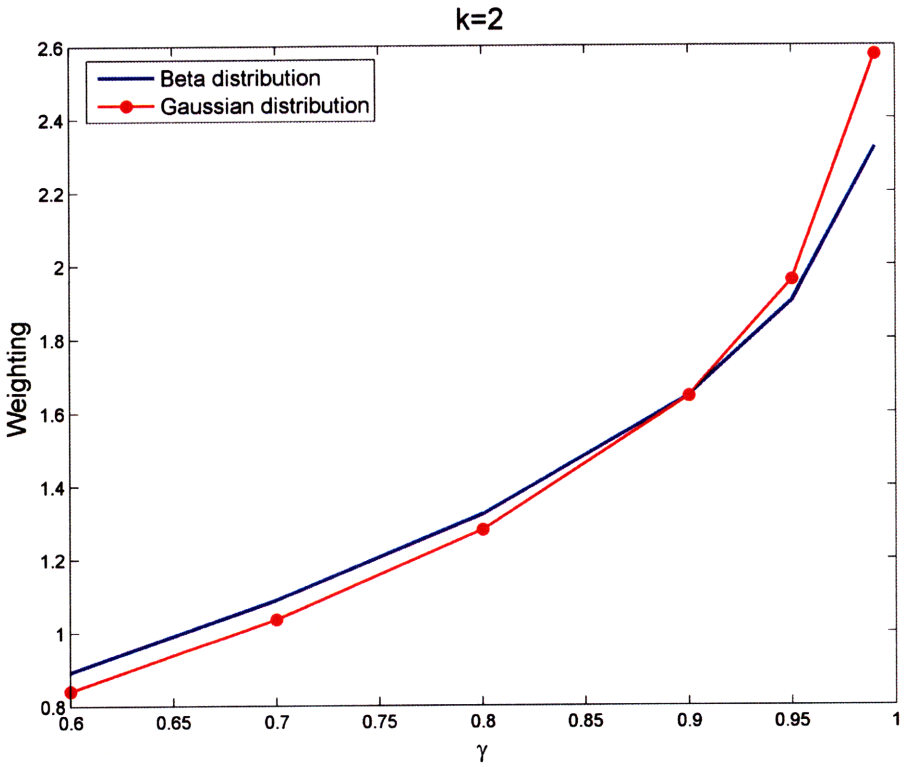


Fig. 2-8: Choosing β for a Beta distribution (blue) and a Gaussian distribution (red) is similar, but not identical for a two-state system. Here, the distribution has a large variance ("low weighting"); a user that wanted to include 95% variation for the uncertain variable $\mathbf{p}$ would choose $\beta = 2$ for a Gaussian distribution, but $\beta \approx 2.25$ for a Beta distribution.

This equation can also be solved using a bisection scheme over β, since the Dirichlet is globally concave. In summary, this section has shown how to select the parameter β for the Sigma Points, based on the desired size of the credibility region η.

2.5.5 Relations to the Ellipsoidal Model

This section draws an interesting link to the ellipsoidal scheme of Nilim with the Dirichlet Sigma Points. Nilim's ellipsoidal uncertainty model [69] is a second order approximation to the log likelihood function. In the ellipsoidal formulation, the uncertainty model is an ellipsoid intersected with the probability simplex and results in

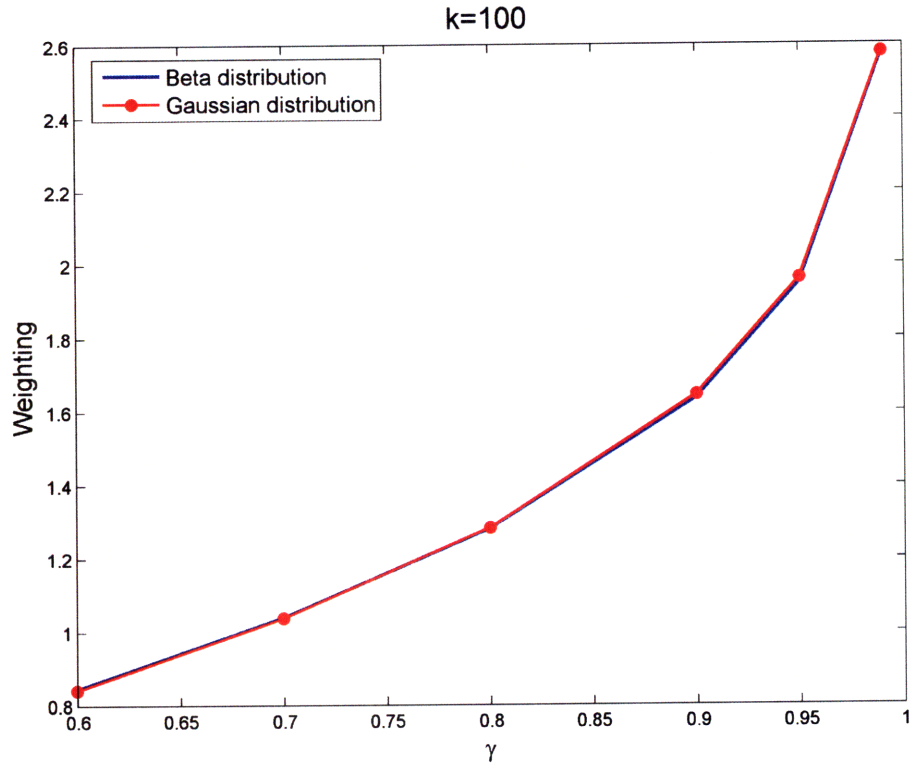


Fig. 2-9: Choosing β for a Beta distribution (blue) and a Gaussian distribution (red) is similar, but not identical for a two-state system. Here, the distribution has a small variance ("high weighting"); a user that wanted to include 95% variation for the uncertain variable $\mathbf{p}$ would choose $\beta = 2$ for a Gaussian distribution, and also for a Beta distribution.

the following expression

$$P_{\text{ell}} = \left\{ p \in \mathcal{R}^N \mid \mathbf{1}^T p = 1, \quad \sum_{j=1}^{N} \frac{(p(j) - f(j))^2}{f(j)} \leq \kappa^2 \right\} \tag{2.39}$$

where κ^2 is provided a priori (or found via resampling), $f(j)$ are the empirical frequencies, and $\mathbf{1}$ is the vector of ones.

This ellipsoidal optimization is performed at each time step in the robust value iteration in order to find a set of feasible p. Consider now solving Eq. 2.39 with the additional constraint that the optimization variables p are required to be Sigma Points: that is, replacing p with $\mathcal{Y}$. Recalling the Sigma Points definition (where we

have replaced $\bar{p}$ with the empirical frequency f),

$$\begin{aligned} \mathcal{Y}_0 &= f \\ \mathcal{Y}_i &= f + \beta_i \left(\mathbf{\Sigma}^{1/2}\right)_i \; \forall i = 1, \ldots, N \\ \mathcal{Y}_i &= f - \beta_i \left(\mathbf{\Sigma}^{1/2}\right)_i \; \forall i = N+1, \ldots, 2N \end{aligned} \tag{2.40}$$

then, the ellipsoidal approximation results in

$$P_{\text{ell}} = \left\{ p \in \mathcal{R}^N \mid \sum_{j=1}^{N} \frac{(\beta_i \sigma_{i,j})^2}{f(j)} \leq \kappa^2 \right\} \tag{2.41}$$

This inequality provides an alternative method for choosing β. Select β_i as

$$\beta_i^2 \leq \frac{\kappa^2}{\sum_{j=1}^{N} \sigma_{i,j}^2 / f(j)} \tag{2.42}$$

Note that in this case, the problem can be either that of *choosing* β (in which case this is a very easy 1-dimensional optimization), or that of simply fixing the choice of β based on the previous discussions of this chapter.

An example of the latter case is shown in Figures 2-10 and 2-11, where the Dirichlet Sigma Points (red) are compared to the ellipsoidal approximation (blue contours) as the algorithm proceeds in the value iteration steps. At convergence, the Dirichlet Sigma Points found an optimal (robust) solution of 16.607, while the ellipsoidal method had a solution of 16.652, which is within 99% of optimality. The Dirichlet Sigma Points solutions were obtained in approximately half the time of the ellipsoidal method, which had to solve a linear program with quadratic constraints.

2.6 Example: Machine Repair Problem

This section considers a small, but illustrative, numerical examples using a machine repair problem adapted from Bertsekas [10], and investigates the effect of the errors in the transition probability.

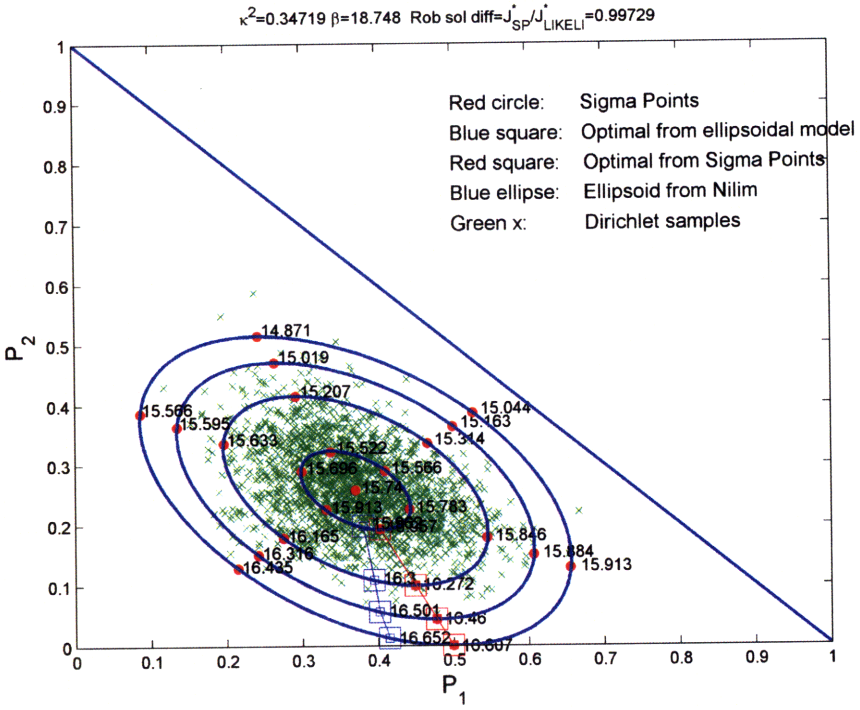


Fig. 2-10: Comparison of ellipsoidal approximation with Sigma Points

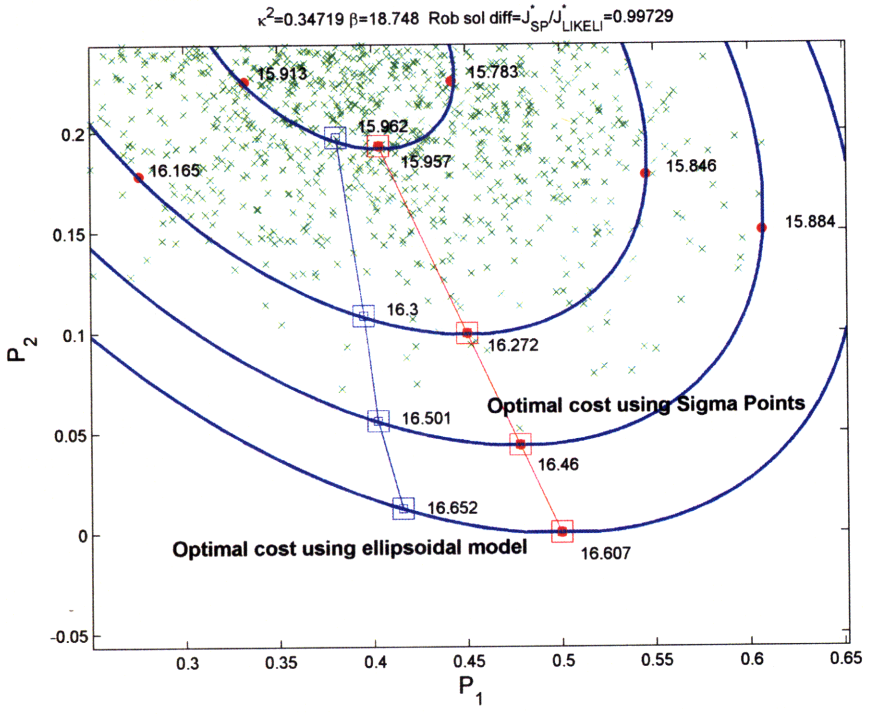


Fig. 2-11: Zoomed comparison of ellipsoidal approximation

Table 2.2: Nominal Machine Repair Problem

$\mathbf{x_k = 1}$ (Machine running):

$$\begin{aligned} \mathbf{J}_{k+1} &= \Pi_1 \mathbf{J}_k + \mathbf{C_1} \\ a_k^*(x_k = 1) &= \arg\max_{a_k} \{J_{k+1}(x_k = 1, a_k = \text{m}), J_{k+1}(x_k = 1, a_k = \text{n})\} \end{aligned}$$

$\mathbf{x_k = 0}$ (Machine not running):

$$\begin{aligned} \mathbf{J}_{k+1} &= \Pi_0 \mathbf{J}_k + \mathbf{C_0} \\ a_k^*(x_k = 0) &= \arg\max_{a_k} \{J_{k+1}(x_k = 0, a_k = \text{r}), J_{k+1}(x_k = 0, a_k = \text{p})\} \end{aligned}$$

$$\begin{aligned} \Pi_1 &= \begin{bmatrix} \gamma_1 & 1-\gamma_1 \\ \gamma_2 & 1-\gamma_2 \end{bmatrix}, \quad \mathbf{C_1} = [C_{maint} \quad 0]^T \\ \Pi_0 &= \begin{bmatrix} \gamma_3 & 1-\gamma_3 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{C_0} = [C_{repair} \quad C_{replace}]^T \\ \mathbf{J} &= [J_{k+1}(x_k = 1) \quad J_{k+1}(x_k = 0)]^T \end{aligned}$$

A machine can take on one of two states x_k at time k: *i*) the machine is either *running* ($x_k = 1$), or *ii*) broken (not running, $x_k = 0$). If the machine is running, a profit of \$100 is made. The control options available to the user are the following: if the machine is running, a user can choose to either *i*) perform maintenance (abbreviated as $a_k = m$) on the machine (thereby presumable decreasing the likelihood the machine failing in the future), or *ii*) leave the maching running without maintenance ($a_k = n$). The choice of maintenance has cost, C_{maint}, e.g., the cost of a technician to maintain the machine.

If the machine is broken, two choices are available to the user: *i*) repair the machine ($a_k = r$), or *ii*) completely replace the machine ($a_k = p$). Both of these two options come at a price, however; machine repair has a cost C_{repair}, while machine replacement is $C_{replace}$, where for any sensible problem specification, the price of replacement is greater than the repair cost $C_{replace} > C_{repair}$. If the machine is replaced, it is *guaranteed* to work for at least the next stage.

For the case of the machine running at the current time step, the state transitions

are governed by the model

$$\Pr\ (x_{k+1} = \text{fails} \mid x_k\text{=running},\ a_k\text{=m}) = \gamma_1$$
$$\Pr\ (x_{k+1} = \text{fails} \mid x_k\text{=running},\ a_k\text{=n}) = \gamma_2$$

For the case of the machine not running at the current time step, the state transition are governed by the following model

$$\Pr\ (x_{k+1} = \text{fails} \mid x_k\text{=fails},\ a_k\text{=r}) = \gamma_3$$
$$\Pr\ (x_{k+1} = \text{fails} \mid x_k\text{=fails},\ a_k\text{=p}) = 0$$

Note that, consistent with our earlier statement that machine replacement guarantees machine function at the next time step, the transition model for the replacement is deterministic. From these two models, we can completely describe the transition model if the machine is running or not running at the current time step:

$$\text{Running}\ (x_k = 1) : \Pi_1 = \begin{bmatrix} \gamma_1 & 1-\gamma_1 \\ 1-\gamma_2 & \gamma_2 \end{bmatrix}$$
$$\text{Not Running}\ (x_k = 0) : \Pi_0 = \begin{bmatrix} \gamma_3 & 1-\gamma_3 \\ 1 & 0 \end{bmatrix}$$

The objective is to find an optimal control policy such that $a_k(x_k = 0) \in \{$ r, p $\}$ if the machine is not running, and $a_k(x_k = 1) \in \{$ m, n $\}$ if the machine is running, for each time step. The state of the machine is assumed to be perfectly observable, and this can be solved using Dynamic Programming

$$J_k(i) = \max_{a_k \in \mathcal{A}} \left[g(x_k, a_k) + \sum_j \Pi_{ij}^a J_{k+1}(j) \right]$$

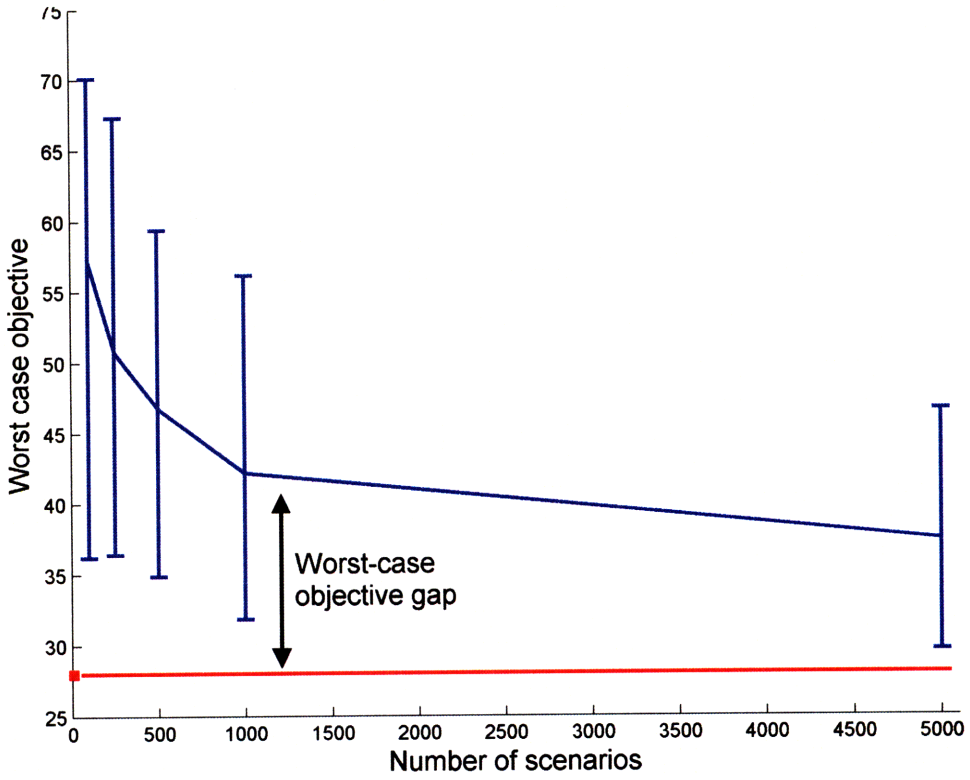


Fig. 2-12: The difference between the worst case objective through sampling (blue) and Sigma Point sampling (red) decreases only slightly as the number of simulations are increased significantly. The Sigma Point sampling strategy only requires 5 samples to find the worst case objective of $J^* = 28$, but the line has been extended for comparison.

2.6.1 Uncertain Transition Models

In this numerical example, it is assumed that the transition model Π_0 is uncertain; that is, there are errors in the likelihood of the machine failing after is repaired. This is a credible assumption if the person repairing it is new to the job, for example, or there is some uncertainty on the original cause of the machine failure.

The robust control $u^*_{R,k}$ maximizes the objective function over all matrices Π_0 in the uncertainty set $\tilde{\Pi}_0$ that minimize the objective function

$$J^*_k(i) = \min_{\tilde{\Pi} \in \tilde{\Pi}} \max_{a_k \in \mathcal{A}} \left[g(x_k, a_k) + \sum_j \tilde{\Pi}^a_{ij} J^*_{k+1}(j) \right]$$

Note that since the transition model Π_1 is well-known, the robust counterpart of the nominal problem only needs to be formulated for the model Π_0.

The solution approach using Sigma Point Sampling generate realizations of the matrix Π_0 based on Algorithm 4, and in particular, the Sigma Points were found by

$$\begin{aligned} \mathcal{Y}_0 &= \mathbf{E}[\Pi_0] \\ \mathcal{Y}_i &= \mathbf{E}[\Pi_0] + \beta_{\max} \left(\mathbf{\Sigma_\Pi}^{1/2} \right)_i \ \forall i = 1, \ldots, N \\ \mathcal{Y}_i &= \mathbf{E}[\Pi_0] - \beta_{\max} \left(\mathbf{\Sigma_\Pi}^{1/2} \right)_i \ \forall i = N+1, \ldots, 2N \end{aligned} \tag{2.43}$$

2.6.2 Numerical Results

The machine repair problem with uncertain Π_0 was evaluated multiple times with random realizations for the transition matrix Π_0, and compared with the Sigma Point algorithm.

The main result comparing the Sigma Point approach to random sampling is shown in Figure 2-12 where the worst case objective (y-axis) is plotted as a function of the number of samples required. The blue line is the worst case found by using conventional sampling, and the red line is the Sigma Point worst-case using $\beta = 3$. This choice of β was in fact sufficient for this example to find the worst case of $J_{wc} = 28$. Note the slow convergence of the brute force sampling, with a significant gap even with 1200 samples. The Sigma Point only required 5 samples, since the uncertainty was only in one transition model of dimension $\mathcal{R}^{2\times 2}$. Hence, $N_s = 2 \times 2 + 1 = 5$. Note that the number of scenarios required to find the worst case varied significantly with the choice of hyperparameters α_i of the Dirichlet distribution. When $\alpha_i \approx 100$, for example, the Dirichlet distribution has a much smaller variance than when $\alpha_i \approx 10$ and the total number of samples required to find the worst case for $\alpha_i \approx 10$ is smaller than $\alpha_i \approx 100$.

Figure 2-13 shows the performance of the worst case as a function of the parameter $\beta \in [0, 1]$. The objective of this figure is to show the tradeoff between protecting against the worst-case and choice of the parameter β. Since the Sigma Points only

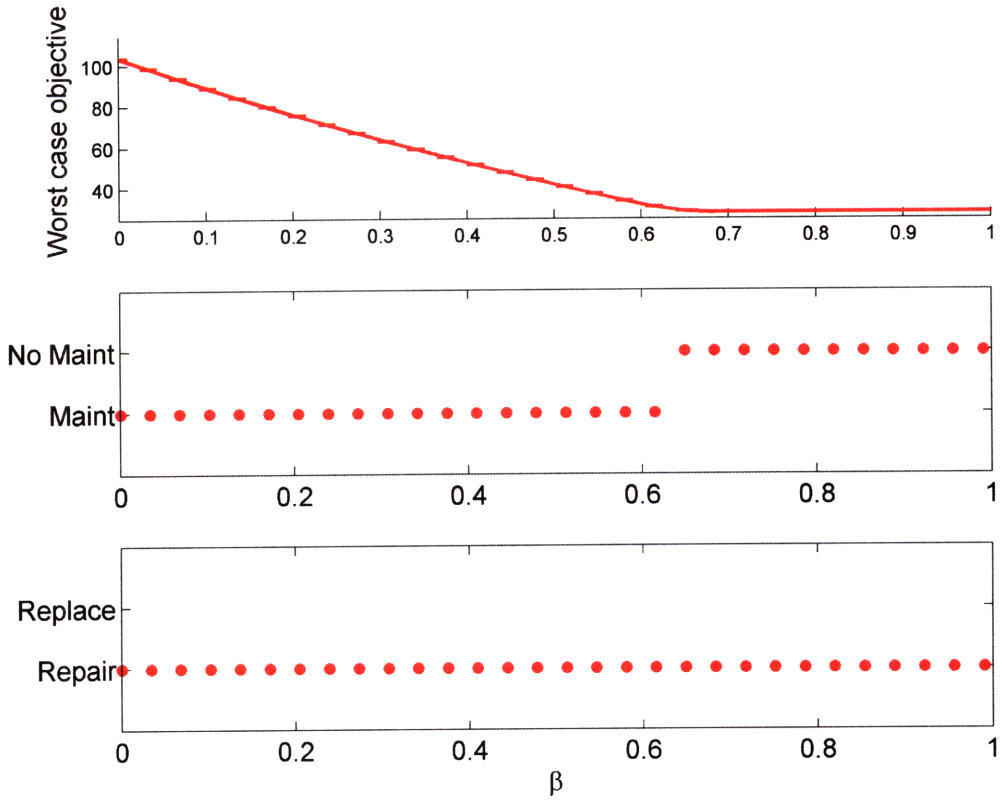


Fig. 2-13: Sigma Point sample tradeoff of robust performance (top subfigure) vs. normalized β shows that increasing the robustness also decreases the objective. The robust policy (bottom two figures) switches at $\beta = 0.65$.

require a small number of samples to find the worst case in this smaller machine repair example, this tradeoff can be performed very quickly.

The worst case objective was found for each value of β and is shown in the top figure. The bottom two subfigures show the policy as a function of β. For $\beta < 0.65$, the optimal (robust) policy is to perform maintenance, while if $\beta \geq 0.65$, the outcome of the maintenance is too uncertain, and it will be more cost effective (in a worst-case sense) to not perform maintenance at all. Hence, there is a discrete policy switch at $\beta = 0.65$ that indicates that a different decision should be made in response to the high uncertainty in the transition model.

2.7 Example: Robot on a Grid

In this next numerical example adapted from Russell and Norvig [78], consider an autonomous agent on a $M \times N$ grid that has to reach an exit (in minimum time) while accumulating a maximum reward. The exit is either a $+1$ reward or a -1 reward. The agent can move in any of the squares in othogonal directions and the control actions, u, available are $u = \{\text{up, down, left, right}\}$. To model the uncertainty in the agent's actions, the desired action u^* will only occur with probability p, and with probability $1-p$ the agent will move in an orthogonal direction. That is, if the agent selects $u = \text{up}$, then the agent will go up with probability p, but will move left or right with probability $1-p$. If the agent hits the wall, it will bounce back to the original location.

In this problem, the nominal objective is to maximize the expected reward; in the presence of uncertainty in the model parameter p, the adversarial effect of any uncertainty in the transition model will be to *decrease* the reward. The transition models Π^a of this MDP are parameterized by p, $\Pi^a(p) \in \mathcal{R}^{MN \times MN}$. Therefore, simply choosing different values for p will result in different transition models, and as a result, different optimal actions. The actual transition model is of dimension $\mathcal{R}^{MN \times MN}$, but for this problem is very sparse since the agent can only transition to adjacent cells.

Also, for the discounted case, infinite-horizon policies, $u^*(i \mid p, r(i))$ are parameterized by p and will in general vary with the reward $r(i)$ for being in each state i. Here it is assumed that $r(i)$ is given by

$$r(i) = \begin{cases} +1, & \text{Agent in high reward exit} \\ -1, & \text{Agent in low reward exit} \\ -0.02, & \text{In all other cells} \end{cases} \tag{2.44}$$

That is, the cell rewards are equal except for the exits, where the agent may either obtain a negative reward or a positive reward.

An optimal policy for an almost deterministic case is shown in Figure 2-14, where $p = 0.99$. In this case, an agent starting from cell(1,1) will try to reach the goal

cell(4,3) by implementing $u^* =$ up for two steps, and then $u^* =$ right for the remaining three steps. This will take 5 steps. Likewise, an agent starting from cell(3,1) will implement $u^* =$ up for two steps, and $u^* =$ right for a single step, and this will take 3 steps.

Next, we consider the case of uncertainty in the parameter p, the probability of performing the desired action. This parameter may not be well known, for example, if the agent is a newly designed robot and p may only be at best estimated by the designer.

We take a Bayesian approach, and use the Dirichlet distribution to model the prior of this parameter. (For this simple case, this Dirichlet becomes a Beta distribution on p):

$$f_D(\mathbf{p} \mid \alpha) = Kp^{\alpha_1 - 1}(1-p)^{\alpha_2 - 1} \tag{2.45}$$

where K is a normalization constant that ensures $f_D(\mathbf{p} \mid \alpha)$ is a proper distribution. From the parameters α_1 and α_2, we can calculate the mean and variance of p as

$$\begin{aligned} \bar{p} &= \alpha_1/(\alpha_1 + \alpha_2) \\ \sigma_p^2 &= \frac{\alpha_1 \alpha_2}{(\alpha_1 + \alpha_2)^2 \, (\alpha_1 + \alpha_2 + 1)} \end{aligned}$$

By appropriately choosing α_1 and α_2, we can come up with three distinct cases for p. These are also shown in Table 2.3:

- Case I (Low Uncertainty): $\alpha_1 = 40$, $\alpha_2 = 10$
- Case II (Medium Uncertainty): $\alpha_1 = 12$, $\alpha_2 = 3$
- Case III (High Uncertainty): $\alpha_1 = 4$, $\alpha_2 = 1$

Note that for each case, the mean of p is the same, $\bar{p} = 0.8$, but the variance is different.

Table 2.3: Different Levels of Uncertainty for p

Case #	Uncertainty Level	α_1	α_2	Mean, $\bar{p}$	Variance, σ_p^2
I	Low	40	10	0.80	0.003
II	Medium	12	3	0.80	0.010
III	High	4	1	0.80	0.027

Nominal Policies

The Certainty Equivalent optimization will be identical for each Low, Medium, or High Uncertainty case since $\hat{A} = A(\bar{p})$. The resulting policies will also be the same. Hence, the certainty equivalent policy (CE) will be

$$u^*_{CE}(i) = \arg\max_u \left[g(i,u) + \sum_j \hat{\pi}^a_{ij} J^*(j) \right], \quad \forall i, a \tag{2.46}$$

where we are maximizing the reward (hence, $\max_u$ instead of $\min_u$). Such a policy is visualized in Figure 2-15. For this policy, the optimal action at cell(3,1) is to go left, since there is only an 80% of implementing the desired control action, as opposed to selecting u^* = up for the case of $\bar{p} = 0.99$. Note that for case III (High Uncertainty), p can actually take worst-case values much lower than $\bar{p} = 0.8$; the policies found from these worst-case p values result in different policies from policies using $\bar{p}$. For example, a 2-σ deviation from $\bar{p} = 0.80$ for case III will result in $p = 0.57$, and this policy is quite different from a policy that assumes $p = 0.80$: see Figure 2-16. In particular, if the agent is in the proximity of the low reward exit (cell(3,1)) and since the probability of performing the desired action is so low, the agent will perform actions that on average would not let it enter this cell. In this case, the optimal action u^* = left. This is so that with probability $1 - p$, the actions will be either up or down, but not right, which would send the agent in the low reward exit. The CE policy completely ignores this behavior by excluding the variability of p.

An example of the impact of the variability in p is shown in Figures 2-14 and 2-16. In Figure 2-14, the optimal policy was found using $p = 0.99$, and the true transition model was $A(p)^a$. Here, the optimal path starting from cell(2,1) takes the agent to the high reward goal in 4 time steps. In case of a worst-case value for p, $p = 0.6$, the

agent still uses the optimal policy found for $p = 0.99$ (see Figure 2-16, but now the agent ends up in a low reward state, and takes 5 steps. This was because the agent oscillated between two states.

Sigma Point Policies

The Sigma Point policies explicitly take into account the variability of p. For this simple problem, the Sigma Points are

$$\begin{aligned}\mathcal{Y}_1 &= \bar{p} \\ \mathcal{Y}_2 &= \bar{p} + \beta\sqrt{\Sigma_p} \\ \mathcal{Y}_3 &= \bar{p} - \beta\sqrt{\Sigma_p}\end{aligned}$$

where β is chosen to ensure that all the Sigma Points satisfy $0 \leq \mathcal{Y}_i \leq 1$.

The Sigma Point optimization is

$$u_S^*(i) = \arg \min_{\Pi \in (\mathcal{Y}_i(\beta))} \max_u \left[g(i,a) + \sum_j \tilde{A}_{ij}^a J^*(j) \right], \quad \forall i, a$$

The Sigma Point policy for the High Uncertainty environment, $\beta_{\max} = 10$, and $\beta = 0.5$ is identical to the policy $p = 0.60$. We solve this problem using scenario-based Robust Value Iteration, where each of the scenarios are the Sigma Points $\mathcal{Y}_i(\beta)$.

2.7.1 Numerical Results

To compare SP and CE, we computed the policies off-line, and then simulated sample agent paths using the worst-case values of p calculated as $p = \bar{p} - \beta\sqrt{\Sigma_p}$ for $\beta = \{0, 0.1, \ldots, 1\}$. This resulted in the following p (note they are parameterized by β),

Table 2.4: Uncertainty in p

p	0.8	0.76	0.73	0.69	0.65	0.62	0.58	0.55	0.51	0.47	0.44
β	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1

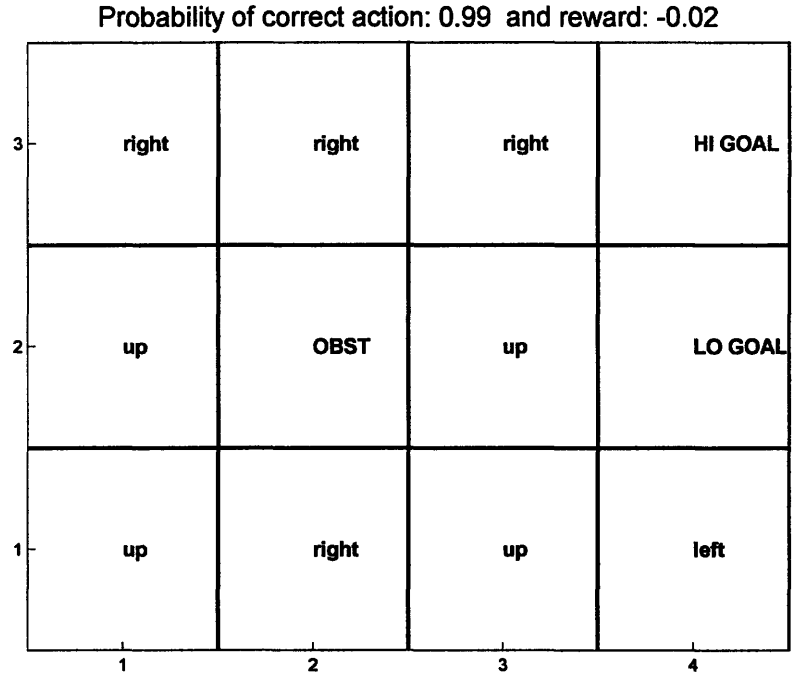


Fig. 2-14: Nominal policy when $p = 0.99$. Agent starting in cell (3,1) will choose $u =$ up to approach the high-reward goal (4,3) since agent will successfully implement desired control action with 99% probability.

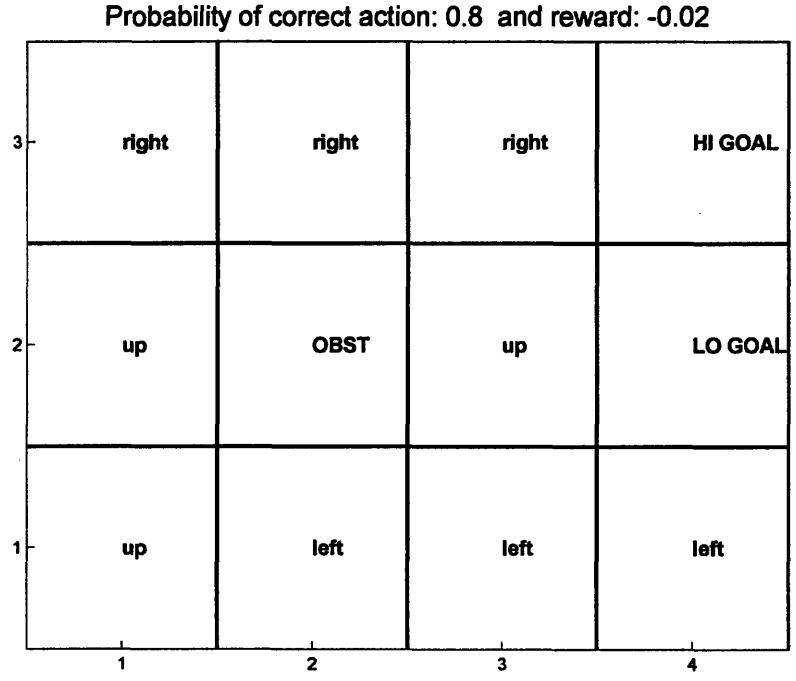


Fig. 2-15: Nominal policy when $p = 0.80$. Agent starting in cell (3,1) will move left since agent will successfully implement desired control action with 80% probability, and there is less risk by avoiding the neighborhood of the low reward cell.

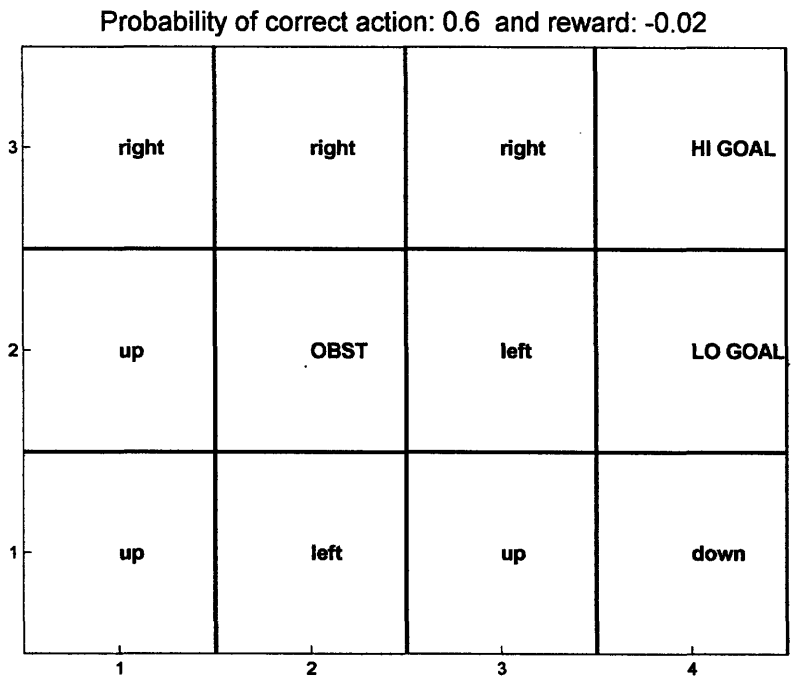


Fig. 2-16: Nominal policy when $p = 0.60$. Agent starting in cell (3,1) will now move up because there is a low probability (60%) of the agent actually performing this action successfully. The actions in the states neighboring the low reward cell are consistent with the agent avoiding this cell: for example, when the agent is in cell (3,2), the optimal action is to choose $u =$ left since, if this action is not performed, the agent will either implement $u =$ up or $u =$ down, but not $u =$ right, which is towards the low reward.

2.7.2 More general scenario

We conclude the autonomous agent planning problem with a larger scenario and slightly more complex obstacle structure (see Figure 2-17). The agent starts in cell (1,5) and can either exit in cell (2,1) collecting a low reward, or in cell (4,2) collecting a high reward. The optimal policy turns out to be the one that has the robot collecting the high reward. The probability of transitioning to an adjacent cell is different from cell to cell.

However, in the presence of transition model uncertainty (Figure 2-17), the robot path realizations end up in the low reward exit. By using the robust formulation with the Dirichlet Sigma Points (see Figure 2-18), the robot takes the slightly longer path, but avoids the low reward altogether.

The robust policy was obtained with the Dirichlet Sigma Points, and the compu-

Table 2.5: Suboptimality and Computation Time (T_c) for Different η

	$\eta = 95\%$		$\eta = 99\%$	
# Scenarios	Suboptim (%)	T_c	Subptim (%)	T_c
1250	3.5	3	6.7	3.7
2500	1.1	13	4.3	10.9
3750	0.5	15	1.6	17.0
Sigma Point	1.3	0.7	3.2	0.7

tational savings are shown in Table 2.5, as a function of the total number of scenarios used, the suboptimality ratio of the optimal (robust) objective using 5000 scenarios, and the overall computation time T_c. In order to achieve a suboptimality ratio of 1.1% with 2500 scenarios required a computation time of 13 seconds, while using the Dirichlet Sigma Points, a similar performance was obtained in only 0.7 seconds.

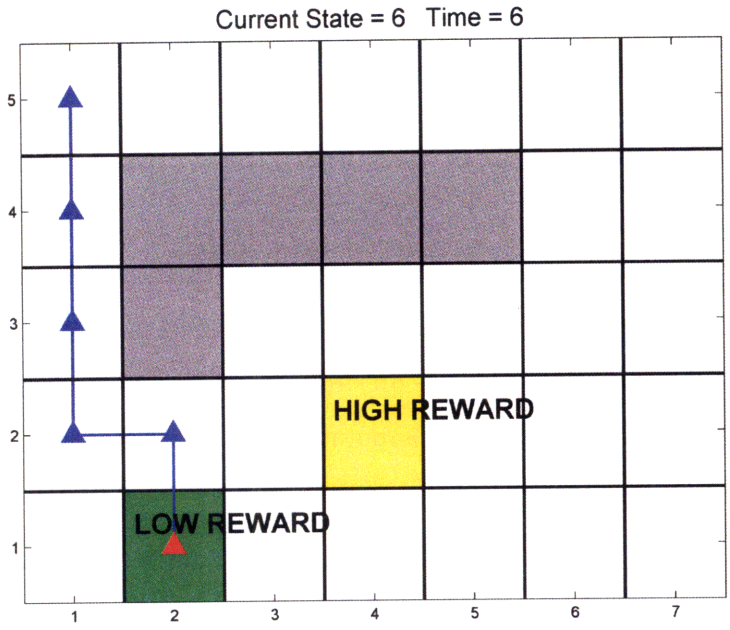


Fig. 2-17: A realization of the nominal policy under transition model uncertainty: the robot falls in the low reward exit, and accrues a large negative reward.

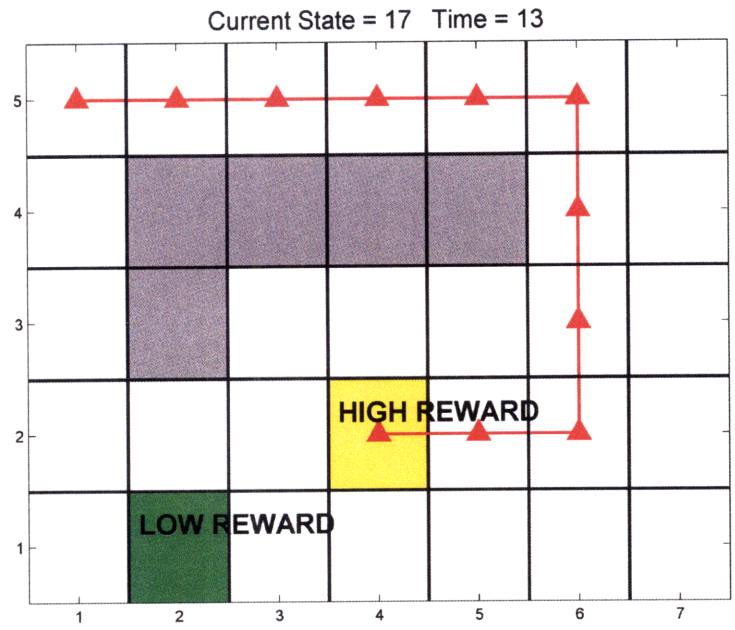


Fig. 2-18: A realization of the robust policy under transition model uncertainty: the robot takes a longer path, but avoids the low reward exit altogether.

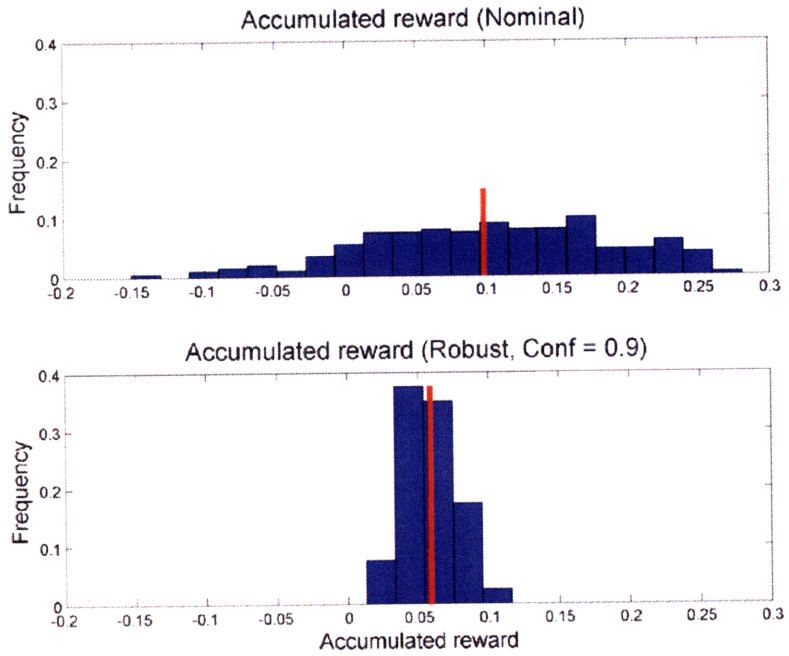


Fig. 2-19: Histogram of expected rewards for a nominal (above) and robust policy (below). Note that the robust policy has a slightly lower on average performance, but greatly improved worst case performance.

2.8 Conclusions

This chapter has discussed the role of the uncertainty in the transition probabilities of a Markov Chain, and how it impacts decision-making algorithms formulated as an MDP. We have taken a Bayesian approach to modeling the uncertainty in the transition probabilities, and presented a Monte Carlo-based bisection algorithm to precisely define a model uncertainty for these probabilities. This extends the applicability of results obtained by other authors to more Bayesian-based uncertainty descriptions.

We found that sampling-based strategies to find the robust policy was computationally expensive, and presented a computationally tractable technique – the Dirichlet Sigma Points – for efficiently creating the samples of the uncertainty transition probabilities. We have shown computational savings over otherwise straight Monte Carlo simulation, while at the same time maintaining consistent performance with these numerically intensive approaches.

Appendix 2A: Dirichlet Sigma Point Selection Proofs

This appendix shows that the Dirichlet Sigma Point algorithm generates samples that are proper transition probabilities, a critical point for using this approach in Markov Decision Processes. In particular, we will show that the quantity $\mathcal{Y} = \mathbf{E}[\mathbf{p}] + \beta\boldsymbol{\Sigma}^{1/2}$ satisfies a proper probability, namely that $\mathbf{1}^T(\mathbf{E}[\mathbf{p}] + \beta\boldsymbol{\Sigma}^{1/2}) = 1$. This is done by first showing (Proposition 5) that the row sums of a covariance matrix of a probability distribution sum to 0. Then we show (Proposition 6) that for any matrix whose row sum equal to zero, its square root (if it exists), will also have row sums equal to zero. Using these two proofs, we can then show that the quantity $\mathcal{Y}$ satisfies a probability (Proposition 7), and show additional requirements on the choice of β (Proposition 8).

Proposition 5 *(Row/column sum constraint on covariance matrix $\boldsymbol{\Sigma}$) The row and column sums of the entries of the covariance matrix of a probability vector $\boldsymbol{\Sigma}$ are equal to 0.*

Proof: Given a probability vector $\mathbf{p} = [p_0, p_1, \ldots, p_N]^T$, then the covariance matrix of this probability vector is given by

$$\boldsymbol{\Sigma} = \mathbf{E}[(\mathbf{p} - \mathbf{E}[\mathbf{p}])(\mathbf{p} - \mathbf{E}[\mathbf{p}])^T] \tag{2A-1}$$

However, since $\mathbf{p}$ is a probability vector, then $p_N = 1 - \sum_i p_i$, and thus the covariance matrix $\boldsymbol{\Sigma}$ will *not* be full rank, implying that $\exists v$ (a left eigenvector) such that

$$\mathbf{v}^T\boldsymbol{\Sigma} = \lambda\mathbf{v}^T = \mathbf{0} \tag{2A-2}$$

where λ is the eigenvalue, equal to 0 since the matrix $\boldsymbol{\Sigma}$ is not full rank. One such eigenvector is the vector of ones, $\mathbf{1} = [1,\ 1,\ 1,\ \ldots,\ 1]^T$

$$\begin{aligned}\mathbf{1}^T\boldsymbol{\Sigma}_i &= \mathbf{E}\left[(p_0 - \mathbf{E}[p_0])\sum_{i=0}^{N}(p_i - \mathbf{E}[p_i])\right] \\ &= \mathbf{E}\left[(p_0 - \mathbf{E}[p_0])\left(\underbrace{\sum_i p_i}_{=1} - \underbrace{\sum_i \mathbf{E}[p_i]}_{=1}\right)\right] = 0 \quad \blacksquare\end{aligned}$$

Example: (Dirichlet density) Consider the covariance matrix of the Dirichlet, which is given by

$$\Sigma(i,j) = \begin{cases} -\frac{\alpha_i\alpha_j}{\alpha_0^2(\alpha_0+1)} \text{If } i = j, \\ \frac{\alpha_i(\alpha_0-\alpha_i)}{\alpha_0^2(\alpha_0+1)} \end{cases}$$

Then,

$$\begin{aligned}\sum_j \Sigma(i,j) &= -\sum_{j\neq i}\frac{\alpha_i\alpha_j}{\alpha_0^2(\alpha_0+1)} + \frac{\alpha_i(\alpha_0-\alpha_i)}{\alpha_0^2(\alpha_0+1)} = \frac{1}{\alpha_0^2(\alpha_0+1)}\left(-\alpha_i\sum_{j\neq i}\alpha_j + \alpha_0\alpha_i - \alpha_i^2\right) \\ &= \frac{1}{\alpha_0^2(\alpha_0+1)}\alpha_i\left(\underbrace{(\sum_{j\neq i}\alpha_j + \alpha_i)}_{=0} - \alpha_0\right) = 0 \end{aligned} \tag{2A-3}$$

since the last summation term is equal to α_0.

Ultimately, however, the goal is to demonstrate that the quantity $\mathcal{Y}_i = \mathbf{E}[\mathbf{p}] + \beta^{1/2}(\Sigma^{1/2})_i$ satisfies the unit sum constraint of a probability vector. In order to prove this, we need to demonstrate the following intermediate result.

Proposition 6 *(Properties of a square root matrix) The matrix square root (B) of a matrix A whose row and column sums are zero also satisfies the property that row and column sums are equal to 0.*

Proof: Consider a positive semi-definite matrix $B \in \mathcal{R}^{N\times N}$ whose $(i,j)^{th}$ entry is B_{ij}. Also consider a matrix A such that B is the square root of A (when such a square root exists), namely $A = B^2$. In the case of a symmetric B, this implies that $A = BB = BB^T$. Consider the $(k,m)^{th}$ entry of A, A_{km}. Then, by direct matrix multiplication,

$$A_{km} = \sum_{j=1}^{N} B_{kj}B_{jm} \tag{2A-4}$$

Now, since the rows and columns of the matrix sum to zero, namely $\sum_k A_{km} = 0$ and $\sum_m A_{km} = 0$. Then, consider the k^{th} row sum

$$\sum_k A_{km} = \sum_k \sum_{j=1}^{N} B_{kj}B_{jm} = 0 \tag{2A-5}$$

which, by rearranging the summation is satisfied by

$$\sum_j \left(B_{jm} \underbrace{\sum_k B_{kj}}_{=0} \right) = 0 \tag{2A-6}$$

that is if the sum on the j^{th} column is zero, $\sum_k B_{kj} = 0$. In the case of the symmetric B, of course, this also implies $\sum_j B_{kj} = 0$. ∎

This is the result that we needed to therefore show that if $\mathbf{1}^T(\mathbf{E}[\mathbf{p}] + \beta\boldsymbol{\Sigma}_i) = 1$, and $\mathbf{1}^T\boldsymbol{\Sigma}_i = 0$, then the sum of the rows/columns of the matrix square root, $\boldsymbol{\Sigma}^{1/2}$, will also sum to 0, namely, $\mathbf{1}^T(\boldsymbol{\Sigma}^{1/2})_i = 0$. Thus, the probabilities formed by $\mathbf{E}[\mathbf{p}] + \beta\boldsymbol{\Sigma}_i$ will necessarily satisfy a probability vector.

Proposition 7 *(Mean-variance satisfies a probability vector) If* $\mathbf{E}[\mathbf{p}]$ *and* $\boldsymbol{\Sigma}$ *are the mean and covariance of a Dirichlet distribution,*

$$\mathbf{E}[\mathbf{p}] + \beta\boldsymbol{\Sigma}_i^{1/2} \tag{2A-7}$$

is a probability vector, where $\boldsymbol{\Sigma}_i^{1/2}$ *is the* i^{th} *column of the square root of the covariance matrix* $\boldsymbol{\Sigma}$

Proof: Following directly from the earlier propositions, since the square root of the covariance matrix satisfies $\mathbf{1}^T\boldsymbol{\Sigma}_i^{1/2} = 0$ (by Proposition 6), then

$$\begin{aligned}\mathbf{1}^T\left(\mathbf{E}[\mathbf{p}] + \beta\boldsymbol{\Sigma}_i^{1/2}\right) &= \mathbf{1}^T(\mathbf{E}[\mathbf{p}]) + \underbrace{\mathbf{1}^T\beta\boldsymbol{\Sigma}_i^{1/2}}_{=0} \\ &= \mathbf{1}^T(\mathbf{E}[\mathbf{p}]) \\ &= 1 \quad \blacksquare\end{aligned}$$

An important point, nonetheless, is that an appropriate selection for β is still required; while the probability vector constraint is implicitly satisfied (as we have shown), each entry is not enforced to satisfy a valid probability: i.e., there is no constraint on each probability to be non-negative or greater (in magnitude) to 1, only the sum constraint is satisfied with this approach.

Proposition 8 *(Selection of* β*) If* $\mathbf{E}[\mathbf{p}]$ *and* $\boldsymbol{\Sigma}$ *are the mean and covariance of a Dirichlet distribution, the maximum positive value for the parameter* β*,* β_{max}*, that guarantees that* $\mathbf{E}[\mathbf{p}] \pm \beta_{max}\boldsymbol{\Sigma}_i^{1/2}$ *is a probability vector is given by*

$$\beta_{max} = \frac{1}{\boldsymbol{\Sigma}_{ij}^{1/2}} \min\left(\mathbf{E}[\mathbf{p}]_i, 1 - \mathbf{E}[\mathbf{p}]_i\right) \tag{2A-8}$$

where $\mathbf{\Sigma}_{ij}^{1/2}$ *is the* $(i,j)^{th}$ *entry of the square root of the covariance matrix* $\mathbf{\Sigma}$*, and* $\mathbf{E[p]}_i$ *is the* i^{th} *row of the mean probability vector*

Proof: For $\mathbf{E[p]} \pm \beta\mathbf{\Sigma}_i^{1/2}$ to satisfy a probability vector, two conditions must be satisfied:

i) $\mathbf{1}^T(\mathbf{E[p]} \pm \beta\mathbf{\Sigma}_i^{1/2}) = 1, \quad \forall i$ (a probability vector sums to 1)

ii) $0 \leq \mathbf{E[p]}_i \pm \beta\mathbf{\Sigma}_{ij}^{1/2} \leq 1 \quad \forall i$ (each entry of the probability vector lies between 0 and 1)

Item *i)* is satisfied by Proposition 7, and hence we seek to find the maximum β that will satisfy item *ii)*. Addressing each side of the inequality,

$$0 \leq \mathbf{E[p]}_j \pm \beta\mathbf{\Sigma}_{ij}^{1/2} \longrightarrow \frac{\pm\mathbf{E[p]}_j}{\mathbf{\Sigma}_{ij}^{1/2}} \leq \beta$$

and

$$\mathbf{E[p]}_j \pm \beta\mathbf{\Sigma}_{ij}^{1/2} \leq 1 \longrightarrow \beta \leq \pm\frac{(1 - \mathbf{E[p]}_j)}{\mathbf{\Sigma}_{ij}^{1/2}}$$

In the first inequality, since only positive values are considered, then $\frac{\mathbf{E[p]}_j}{\mathbf{\Sigma}_{ij}^{1/2}} \leq \beta$. The minimum value for the second inequality to hold is given by $\beta \leq \frac{1-\mathbf{E[p]}_j}{\mathbf{\Sigma}_{ij}^{1/2}}$. Note that since $\mathbf{E[p]}_i < 1$ and typically $\mathbf{\Sigma}_{ij}^{1/2} < \mathbf{E[p]}_i$, the value of β_{max} will generally be greater than 1. ∎

Appendix 2B: Dirichlet Distribution Parameter Identification

Given a Dirichlet distribution f_D for an N-dimensional state with probability given by $\mathbf{p} = [p_1, p_2, \ldots, p_N]^T$ and parameters (or can interpret them as *counts* of a particular transition) α,

$$f_D(\mathbf{p}|\alpha) = K \prod_{i=1}^{N} p_i^{\alpha_i - 1} = K\, p_1^{\alpha_1 - 1} p_2^{\alpha_2 - 1} \ldots (1 - \sum_{i=1}^{N-1} p_i)^{\alpha_N - 1} \qquad \text{(2B-1)}$$

the first and second moments can be derived as

$$\begin{aligned}\mathbf{E}[\mathbf{p}] &= [\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_N]^T \\ &= \frac{1}{\sum_{i=1}^{N} \alpha_i} [\alpha_1, \alpha_2, \ldots, \alpha_N]^T \\ &= \frac{1}{\alpha_0} [\alpha_1, \alpha_2, \ldots, \alpha_N]^T\end{aligned}$$

and

$$\begin{aligned}\mathbf{\Sigma} &= \mathbf{E}[(\mathbf{p} - \mathbf{E}[\mathbf{p}])(\mathbf{p} - \mathbf{E}[\mathbf{p}])^{\mathbf{T}}] \\ &= \frac{1}{\alpha_0^2(\alpha_0 + 1)} \begin{bmatrix} \alpha_1(\alpha_0 - \alpha_1) & -\alpha_1\alpha_2 & \ldots & -\alpha_1\alpha_N \\ -\alpha_2\alpha_1 & \alpha_2(\alpha_0 - \alpha_2) & \ldots & -\alpha_2\alpha_N \\ \vdots & & \ddots & \\ -\alpha_N\alpha_1 & -\alpha_N\alpha_2 & \ldots & \alpha_N(\alpha_0 - \alpha_N) \end{bmatrix}\end{aligned}$$

where $\alpha_0 = \sum_{i=1}^{N} \alpha_i$.

The parameter identification problem is as follows: **Given the mean E[p] and covariance $\mathbf{\Sigma}$ of the Dirichlet distribution, determine the parameters α.** A first approximation of the parameters can be made by observing that the trace of the covariance matrix $\mathbf{\Sigma}$ is given by

$$\begin{aligned}\mathrm{tr}(\mathbf{\Sigma}) &= \frac{1}{\alpha_0^2(\alpha_0 + 1)} \left(\alpha_0 \underbrace{\sum_i \alpha_i}_{=\alpha_0} - \sum_i \alpha_i^2 \right) \\ &= \frac{1}{\alpha_0^2(\alpha_0 + 1)} \left(\alpha_0^2 - \sum_i \alpha_i^2 \right)\end{aligned} \tag{2B-2}$$

However, since $\alpha_i = \alpha_0 \mathbf{E}[\mathbf{p}]$, then substituting this in obtain that

$$\mathrm{tr}(\mathbf{\Sigma}) = \frac{1}{\alpha_0 + 1} \left(1 - \mathbf{E}[\mathbf{p}]^T \mathbf{E}[\mathbf{p}]\right) \tag{2B-3}$$

in the following way

$$\begin{aligned} \alpha_0 &= \frac{1 - \mathbf{E}[\mathbf{p}]^T \mathbf{E}[\mathbf{p}]}{\mathrm{tr}(\boldsymbol{\Sigma})} - 1 \\ \alpha &= \alpha_0 \, \mathbf{E}[\mathbf{p}] \end{aligned} \tag{2B-4}$$

Note that these are *estimates* of the parameters. In order to obtain the *Maximum Likelihood* estimate of the parameters, $\hat{\alpha}$, we must first form the log likelihood of the Dirichlet distribution (see for example, Wicker [88], where N observations are made),

$$\begin{aligned} \mathcal{L}(\alpha \mid p) &= \log(f_D(\mathbf{p}|\alpha)) \\ &= N \left[\log(\Gamma(\alpha_0)) - \sum_k \log(\Gamma(\alpha_k)) + \sum_k (\alpha_k - 1) \log(p_k) \right] \end{aligned} \tag{2B-5}$$

and solve the optimization

$$\hat{\alpha} = \arg\max_{\alpha} \mathcal{L}(\alpha \mid p) \tag{2B-6}$$

The log likelihood is globally concave (since the Dirichlet belongs to the exponential distribution), and a solution to this optimization problem is globally optimal, and furthermore, can be found for example by using a Newton-Raphson method. However, we have noted in our work, that using the two-moment approximation (without the need for the optimization) provides very accurate values for the parameters.

Chapter 3

Hybrid Estimation with Model Uncertainty

This chapter addresses the role of Markov Chain uncertainty in a common class of stochastic hybrid estimation problems. The key distinction from the previous chapter is that the performance loss in this class of problems is the estimation inefficiency that arises from the uncertainty in the transition probabilities. In this chapter, the *state* of the system is more general that than of the previous chapter, in that it is composed of both a continuous and a discrete set of dynamics.

This chapter presents two key results. First, we show that uncertainty in the transition model can lead to covariance mismatches. This is an extension to previous work that only considered estimation bias in the case of uncertain transition models. An important effect of mismatched covariances is that they can lead to overconfident estimates, and ultimately lead to large estimation errors. An example of this is in the context of a UAV multi-target tracking problem, where covariance underestimation can lead to unacceptable estimation errors. Our second main result is the development of an algorithm that explicitly accounts for the uncertainty in the transition probabilities and hedges against the overconfidence phenomenon. This new Robust Multiple Model filter shows improved tracking performance in the presence of this uncertainty.

3.1 Introduction

3.1.1 Previous Work

A broad range of modern systems can be modeled as hybrid systems, or systems that have both a continuous and discrete set of dynamics [81]. A common example of a *stochastic* hybrid system is a Jump Markov Linear System, which is composed of a finite set of dynamic models, and at any given time, the switch between the different dynamic models (or "modes") is modeled by a Markov Chain with a known probability transition matrix. This chapter focuses on these types of systems, as they are fairly general models for a broad range of applications. In the engineering community, for example, hybrid systems show up in sensor management problems [36], Air Traffic Control [4], failure detection [65, 87, 89] and diagnostics [37], Bayesian tracking [38, 48, 67], and in underwater applications, such as tracking jellyfish [73]. The medical community has applied hybrid models to tracking ventricular motion from ultrasound [68] and tumors [77].

Multiple model estimation is used to find the state estimate and covariance for stochastic hybrid systems [4, 40, 41]. There are numerous techniques in the literature for tackling this challenging problem [57, 63]. The Interacting Multiple Model (IMM) [4, 15, 56] and Generalized Pseudo Bayesian (GPB) estimators are two popular implementations of multiple model filters and it has been shown that under certain conditions, these filters can significantly outperform individually tuned Kalman filters [4]. These empirical results generally assume that the probability transition matrix is available to the estimator designer. In reality there may be insufficient data to justify this assumption, or the transition model may simply not be available to the estimator designer at all. It has been recently shown that multiple model estimators may be sensitive to the transition parameters of the Markov Chain, and that uncertainty in the transition model can lead to biased nominal estimates [27, 46]. This chapter extends these results to the case of the covariance mismatch problems that can in turn lead to estimation errors.

Accounting for uncertainty in the transition probabilities is not a new problem.

However, the main emphasis in the estimation community has been the identification of the uncertain probability transition model. For example, Tugnait [82] considers linear systems with stochastic jump parameters. The linear dynamic model is a function of an unknown, but stationary probability transition matrix which is estimated by online observations using a truncated maximum likelihood technique; Tugnait shows that this estimate of the probability transition matrix converges after the system achieves quiescence.

More recently, Jilkov and Li [46] and Doucet and Ristic [27] have considered the problem of probability transition matrix identification using noisy observations, and empirically show the estimation bias that can occur from the unknown transition matrix. Jilkov and Li propose new algorithms for identifying the most likely transition model driving the Markov Chain of the system, $\hat{\Pi}$. Their work relies on a certainty equivalence-like approximation where at each time step, the most likely estimate of the transition model is used to update the state estimate and covariance. Doucet [27] presents an analytical approach for identifying the transition probabilities using a Dirichlet model to estimate the probability transition matrix. While Doucet [27] and Jilkov [46] assume the transition matrix is unknown and develop a systematic identification process to reduce the uncertainty, they do not consider the impact of the uncertain transition model $\tilde{\Pi}$ on the covariance mismatch problem, which is one of the results of this work.

3.1.2 Outline

This chapter is organized as follows: Section 3.2 reviews multiple model estimation and Section 3.3 discusses the issue of transition probability uncertainty in multiple model estimation and describes the covariance mismatch problem. We introduce the Robust Multiple Model Filter in Section 3.4 and present some conditions for covariance underestimation. Some numerical results are presented in Section 3.6 in the context of a UAV multi-target tracking problem.

3.2 Background

Linear multiple model estimation assumes that a dynamic system is driven by a unique set of N_m different dynamic models, but the system actually is in mode i at some time k. Each dynamic model is described by a different system with state $x_k^i \in \Re^N$

$$x_{k+1}^i = \Phi^i x_k^i + G^i u_k^i + w_k^i, \quad z_k = H^i x_k + v_k^i \tag{3-1}$$

A noisy measurement $z_k \in \Re^{N_o}$ is available at time k. For each model i, the system matrices $\Phi^i \in \Re^{N \times N}$, $G^i \in \Re^{N \times N_u}$, $H^i \in \Re^{N_o \times N}$ and control inputs $u^i \in \Re^{N_u}$ are assumed known.[1] The noise term w_k^i (respectively, v_k^i) is zero mean, Gaussian, $w_k^i \sim N(0, Q^i)$ (respectively, $v_k^i \sim N(0, R^i)$). At a time increment from k to $k+1$, the system can transition from mode i to a mode j according to the probability transition matrix $\Pi \in \Re^{N_m \times N_m}$. The probability transition matrix is a stochastic matrix with column sums equal to unity, and each entry satisfies the definition of a probability: $0 \leq \pi_{ij} \leq 1$. The current mode of the system is not directly observable due to the noisy measurements z_k. Hence the current model is only known with some probability, $\mu_{k|j}^i$, which denotes the probability of being in model i at time k given the information at time j. Note that the key difference between the transition matrix in this chapter is that it does not depend on the control input, whereas in the previous chapter, the transition matrix depended on the control action.

It turns out that the *optimal* multiple model filter cannot be realized in practice since this requires keeping track of a combinatorial number of mode transitions of the system throughout the course of the estimation process. As a result, one generally resorts to suboptimal schemes such as the Generalized Pseudo Bayesian (GPB) and Interacting Multiple Model (IMM) to overcome this complexity. [4] The prediction and measurement updates for the GPB1 filter are shown in Table 3.1 and a diagram of a GPB1 implementation is shown in Figure 3-1.[2] The state estimate, $\hat{x}_{k+1|k+1}^i$,

[1] Here N_o is the dimension of the observation vector, N_u is the dimension of the control input, and N is the dimension of the state vector.

[2] Note that the GPB1 estimator is one of the many forms of the GPB estimator and we use it to

Table 3.1: Estimation Steps for a GPB1 implementation showing the prediction and measurement update steps for both the probabilities (left) and dynamic models (right). Note that each estimator cycle, the estimator for each model i is re-initialized with the combined estimate $\hat{x}_{k+1|k+1}$ and covariance $P_{k+1|k+1}$.

	Probabilities	Model
Propagation step:	$\mu_{k+1\|k} = \Pi\mu_{k\|k}$	$\hat{x}^i_{k+1\|k} = \Phi^i\hat{x}_{k\|k} + G^i u^i_k$ $P^i_{k+1\|k} = \Phi^i P_{k\|k}(\Phi^i)^T + Q^i$
Measurement update:	$\mu^j_{k+1\|k+1} \propto \Lambda^j_k \mu^j_{k+1\|k} \quad \forall j$	$\hat{x}^i_{k+1\|k+1} = \hat{x}^i_{k+1\|k} + W^i_k(z_k - \hat{z}^i_k)$ $P^i_{k+1\|k+1} = P^i_{k+1\|k} - W^i_k S^i_k W^{i,T}_k$

error covariance $P^i_{k+1|k+1}$, and the probability $\mu^i_{k+1|k+1}$ are computed recursively for each model i. For linear systems, each filter estimate and error covariance is the output of a Kalman filter tuned to each particular model. The probability updates are shown in the left part of the table, and the state estimate and covariance updates are shown in the right hand side.[3] Note that just as in classical estimation, there is a propagation and measurement update step for both the continuous state and the probabilities.

In order to maintain computational tractability (and avoid the combinatorial explosion of maintaining all possible mode sequences), suboptimal filters (such as the GPB and IMM) then approximate the combined posterior distribution from the individual Kalman filters, into a single Gaussian distribution with mean $\hat{x}_{k+1|k+1}$ and covariance $P_{k+1|k+1}$. This process relies on a two-moment approximation of a mixture of Gaussians, and the following expression can be derived [4] for the *combined* state estimate $\hat{x}_{k+1|k+1}$ and *combined* covariance $P_{k+1|k+1}$

$$\hat{x}_{k+1|k+1} = \sum_i \mu^i_{k+1|k+1}\hat{x}^i_{k+1|k+1} \tag{3-2}$$

$$\begin{aligned} P_{k+1|k+1} &= \sum_i \mu^i_{k+1|k+1}\left[P^i_{k+1|k+1} + (\hat{x}^i_{k+1|k+1} - \hat{x}_{k+1|k+1})(\hat{x}^i_{k+1|k+1} - \hat{x}_{k+1|k+1})^T\right] \\ &= \left[\sum_i \mu^i_{k+1|k+1}\left(P^i_{k+1|k+1} + \hat{x}^i_{k+1|k+1}\hat{x}^i_{k+1|k+1}\right)\right] - X \end{aligned} \tag{3-3}$$

highlight the key features of the covariance mismatch problems in the following sections.

[3]The parameters S^i_k and the Kalman filter gain W^i_k are given by $S^i_k = H^i P^i_{k+1|k}(H^i)^T + R^i$, $W^i_k = P^i_{k+1|k}(H^i)^T S^{i,-1}_k$, and Λ^i_k is the likelihood function.

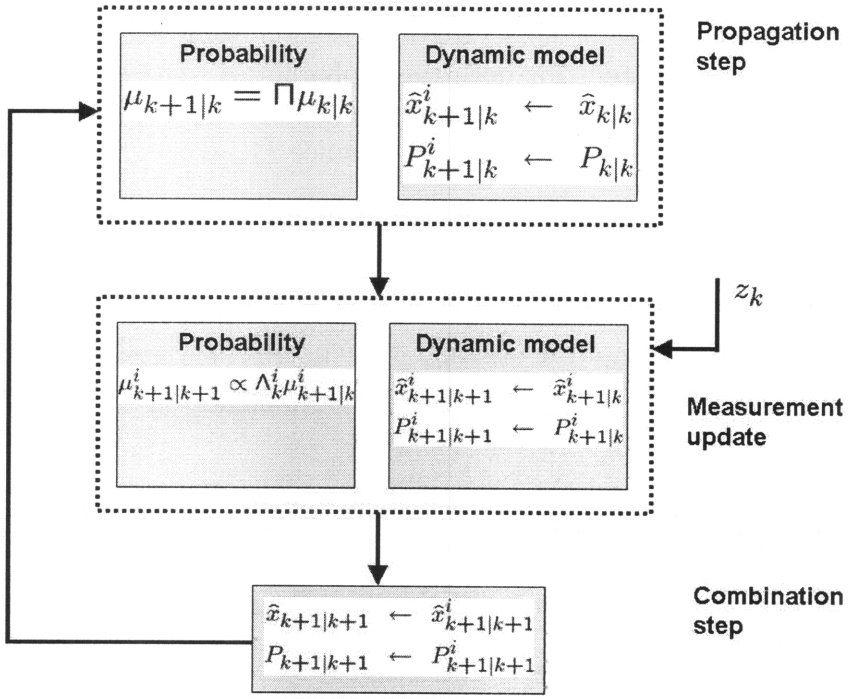


Fig. 3-1: Multiple Model updates (propagation, measurement update, and combination) for a Generalized PseudoBayesian formulation.

where $X = (\hat{x}_{k+1|k+1})(\hat{x}_{k+1|k+1})^T$. The GPB1 formulation then uses this combined estimate and covariance as a *common* initial condition for the Kalman filters at the next estimator cycle (see Figure 3-1).

3.3 Model uncertainty and covariance mismatch

Multiple model filters are parametrized by many different quantities that are potentially uncertain or unknown, such as the statistics of the process noise Q^i and the dynamic models Φ^i. In this chapter, we are principally concerned with the uncertainty of the transition matrix Π. We motivate the importance of accounting for this uncertainty in this section, and show how it ultimately leads to both biased nominal estimates and mismatched covariances through the combination step of Equations 3-2 and 3-3.

3.3.1 Source of Uncertainty in Π

While the transition probability matrix inherently captures uncertainty in mode transitions by the use of probabilities, these probabilities are generally the outcome of an estimation process themselves [46]. For example, the frequentist interpretation of these probabilities is that they are calculated by normalizing the counted mode transitions n_{ij} by the total number of transitions N_j

$$\hat{\pi}_{ij} = \frac{n_{ij}}{N_j}, \quad N_j = \sum_i n_{ij} \tag{3-4}$$

In practice, this counting process requires a large number of observed transitions before the estimated probability $\hat{\pi}_{ij}$ converges to the true probability, $\pi_{ij} = \lim_{N_i \to \infty} \hat{\pi}_{ij}$. Hence, with a small number of observations, the transition probabilities themselves can be thought of as uncertain parameters of the multiple model estimator. Furthermore, even if an estimate of the transition probabilities were available for a complex systems, it is unlikely that this estimate would be precisely matched to the true underlying stochastic process.

3.3.2 Covariance Mismatch

We next provide the key steps showing the impact of an uncertain probability transition matrix in the overall multiple model estimator. First, we express the uncertain probability transition matrix $\tilde{\Pi}$ as a sum of a nominal probability transition matrix $\bar{\Pi}$, and a perturbation Δ_Π: $\tilde{\Pi} = \bar{\Pi} + \Delta_\Pi$.

$$\begin{bmatrix} \tilde{\pi}_{1,1} & \tilde{\pi}_{1,2} & \dots & \tilde{\pi}_{1,N} \\ \tilde{\pi}_{2,1} & \tilde{\pi}_{2,2} & \dots & \tilde{\pi}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\pi}_{N,1} & \tilde{\pi}_{N,2} & \dots & \tilde{\pi}_{N,N} \end{bmatrix} = \begin{bmatrix} \bar{\pi}_{1,1} & \bar{\pi}_{1,2} & \dots & \bar{\pi}_{1,N} \\ \bar{\pi}_{2,1} & \bar{\pi}_{2,2} & \dots & \bar{\pi}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\pi}_{N,1} & \bar{\pi}_{N,2} & \dots & \bar{\pi}_{N,N} \end{bmatrix} + \begin{bmatrix} \Delta_\Pi(1,1) & \Delta_\Pi(1,2) & \dots & \Delta_\Pi(1,N) \\ \Delta_\Pi(2,1) & \Delta_\Pi(2,2) & \dots & \Delta_\Pi(2,N) \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_\Pi(N,1) & \Delta_\Pi(N,2) & \dots & \Delta_\Pi(N,N) \end{bmatrix} \tag{3-5}$$

Note that each column of the perturbation matrix Δ_Π has to sum to 0 to ensure that the perturbed transition matrix $\tilde{\Pi}$ is a proper probability transition matrix with column sums equal to unity. The probability propagation step (see Table 3.1) can then be written out as

$$\tilde{\mu}_{k+1|k} = \tilde{\Pi}\mu_{k|k} = \bar{\Pi}\mu_{k|k} + \Delta_\Pi \mu_{k|k} = \bar{\mu}_{k+1|k} + \Delta_{k+1|k} \tag{3-6}$$

which shows that the uncertainty in the transition model has impacted the propagated probabilities $\tilde{\mu}_{k+1|k}$. Note that the probabilities have been expressed as a sum of a nominal term $\bar{\mu}_{k+1|k}$ and a deviation $\Delta_{k+1|k}$. These propagated probabilities are then updated with the likelihood function Λ_k^i in the measurement update step (see Table 3.1), renormalized, and result in uncertain posterior probabilities

$$\tilde{\mu}_{k+1|k+1} = \bar{\mu}_{k+1|k+1} + \Delta, \quad \tilde{\mu}_{k+1|k+1} \in \mathcal{M}_{k+1} \tag{3-7}$$

where Δ is the perturbation in the posterior probabilities and $\mathcal{M}_{k+1}$ is the uncertainty set for the posterior probabilities. For clarity of exposition, we delay remarking on how this set is found until the end of this section.

When the uncertain posterior probabilities are incorporated in the combination step, they perturb the combined estimate and covariance, $\hat{x}'_{k+1|k+1}$ and $P'_{k+1|k+1}$

$$\begin{aligned} \hat{x}'_{k+1|k+1} &= \sum_i (\bar{\mu}^i_{k+1|k+1} + \Delta^i)\hat{x}^i_{k+1|k+1} \\ &= \hat{x}_{k+1|k+1} + \Delta_x \end{aligned} \tag{3-8}$$

$$\begin{aligned} P'_{k+1|k+1} &= \sum_i (\bar{\mu}^i_{k+1|k+1} + \Delta^i)\{P^i + \hat{x}^i_{k+1|k+1}\,(\hat{x}^i_{k+1|k+1})^T\} - (\hat{x}'_{k+1|k+1})(\hat{x}'_{k+1|k+1})^T \\ &= P_{k+1|k+1} + \Delta_p \end{aligned} \tag{3-9}$$

Here $\Delta_x \in \Re^N$ and $\Delta_p \in \Re^{N\times N}$ are the respective perturbations from the nominal state estimate and covariance, $\hat{x}_{k+1|k+1}$ and $P_{k+1|k+1}$.

In summary, the uncertainty in the transition matrix Δ_Π generates uncertainty

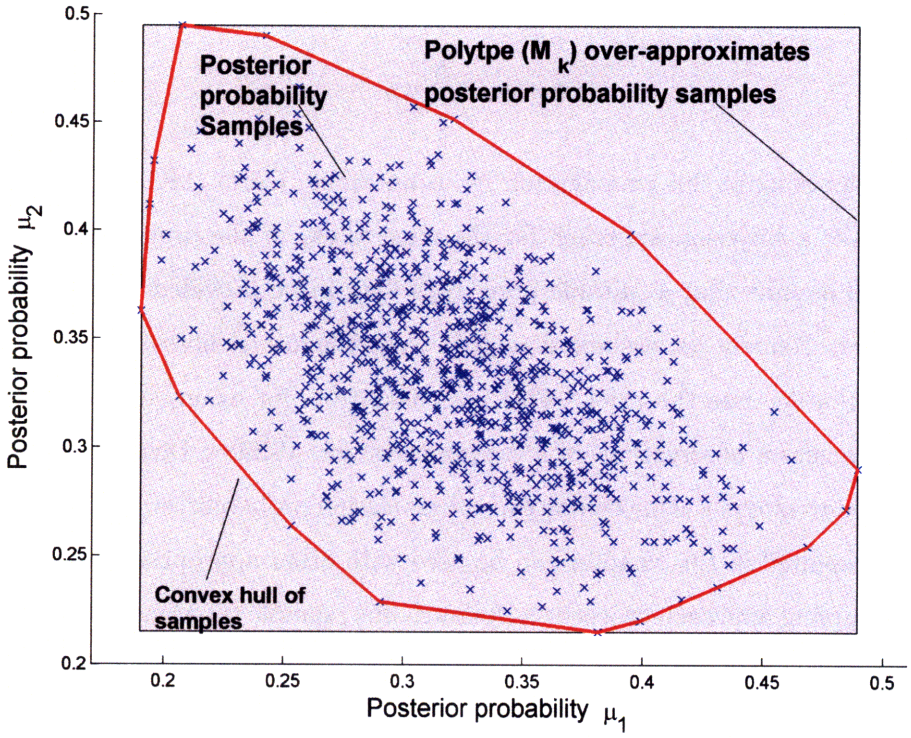


Fig. 3-2: Visualization of the sampling component of the Robust Multiple Model algorithm. The samples of the posterior probabilities (blue) are overapproximated by both the convex hull (red) and a polytope (gray) which is the definition of the uncertainty set $\mathcal{M}_k$.

in the posterior probabilities $\Delta \neq 0$, which ultimately gives rise to a biased nominal combined estimate ($\Delta_x \neq 0$) and a mismatched covariance $\Delta_p \neq 0$. Alternatively, one can think of this as a Gaussian mixture problem with uncertain weights. We will address the case of $\Delta_p \neq 0$ in Section 3.4.

3.3.3 Constructing Uncertainty Set $\mathcal{M}_{k+1}$ using Sampling

It is not generally possible to propagate the effect of the uncertain probability transition model $\tilde{\Pi}$ to the posterior probabilities $\tilde{\mu}_{k+1|k+1}$ of Eq. 3-7 in closed form. That is, in general there is no analytical expression for finding the posterior uncertainty

set of $\tilde{\mu}_{k+1|k}$ in an exact form

$$\tilde{\mu}_{k+1|k} = \tilde{\Pi}\tilde{\mu}_{k|k} \tag{3-10}$$

We thus approximate this propagation by using Monte Carlo (MC) sampling methods. We take a Bayesian approach for the uncertainty in the probability transition matrix, and assume that a suitable prior $f_D(p \mid \alpha)$can be provided on the transition probabilities. Namely, we assume $\tilde{\pi} \sim f_D(p \mid \alpha)$, where α denotes the hyperparameters that characterize the prior. This prior could be, for example, the outcome of previous transition observations of a dynamic system. While a Dirichlet prior is frequently used to model the uncertainty in the transition probabilities, [33] the sampling strategy presented in this chapter can be used with other appropriate priors.

The sampling approach proceeds as follows: first, sample the transition probability from its prior, and obtain N_s unique samples of the probability transition model, $\tilde{\Pi}^s$. Thus, the propagation step can be approximated as

$$\tilde{\mu}^s_{k+1|k} = \tilde{\Pi}^s \mu_{k|k}, \quad \forall s = 1, \ldots, N_s \tag{3-11}$$

We can then perform the mode probability update step for each sample s, and using the likelihood Λ^j_k (from the measurement update of Table 3.1), the posterior samples of the probabilities are proportional to the product of the propagated samples $\mu^{j,s}_{k+1|k}$ and the likelihood

$$\tilde{\mu}^{j,s}_{k+1|k+1} \propto \Lambda^j_k \, \tilde{\mu}^{j,s}_{k+1|k}, \quad \forall j, \; s \tag{3-12}$$

Note that this is done for all samples s and for all models j. The posterior probability samples $\tilde{\mu}^{j,s}_{k+1|k+1}$ are then normalized for all N_s realizations.

The uncertainty set $\mathcal{M}_{k+1}$ is then constructed from some approximation of the posterior probability samples $\tilde{\mu}^s_{k+1|k+1}$. Examples of the sampling scheme and uncertainty sets $\mathcal{M}_{k+1}$ are shown in Figure 3-2, where the posterior probability samples (blue) are over approximated by both an appropriately chosen credibility region (red

line) and a polytope (gray). Note that either of these choices results in a convex uncertainty set. Furthermore, the polytope can be described in terms of the maxima and minima of the samples, where the minimum and maximum of $\mu^{j,s}_{k+1|k+1}$ for all realizations s, that is $\mu^- = \min_s\{\mu^{j,s}_{k+1|k+1}\}$ and $\mu^+ = \max_s\{\mu^{j,s}_{k+1|k+1}\}$, and the uncertainty set $\mathcal{M}_{k+1}$ is defined as

$$\mathcal{M}_{k+1} = \{\tilde{\mu}_{k+1|k+1} \mid \mu^- \leq \tilde{\mu}_{k+1|k+1} \leq \mu^+\} \tag{3-13}$$

Note that this polytope over-approximates the posterior probability, and the computational effort increases with an increased number of samples. We show some preliminary computational results in Section 3.6.

3.4 Robustness in Hybrid Estimation

As shown in the last section, a key issue resulting from the transition model uncertainty is that it affects the combination step of the filter, by introducing uncertainty in the posterior probabilities, and in turn, generating biased nominal estimates and mismatched combined covariances. In this work, we are primarily concerned with the covariance underestimation problem, where multiple model filters can over-predict their confidence and ultimately accrue large estimation errors. In this section, we introduce the Robust Multiple Model filter (RMM), which mitigates the problem of covariance underestimation by finding the largest covariance matrix given the uncertainty description $\mathcal{M}_{k+1}$ for the posterior probabilities.

3.4.1 Problem Statement

Recall that the covariance mismatch problem arises when the perturbed combined covariance ($P'_{k+1|k+1}$, see Eq. 3-9) differs from the nominal combined covariance $P_{k+1|k+1}$ due to the uncertainty in the posterior probabilities $\tilde{\mu}_{k+1|k+1} \in \mathcal{M}_{k+1}$. Furthermore, by simply using mismatched probabilities $\mu'_{k+1|k+1}$ (arising from a mismatched probability transition model) in the combination step, the estimator effectively ignores any

uncertainty in the probabilities themselves, and this in turn can cause the perturbed covariance to be smaller than the nominal, $P'_{k+1|k+1} < P_{k+1|k+1}$. By underestimating the covariance, the estimator is in fact overconfident, and for our applications, overconfidence is an undesirable prospect.

The main idea of our approach is to use the entire uncertainty set $\mathcal{M}_{k+1}$ of the posterior probabilities to construct a combined covariance that is not underestimated, by solving for the largest combined covariance $P'_{k+1|k+1}$ admissible from *any* uncertain posterior probabilities in the uncertainty set $\tilde{\mu} \in \mathcal{M}_{k+1}$. We quantify the *size* of the covariance matrix by using the *trace* of the combined covariance matrix. In other words, this optimization finds the maximum mean square error that could result from the uncertain posterior probabilities $\tilde{\mu}_{k+1|k+1} \in \mathcal{M}_{k+1}$. Note that maximizing the trace is a fairly standard approach in estimation, as it forms the basis for the Kalman filter. Trace maximization is also used in robust estimation problems [34]. We summarize our goal in the following problem statement:

$$\begin{aligned} &\textit{Find the combined covariance } P'_{k+1|k+1} \textit{ with the maximal trace} \\ &T^*_{k+1|k+1} = \max_{\tilde{\mu}} \left(\text{Trace } P'_{k+1|k+1}(\tilde{\mu})\right) \text{ subject to: } \tilde{\mu} \in \mathcal{M}_{k+1} \end{aligned} \tag{3-14}$$

where

$$\begin{aligned} \hat{x}_{k+1|k+1}(\tilde{\mu}) &= \sum_i \tilde{\mu}^i_{k+1|k+1} \hat{x}^j_{k+1|k+1} \\ P'_{k+1|k+1}(\tilde{\mu}) &= \sum_i \tilde{\mu}^i_{k+1|k+1} \{P^i_{k+1|k+1} + \hat{x}^i_{k+1|k+1} \, (\hat{x}^i_{k+1|k+1})^T\} - \hat{x}_{k+1|k+1}(\tilde{\mu})(\hat{x}_{k+1|k+1}(\tilde{\mu}))^T \end{aligned}$$

The probabilities $\tilde{\mu}^*$ that corresponds to this robust covariance are found with $\tilde{\mu}^* = \arg\max T^*_{k+1|k+1}$.

Remark: A game theoretic interpretation for this optimization is as follows. First note that the Kalman filters for each of the i models are the *minimum variance* estimators for each model [32], and the combination step merges the optimal estimates and variances into a single combined state estimate $\hat{x}_{k+1|k+1}$ and covariance $P_{k+1|k+1}$ using moment-matching. This combination can be loosely interpreted as outputting

the minimum *combined covariance*, conditioned on the probabilities $\mu_{k+1|k+1}$. Since the uncertainty in the posterior probabilities can cause $P_{k+1|k+1}$ to be mismatched, the goal of this optimization is maximize the minimum combined covariance.

3.4.2 Finding the maximum trace

Due to the linearity of the trace operator, the following proposition states that the optimization in Eq. 3-14 can be solved using a quadratic program:

Proposition 9 *The trace of $P'_{k+1|k+1}$ is quadratic in $\tilde{\mu}$, and can be solved with the following quadratic program*

$$T^*_{k+1|k+1} = \max_{\tilde{\mu}} \left(-\tilde{\mu}^T A_{k+1}\tilde{\mu} + B_{k+1}\tilde{\mu}\right) \text{ subject to: } \quad \tilde{\mu} \in \mathcal{M}_{k+1} \tag{3-15}$$

where

$$\begin{aligned} A_{k+1}(j,m) &\doteq \mathit{Trace}\{\hat{x}^j_{k+1|k+1}(\hat{x}^m_{k+1|k+1})^T\}, \quad \forall j,m \\ B_{k+1}(j) &\doteq \mathit{Trace}\,\{P^j + \hat{x}^j_{k+1|k+1}\,(\hat{x}^j_{k+1|k+1})^T\}, \quad \forall j \end{aligned}$$

$A_{k+1} \in \Re^{N\times N}$ and $B_{k+1} \in \Re^N$ are a function of the estimate and covariance of each model. Furthermore, $A_{k+1} \succ 0$.

We summarize this result by noting that since the quadratic objective is convex ($A_{k+1} \succ 0$), and since $\mathcal{M}_{k+1}$ is a convex uncertainty set, then this is a convex optimization with a global maximum. In fact, this guarantees that the maximal trace of $P'_{k+1|k+1}(\tilde{\mu})$ is unique.

Solving for the robust covariance gives rise to our Robust Multiple Model filter, where the algorithm is summarized in Figure 3-3. All steps are identical to the classical MM filter (see Figure 3-1) except for the robust combination step, where we solve for the maximum trace of the combined covariance $P'_{k+1|k+1}$. (Note that this must be done at each time step as the estimates $\hat{x}^i_{k+1|k+1}$ and covariance $P^i_{k+1|k+1}$ are time-varying.)

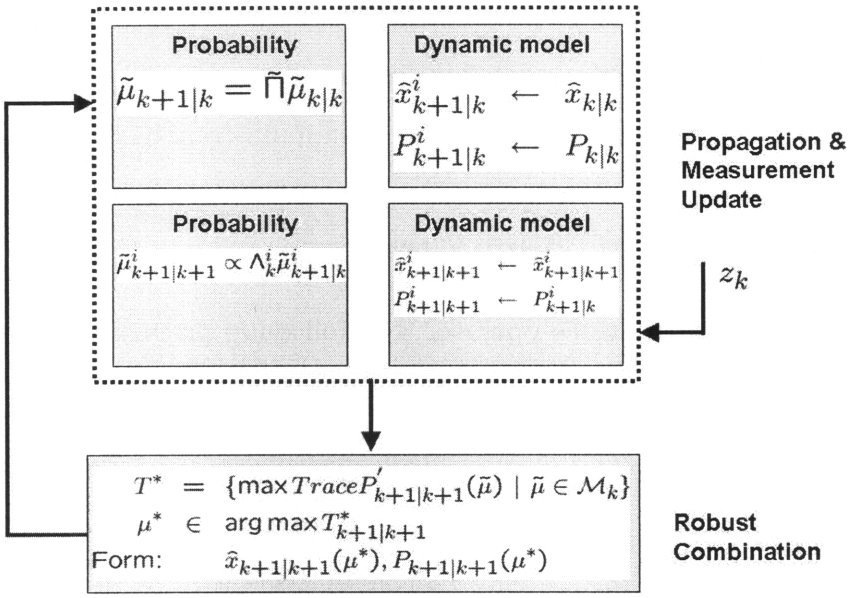


Fig. 3-3: Diagram of the feedback implementation Robust Multiple Model updates (note in particular the robust combination step) for a GPB1 formulation.

To complete the Robust Multiple Model filter, recall that the GPB1 implementation of the MM filter feeds back the combined estimate and covariance at the beginning of each estimator cycle. For the Robust MM filter, a feedback formulation, feeds back the the *robust* estimate $\hat{x}_{k+1|k+1}(\tilde{\mu}^*)$ and robust covariance $P'_{k+1|k+1}(\tilde{\mu}^*)$ at the beginning the estimator cycle

$$\begin{aligned}
\tilde{\mu}^* &= \arg\max_{\tilde{\mu}} T^*_{k+1|k+1} \\
\hat{x}_{k+1|k+1}(\tilde{\mu}^*) &= \sum_i \tilde{\mu}^{*,i}_{k+1|k+1} \hat{x}^j_{k+1|k+1} \\
P'_{k+1|k+1}(\tilde{\mu}^*) &= \sum_i \tilde{\mu}^{*,i}_{k+1|k+1} \{P^i_{k+1|k+1} + \hat{x}^i_{k+1|k+1} \, (\hat{x}^i_{k+1|k+1})^T\} - \hat{x}_{k+1|k+1}(\tilde{\mu}^*)(\hat{x}_{k+1|k+1}(\tilde{\mu}^*))^T
\end{aligned} \tag{3-16}$$

(An alternative implementation is use the robust estimate and covariance only as *outputs* with which to measure the effect of the uncertainty in the transition probability matrix on the the combined covariance.) Note that if the transition probabilities are completely *unknown*, the robust MM filter is effectively a *worst-case* filter. In this situation, the only requirement on the uncertainty set $\mathcal{M}_{k+1}$ is that each entry is bounded and non-zero, $0 \leq \mu^i \leq 1$, subject to the unit sum constraint $\sum_i \mu^i = 1$.

3.4.3 Summary

In summary, the Robust Multiple Model algorithm (RMM) has similar prediction and measurement update steps as a conventional multiple model algorithm. There are however two key differences from a conventional multiple model estimator. First, the RMM *solves for* the combined covariance $P'_{k+1|k+1}$ with the *maximal trace* rather than simply computing the covariance from the posterior probabilities, as these probabilities are now uncertain. This ensures that the trace of the covariance under some uncertainty in the probability transition matrix is not underestimated. Secondly, the RMM requires an uncertainty set for the posterior probabilities $\mathcal{M}_{k+1}$; this uncertainty set is obtained from numerical sampling. If the uncertainty set is unavailable, we have also remarked on a *worst case* estimator, where the uncertainty set is the most conservative set over the entire probability simplex, and constrains the probabilities to their definition: namely, being between 0 and 1, and summing to unity.

3.5 Sampling with the Dirichlet Sigma Points

If the prior on the transition probability is described by a Dirichlet density, the results from the previous chapter on Dirichlet Sigma Points are applicable, and we can find the uncertainty set $\mathcal{M}_{k+1}$ using a much smaller number of scenarios, which in turn leads to an economical approach to find the robust covariance.

Recall that for a row $\mathbf{p}$ of the transition probability matrix, the Dirichlet density is defined as

$$f_D(\mathbf{p}|\alpha) = K \prod_{i=1}^{N} p_i^{\alpha_i - 1}, \quad \sum_{i}^{N} p_i = 1, \quad 0 \leq p_i \leq 1 \tag{3-17}$$

and the corresponding Dirichlet Sigma Points $\mathcal{Y}_i \in \Re^N$ are defined as

$$\begin{aligned}
\mathcal{Y}_0 &= \mathbf{E}[\mathbf{p}] \\
\mathcal{Y}_i &= \mathbf{E}[\mathbf{p}] + \beta_{\max} \left(\boldsymbol{\Sigma}^{1/2}\right)_i \; \forall i = 1, \ldots, N \\
\mathcal{Y}_i &= \mathbf{E}[\mathbf{p}] - \beta_{\max} \left(\boldsymbol{\Sigma}^{1/2}\right)_i \; \forall i = N+1, \ldots, 2N
\end{aligned}$$

where $\beta_{\max}$ is a tuning parameter that reflects how much uncertainty is desired in the Dirichlet Sigma Points. Each of these Dirichlet Sigma Points correspond to individual realizations of the *row* of the transition probability matrix, and the complete realization s of the transition probability *matrix* is given by

$$\tilde{\Pi}^s = \begin{bmatrix} \mathcal{Y}_s^1 \\ \mathcal{Y}_s^2 \\ \cdots \\ \mathcal{Y}_s^N \end{bmatrix}, \tag{3-18}$$

The complete sequence of iterations is to propagate each of these samples through the prediction and measurement update steps,

$$\tilde{\mu}_{k+1|k}^s = \tilde{\Pi}^s \mu_{k|k}, \quad \forall s = 1, \ldots, N_s \tag{3-19}$$

where $\tilde{\mu}^s \in \Re^N$ is the full probability vector. Each of the elements j of this probability vector, for each s realization, is updated as

$$\tilde{\mu}_{k+1|k+1}^{j,s} \propto \Lambda_k^j \, \tilde{\mu}_{k+1|k}^{j,s}, \quad \forall j, \; s \tag{3-20}$$

We will demonstrate the computational advantages of using the Dirichlet Sigma Points in the next numerical section.

3.6 Numerical results

We present results on the impact on transition model uncertainty, and benefits of robust multiple model estimation, in two different tracking problems. In the first example, we consider a single UAV, multi-target tracking problem (see Figure 3-4). In the second example, we revisit a slight variation of a tracking problem originally analyzed in Jilkov and Li [46] and Doucet and Ristic [27].

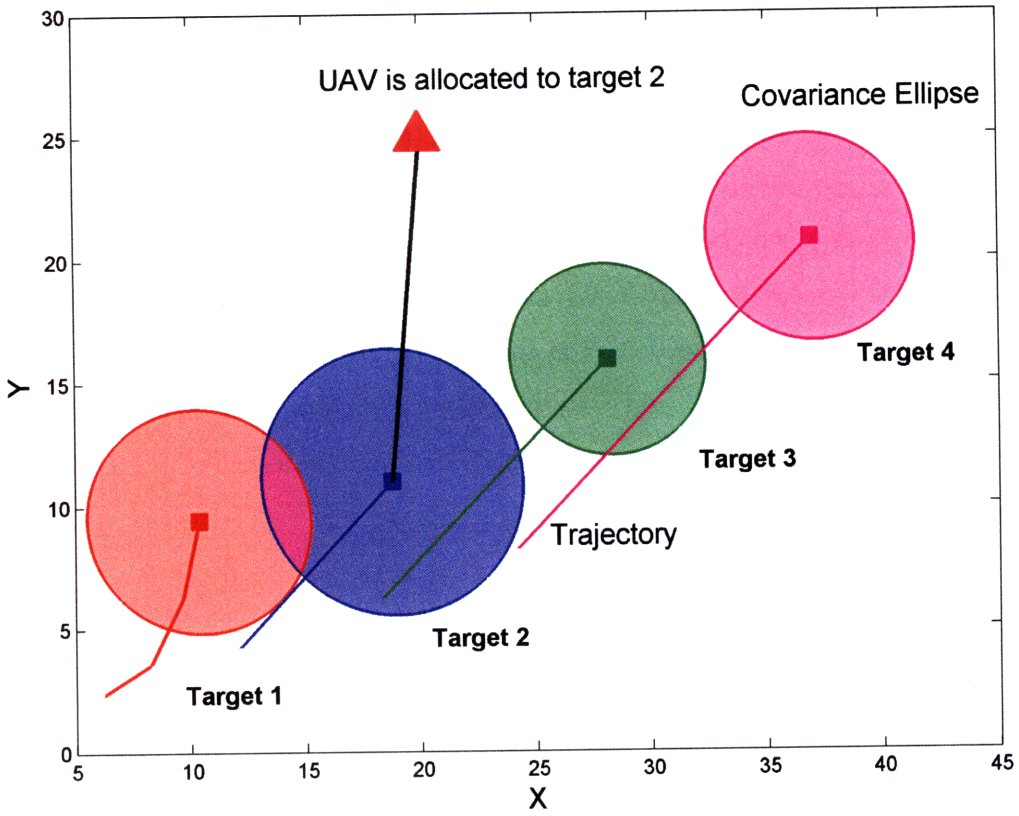


Fig. 3-4: Problem setup: 1 UAV needs to track 4 targets. Due to the resource constraint and in order to model the non-zero flying time to reach a target, the UAV is allocated among the 4 different targets when the trace of the target's covariance exceeds a threshold γ. Each of the targets is modeled with 4 distinct dynamics models: constant velocity, constant x-acceleration, constant y-acceleration, acceleration in both x and y.

3.6.1 UAV Tracking Problem

In the first example, we consider a UAV with a multiple model filter that has to maintain a constant track on $N_v \gg 1$ unique targets (see Figure 3-4). Each j^{th} target can be described by a unique set of N_m kinematic models $\{\Phi^i, G^i, Q^i, \Pi^i\}, \forall i = 1, \ldots, N_m$. Each target can have a unique probability transition matrix Π^i between the different models. The data association problem is not considered in these simulations.

Since there is only a single UAV tracking 4 targets, the UAV has to be allocated among the different targets in order to maintain good estimates of target states; to model the non-zero flight time it takes for a UAV to fly to a different target, the allocation is performed according to the following, simple rule: *if the trace of the combined covariance of any target exceeds a threshold γ, revisit that target.* For these numerical simulations, we used $\gamma = 100$ and $\gamma = 500$ meters.

We considered both a 2- and 4-state kinematic models. For the 2-state problem, each target's state (position and velocity) is denoted as $x_k = [x \;\; v_x]$ and the kinematics are

$$\Phi^i = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}, \quad G^i = \begin{bmatrix} \Delta T^2/2 \\ 0 \end{bmatrix}, \quad u_k^i = \begin{bmatrix} 0 & 2 \end{bmatrix} \qquad [m/s^2] \qquad (3\text{-}21)$$

where $\Delta T = 1$ for both the 2-state and 4-state models.

For the 4-state problem, each target's state is denoted as $x_k = [x \;\; y \;\; v_x \;\; v_y]$, and the targets operate under the following set of kinematics

$$\Phi^i = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad G^i = \begin{bmatrix} \Delta T^2/2 & 0 \\ 0 & \Delta T^2/2 \\ \Delta T & 0 \\ 0 & \Delta T \end{bmatrix}, \quad u_k^i = \begin{bmatrix} 0 & 2 & 0 & 2 \\ 0 & 0 & 2 & 2 \end{bmatrix}$$

The four different target control inputs u_k^i were modeled as follows (where each column of u_k^i is the different model): *i*) constant velocity; *ii*) acceleration in the x direction; *iii*) acceleration in y direction; *iv*) acceleration in both x and y. The probability

transition matrix was modeled with a Dirichlet prior, with a maximum likelihood value of

$$\hat{\Pi} = \begin{bmatrix} 0.375 & 0.11 & 0.125 & 0.18 \\ 0.125 & 0.56 & 0.25 & 0.18 \\ 0.25 & 0.22 & 0.50 & 0.18 \\ 0.25 & 0.11 & 0.125 & 0.46 \end{bmatrix} \tag{3-22}$$

The actual target model is the maximum likelihood estimate of the probability transition matrix. While the UAV is tracking a target i, it receives observations on the target's state. The filter simply propagates the state for all other targets $j \neq i$.

The decision mechanism for a revisit is as follows. Since the UAV maintains a multiple model filter on each of the targets, it maintains a combined covariance for each of the i targets, $P_{k+1|k+1}(i)$. The revisitation is determined when

$$\text{Trace}(P_{k+1|k+1}(i)) \geq \gamma \tag{3-23}$$

where γ is an appropriate threshold chosen for the problem, and depends on the UAV ability to track the different targets. For large γ, the UAV spends a lot of time visiting the different targets, and hence the revisitation rate will be lower than for a lower γ.

Tracking Results

We evaluated the performance of the Robust Multiple Model filter to the GPB1 implementation that uses unique realizations of the probability transition matrix in 50 Monte Carlo realizations. Figure 3-5 shows the benefit of using the RMM in terms of mean absolute error in a 2-state tracking example. The filter covariances (position) for the mismatched (green), robust (red), and true (blue) are shown on the bottom, and the mean absolute error over the 50 Monte Carlo simulations in shown in the top figure. Since the UAV revisits a target when the target's combined covariance $P_{k+1|k+1}$ exceeds the threshold γ, the mismatched covariance achieves this threshold approximately 6 units after the true covariance, and the mean absolute error almost

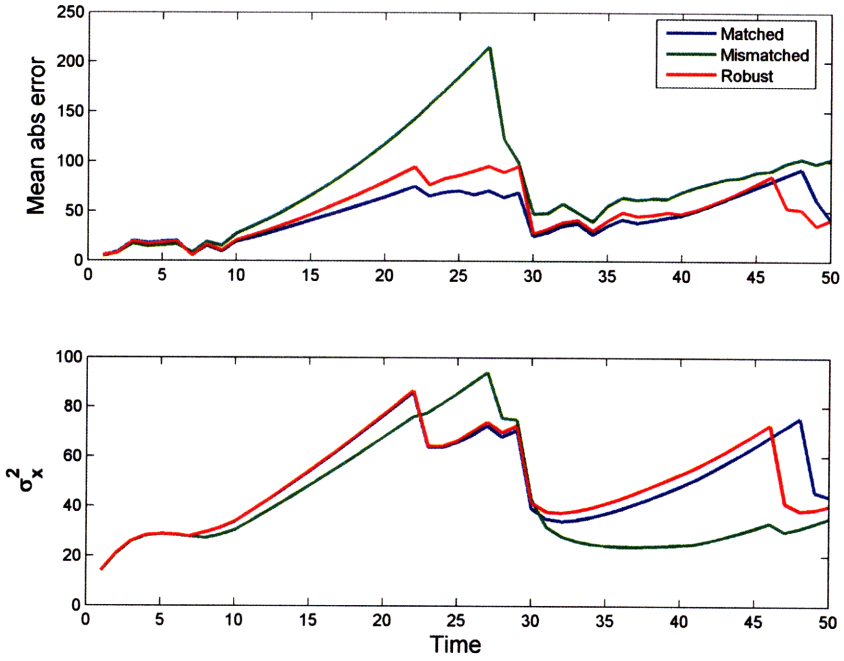


Fig. 3-5: Overall effect of the covariance underestimation problem: by underestimating the covariance, the UAV visits target 2 (shown) later, and accrues a much higher error.

doubles from 100 meters to 200 meters because the UAV revisits the target at a later time. Note that the mismatched estimator accrues a much higher error due to the mis-modeling of the probability transition model, and furthermore, by visiting the target at a later time, incurs additional estimation error. The robust filter ensures that the target is revisited sooner, and manages to keep the estimation error on the order of 100 meters.

Table 3.2: Revisitation Times

$\gamma = 500$	Mismatch (veh 1)	RMM (veh 1)	Mismatch (veh 2)	RMM (veh 2)
Mean time	6.2	4	6.4	4
Max time	9	4	9	4
Min time	6	4	6	4
$\gamma = 100$	Mismatch (veh 1)	Robust (veh 1)	Mismatch (veh 2)	Robust (veh 2)
Mean time	4	4	4	4
Max time	5	4	5	4
Min time	4	4	4	4

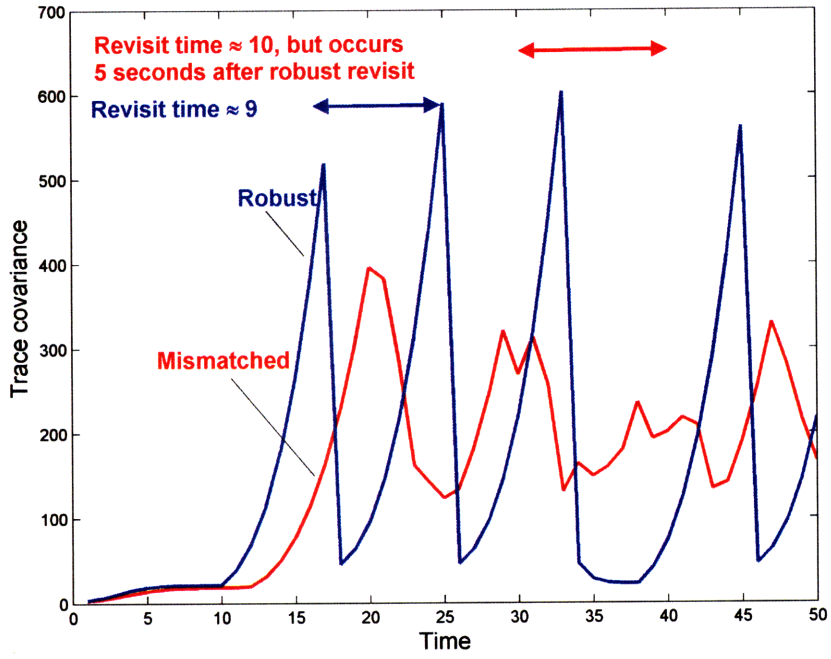


Fig. 3-6: Trace of the robust covariance (blue) and mismatched estimator (red) as a function of time for $\gamma = 500$. Note that the robust (in this case, the worst-case) estimator requires revisits every 9 time steps, while the mismatched on average requests every 10.

Variations in the revisit times of the UAV for the 4-state example were investigated for different values of threshold parameter γ; the mean, maximum, and minimum times are reported in Table 3.2. The RMM filter, by over approximating the covariance, sends the UAV back to revisit the targets sooner than the mismatched model, both on average (in terms of mean time), and in the worst case. This is a desirable feature as the target will travel a smaller distance, and the UAV will reacquire it with greater ease. More importantly, in the worst case, the revisit time for the mismatched case of Target 1 is of 9 time units, while the (conservative) RMM ensures that the target is revisited in at most 4 time units.

In general, the performance results of the robust multiple model estimator varies with the uncertainty model on the probability transition model $\tilde{\Pi}$, but also on the other parameters of the estimators, such as magnitude of the control inputs and

the process noise. An example is shown in Figure 3-6 where the control input was decreased to $u_k^i = \begin{bmatrix} 0 & 0.5 & 0 & 1.5 \\ 0 & 0 & 1.5 & 1.5 \end{bmatrix}$ and the process noise was decreased by a factor of 2. The simulation was for $\gamma = 500$. Here, the revisit time of approximately 10 time steps for the mismatched model is longer than the revisit time of the robust MM estimator, of approximately 9 time steps. Note that the benefits of the robust formulation, as the mismatched model revisit the target for the first time after 20 time steps, while the RMM revisits the target after 15 time steps, ensuring the overall estimation error remains low.

3.6.2 Tracking an Agile Target

We visit a variation of the tracking example used in Jilkov and Li [46] and Doucet and Ristic [27], where they consider the tracking of a single, but agile, target. The target has the same kinematics as the previous 2-state example, but now $\Delta T = 10$ and the control input takes on three distinct values:

$$u_k^i = [0,\ 20,\ -20] \quad [m/s^2] \tag{3-24}$$

The target starts at the initial state $x_0 = [8 \times 10^4 \quad 400]$ with initial covariance $P_0 = \begin{bmatrix} 100^2 & 0 \\ 0 & 100^2 \end{bmatrix}$. The measurement $z_k = x_k + v_k$ is corrupted by zero-mean Gaussian noise, $v_k \sim N(0, R)$, and unlike our earlier example where observations were taken only when the UAV was reallocated to a different target, the measurements in this example are taken at each time step k. For this numerical example, we compared a GPB1 filter operating with a nominal and mismatched transition matrix of

$$\hat{\Pi} = \begin{bmatrix} 0.50 & 0.29 & 0.2 \\ 0.33 & 0.43 & 0.20 \\ 0.17 & 0.29 & 0.6 \end{bmatrix}, \quad \hat{\Pi}_{mm} = \begin{bmatrix} 0.99 & 0.005 & 0.005 \\ 0.005 & 0.99 & 0.005 \\ 0.005 & 0.005 & 0.99 \end{bmatrix} \tag{3-25}$$

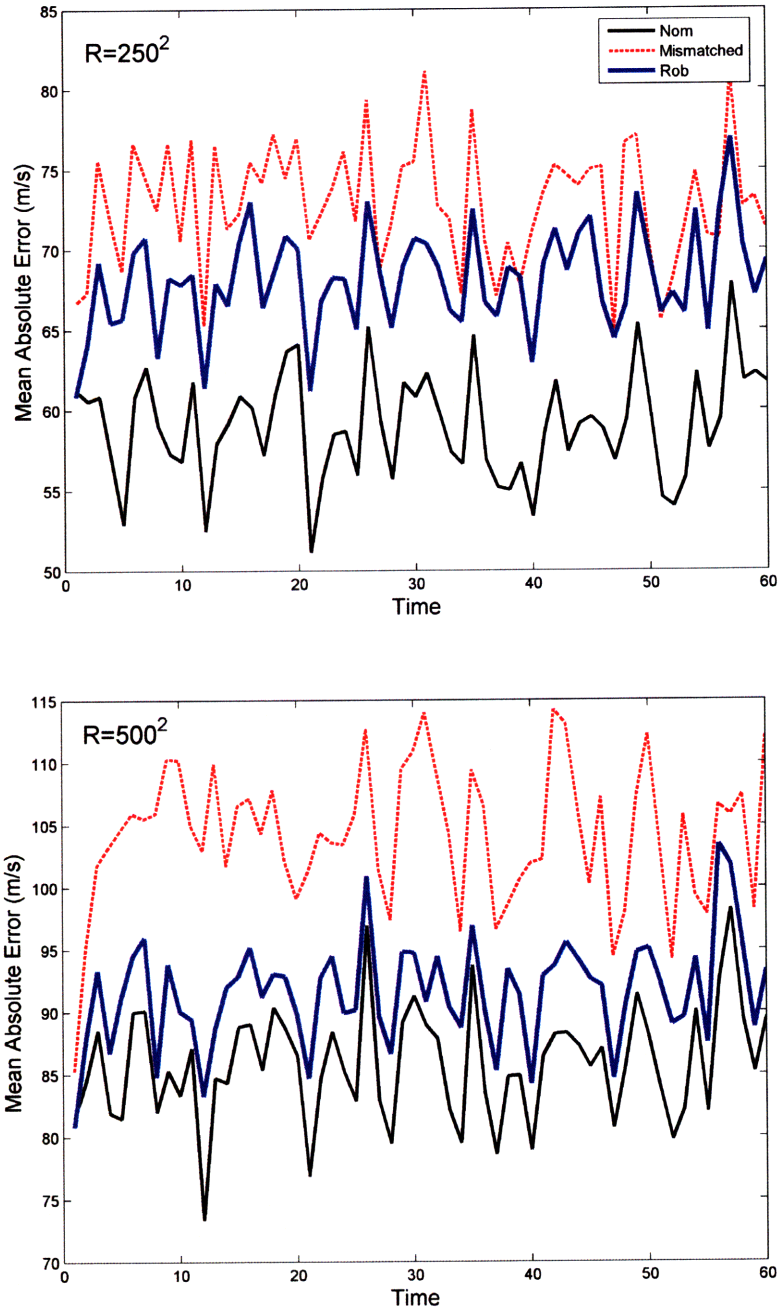


Fig. 3-7: Mean absolute error (MAE) in velocity as a function of time for two different noise covariances: (left) $R = 250^2$, and (right) $R = 500^2$. The nominal model (black) has the lowest MAE, while the mismatched model (red) has a higher MAE. The robust filter (blue) improves the performance of the mismatched model.

and compared them to the performance of the RMM algorithm. For this implementation, we used the sampling-based version of the RMM with $N_s = 100$ samples. We analyze the Mean Absolute Error (MAE) of the velocity for two different noise covariances, $R = 250^2$ and $R = 500^2$. The absolute error was calculated as the absolute value of the difference between the true velocity v_k and the estimated velocity $\hat{v}_k$,

$$\text{Absolute error} = \|\hat{v}_k - v_k\| \tag{3-26}$$

This quantity was then averaged over 200 Monte Carlo simulations to obtain the MAE. The results are shown in Figure 3-7. For a lower noise covariance ($R = 250^2$), the overall MAE is decreased from 74 meters/sec to approximately 66 meters/sec using the RMM, while for $R = 500^2$, the MAE of the mismatched filter was substantially decreased from 105 meters/sec to 90 meters/sec. Hence, the RMM improved the overall MAE by approximately 14%, highlighting the importance of accounting for the transition model uncertainty.

3.6.3 Computation Time

We conclude the results section with the run times of the proposed algorithm as a function of the total number of samples. There is an implicit tradeoff between the total number of samples used and the accuracy of the robust solution, and as the number of samples grows to infinity, the sampling approximation solution is exact. However, a *large* number of samples is generally sufficient, and Table 3.3 shows that the run times for a moderate number of samples is on the order of 0.6 seconds for 4 different targets each having 4 unique models. Note that the worst case formulation, which does not sample at all, has a mean run time of 0.02 seconds.

Table 3.3: Run Times of RMM as a function of number of samples N_s

N_s	0 (Worst case)	50	100	200	500	1000
Time (sec)	0.02	0.05	0.08	0.13	0.19	0.29

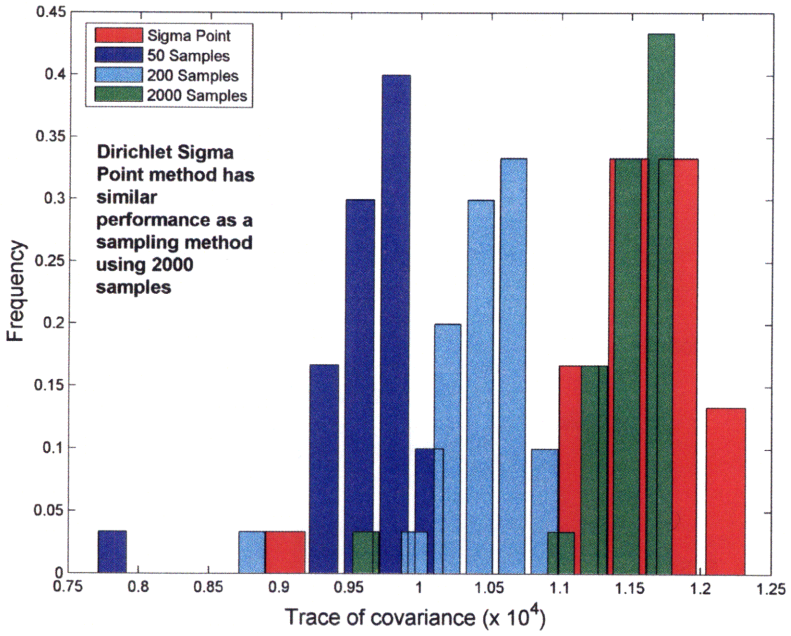


Fig. 3-8: Histogram of the updated covariance using the Dirichlet Sigma Points, and comparing to conventional Monte Carlo sampling. Dirichlet Sigma Point sampling, using only 7 transition matrix samples, recovers the histogram of the robust covariance that was generated using 2000 samples.

3.6.4 Computation time with Dirichlet Sigma Points

We finally compared the computational requirements of the Robust Multiple Model filter with a Dirichlet Sigma Point implementation of the Monte Carlo sampling. In a similar set of scenarios, we compared the covariance underestimation of using $N_s = \{50, 100, 200, 1000, 2000\}$ samples.

Figure 3-8 shows the histograms of the traces of the covariance over 100 Monte Carlo simulations for different choice of number of samples N_s. As N_s is increased, thereby making the Monte Carlo approximation more exact, the histograms converges around approximately 2000 samples. The histogram obtained with the Dirichlet Sigma Points however, converges using only 7 total transition matrix samples.

The effect on the computation time is more apparent in Figure 3-9. The robust covariance found with 2000 samples requires an average of 0.5 seconds per iteration of the RMM, while the robust covariance using the Dirichlet Sigma Points is found using

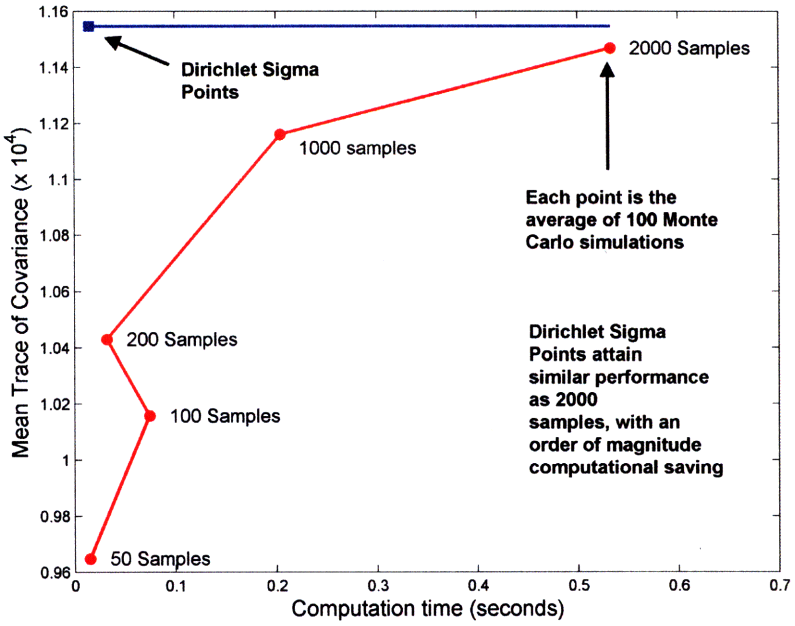


Fig. 3-9: Covariance estimation as a function of run time

only 0.02 seconds per iteration, with an order of magnitude computational savings.

3.7 Conclusions

This chapter has presented a new multiple model estimator that accounts for uncertainty in the probability transition model of the Markov Chain, and has extended previous work with uncertain transition models by identifying covariance mismatch problems in hybrid estimation. In particular, our concern was the covariance under-estimation problem, as it is undesirable for an estimator to be overly confident of its estimates.

To mitigate the worst-case impact of the uncertain transition probabilities, we have developed a new robust estimator with improved behavior over a nominal (mismatched) estimator, and specifically ensures that the estimator is not overly confident in the presence of an incorrect transition model. In the context of a UAV multi-target tracking problem, covariance under-estimation results in longer periods between re-

visits to the target, which ultimately results in larger estimation errors. Our new formulation is capable of keeping these errors small by ensuring more frequent revisits. In the context of a tracking problem for an agile target, the new filter is able to keep the overall tracking errors small in comparison to a fairly mismatched filter.

Appendix 3-A: Multiple Model Estimator Remarks

Remark 1 (Choice of estimator): The Multiple Model Adaptive Estimator (MMAE) [62] and the Generalized Pseudo Bayesian [4] are two common estimators used in multiple model settings. The primary distinction between MMAE and GPB (or other suboptimal filters) is that the MMAE assumes that the discrete state of the hybrid system is invariant throughout the estimation process. That is, MMAE does not explicitly include information about the transition probabilities.

Remark 2 (Impact of Measurement Noise): The sensor quality directly impacts the estimator's ability to uniquely identify the current model μ_i, and update the state estimate and covariance. In this section, we show the limiting effect of the noise on the state estimate, covariance, and probability updates.

The sensor likelihood can be expressed in terms of the measurement noise covariance R^i as

$$\begin{aligned} \Lambda^i(z_k) &\propto |S^i|^{-1} \exp\{-1/2(z-\hat{z}^i)^T\ (S^i)^{-1}\ (z-\hat{z}^i)\} \\ S^i &= H^i P_{k+1|k} H^{i,T} + R^i \end{aligned} \tag{3A-1}$$

In the limiting case $R \to \infty$, the likelihood $\Lambda^i(z_k) \to \Lambda \to 0$, and the Kalman gain $W_k^i \to \mathbf{0}$. As a result, the measurement update simply becomes the prediction step

$$\begin{aligned} \hat{x}^i_{k+1|k+1} &= \hat{x}^i_{k+1|k} = \Phi^i \hat{x}_{k|k} + G^i u^i_k \\ P^i_{k+1|k+1} &= P^i_{k+1|k} = \Phi^i P_{k|k} (\Phi^i)^T + Q^i \end{aligned} \tag{3A-2}$$

As the likelihood $\Lambda \to 0$, the mode probability update step then becomes

$$\mu_j = \frac{1}{c} \Lambda \sum_i \Pi_{ij} \mu_i, \qquad (c = \sum_j \Lambda \sum_i \Pi_{ij} \mu_i) \tag{3A-3}$$

and by taking the limit as $\Lambda \to 0$ and applying l'Hopital's rule,

$$\lim_{\Lambda \to 0} \frac{\Lambda \sum_i \Pi_{ij}\mu_i}{\sum_j \Lambda \sum_i \Pi_{ij}\mu_i} = \lim_{\Lambda \to 0} \frac{\Lambda \sum_i \Pi_{ij}\mu_i}{\Lambda \underbrace{\sum_j \sum_i \Pi_{ij}\mu_i}_{=1}}$$

$$= \sum_i \Pi_{ij}\mu_i \tag{3A-4}$$

Therefore, the probability update with a poor sensor is simply the probability propagation using the transition model Π.

Appendix 3B: Optimization Derivations

The derivations in this Appendix develop the form of the optimization used to find the posterior probability model μ that maximizes the trace of the covariance at the combination step. The first section shows how to form this maximization while including the unit sum constraint ($\sum_j \mu_j = 1$) as part of the constraints. In the second section, the derivation is repeated for the case of the equality constraint directly included in the objective function.

Including equality constraint Given the state and the covariance,

$$\hat{x}_{k+1|k+1} = \sum_i \tilde{\mu}_i \hat{x}^i_{k+1|k+1}$$

$$P_{k+1|k+1} = \sum_i \tilde{\mu}_i \left[P^i_{k+1|k+1} + (\hat{x}^i_{k+1|k+1} - \hat{x}_{k+1|k+1})(\hat{x}^i_{k+1|k+1} - \hat{x}_{k+1|k+1})^T\right] \tag{3B-1}$$

we can rewrite the covariance as

$$P_{k+1|k+1} = \sum_i \tilde{\mu}_i \left[P^i_{k+1|k+1} + (\hat{x}^i_{k+1|k+1} - \hat{x}_{k+1|k+1})(\hat{x}^i_{k+1|k+1} - \hat{x}_{k+1|k+1})^T\right] \tag{3B-2}$$

$$= \sum_i \tilde{\mu}_j \{P^i + \hat{x}^i_{k+1|k+1} \, (\hat{x}^i_{k+1|k+1})^T\} - (\sum_j \tilde{\mu}_j \hat{x}^j_{k+1|k+1})(\sum_j \tilde{\mu}_j \hat{x}^j_{k+1|k+1})^T$$

Note that the last term $(\sum_j \tilde{\mu}_j \hat{x}^j_{k+1|k+1})(\sum_j \tilde{\mu}_j \hat{x}^j_{k+1|k+1})^T$ is a rank one matrix of the form $\mu^T Q \mu$, where $\text{rank}(Q) = 1$. Now, the trace of this matrix is

$$\begin{aligned}\text{Tr}(P_{k+1|k+1}) &= \text{Tr}\sum_i \tilde{\mu}_j\{P^i + \hat{x}^i_{k+1|k+1}\ (\hat{x}^i_{k+1|k+1})^T\} - \text{Tr}(\sum_j \tilde{\mu}_j \hat{x}^j_{k+1|k+1})(\sum_j \tilde{\mu}_j \hat{x}^j_{k+1|k+1})^T \\ &= \sum_j \text{Tr}\{P^i + \hat{x}^i_{k+1|k+1}\ (\hat{x}^i_{k+1|k+1})^T\}\tilde{\mu}_j - \sum_j \sum_m \tilde{\mu}_j \tilde{\mu}_m \text{Tr}\{\hat{x}^j_{k+1|k+1}(\hat{x}^m_{k+1|k+1})^T\}\end{aligned}$$

where the linearity of the trace operator was used. Thus, we can now now define

$$\begin{aligned}A_{j,m} &\doteq \text{Tr}\{\hat{x}^j_{k+1|k+1}(\hat{x}^m_{k+1|k+1})^T\}, \quad \forall j, m \\ B_j &\doteq \text{Tr}\{P^j + \hat{x}^j_{k+1|k+1}\ (\hat{x}^j_{k+1|k+1})^T\}, \quad \forall i\end{aligned}$$

and with $\tilde{\mu} \doteq [\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_{N_M}]$, then the optimization of maximizing the trace of the combined covariance becomes

$$\max_{\tilde{\mu}} \ \{-\tilde{\mu}^T A \tilde{\mu} + B\tilde{\mu} \mid \mathbf{1}^T \tilde{\mu} = 1, \tilde{\mu} \in \mathcal{M}_k\} \tag{3B-3}$$

Note that A_{k+1} can be formed as the product of $G\ G^T$, where

$$G = \left[\hat{x}^1_{k+1|k+1} \mid \hat{x}^2_{k+1|k+1} \mid \ \cdots \ \mid \hat{x}^N_{k+1|k+1}\right]^T \tag{3B-4}$$

is non-singular. Since A_{k+1} can be formed as the product of a (non-singular) matrix and its transpose, this forms the sufficient and necessary condition for A_{k+1} to be positive definite. [24] Hence, $A_{k+1} \succ 0$ and the optimization is convex.

Removing the equality constraint The equality constraint of $\sum_j \mu_j = 1$ can be removed from the set of constraints of the robust optimization, by rewriting the covariance

$$P_{k+1|k+1} = \sum_i \tilde{\mu}_i\{P^i + \hat{x}^i_{k+1|k+1}\ (\hat{x}^i_{k+1|k+1})^T\} - (\sum_j \tilde{\mu}_j \hat{x}^j_{k+1|k+1})(\sum_j \tilde{\mu}_j \hat{x}^j_{k+1|k+1})^T$$

and through a series of algebraic manipulations obtain

$$
\begin{aligned}
P_{k+1|k+1} &= M_1 + M_2 + M_3 \\
M_1 &= \sum_{i=1}^{N_m-1} \sum_{j=1}^{N_m-1} \tilde{\mu}_j \tilde{\mu}_j \{(\hat{x}^i_{k+1|k+1} - \hat{x}^{N_m}_{k+1|k+1})(\hat{x}^j_{k+1|k+1} - (\hat{x}^{N_m}_{k+1|k+1}))^T\} \\
M_2 &= \sum_{i=1}^{N_m-1} \tilde{\mu}_i \{P^i - P^{N_m} + (\hat{x}^i_{k+1|k+1} - \hat{x}^{N_m}_{k+1|k+1})(\hat{x}^i_{k+1|k+1} - (\hat{x}^{N_m}_{k+1|k+1}))^T\} \\
M_3 &= P^{N_m} + (\hat{x}^{N_m}_{k+1|k+1})(\hat{x}^{N_m}_{k+1|k+1})^T
\end{aligned}
$$

Hence new matrices can be evaluated ($\forall i, j = 1, \ldots, N_m - 1$)

$$
\begin{aligned}
\hat{A}_{i,j} &\doteq \mathrm{Tr}\{(\hat{x}^i_{k+1|k+1} - \hat{x}^{N_m}_{k+1|k+1})(\hat{x}^j_{k+1|k+1} - (\hat{x}^{N_m}_{k+1|k+1}))^T\} \\
\hat{B}_i &\doteq \mathrm{Tr}\{P^i + (\hat{x}^i_{k+1|k+1} - \hat{x}^{N_m}_{k+1|k+1})(\hat{x}^i_{k+1|k+1} - (\hat{x}^{N_m}_{k+1|k+1}))^T - P^{N_m}\} \\
\hat{C}_i &\doteq \mathrm{Tr}\{P^{N_m} + (\hat{x}^{N_m}_{k+1|k+1})(\hat{x}^{N_m}_{k+1|k+1})^T\}
\end{aligned}
$$

and the optimization can be posed as

$$
\min_{\tilde{\mu}} \{\tilde{\mu}^T \hat{A} \tilde{\mu} - \hat{B}\tilde{\mu} - \hat{C} \mid \tilde{\mu} \in \mathcal{M}_k\} \tag{3B-5}
$$

where now $\tilde{\mu} = [\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_{N_m-1}]$.

Appendix 3C: Selecting Uncertainty Set $\mathcal{M}_k$

The uncertainty description for $\tilde{\mu}$, $\mathcal{M}_k$, can be described in numerous ways; two common approaches are shown below. Nilim and El Ghaoui [69] developed the ellipsoidal model and interval uncertainty to describe the modeling of the rows of the transition model, but we use them to describe the posterior probabilities.

Interval Bounds Uncertainty Model One possible approach is to assume that each $\tilde{\mu}_j$ is bounded above and below, $\tilde{\mu}_j^- \leq \tilde{\mu}_j \leq \tilde{\mu}_j^+$ by some values $\tilde{\mu}_j^+$ and $\tilde{\mu}_j^-$. This

uncertainty set is therefore defined as

$$\mathcal{M}_k \doteq \{\tilde{\mu} : \tilde{\mu}_j^- \leq \tilde{\mu}_j \leq \tilde{\mu}_j^+\} \tag{3C-1}$$

The optimization of Eq. 3-14 can be rewritten as

$$\min_{\tilde{\mu}} \{\tilde{\mu}^T A\tilde{\mu} - B\tilde{\mu} \mid \mathbf{1}^T\tilde{\mu} = 1, \tilde{\mu}_j^- \leq \tilde{\mu}_j \leq \tilde{\mu}_j^+\} \tag{3C-2}$$

This can be converted into the equivalent quadratic program

$$\begin{aligned} &\min_{\tilde{\mu}} \{\tilde{\mu}^T A\tilde{\mu} - B\tilde{\mu} \mid \mathbf{1}^T\tilde{\mu} = 1, \tilde{A}\tilde{\mu} \leq \tilde{B}\} \\ &\tilde{A} = [I^{Nm\times Nm}, -I^{Nm\times Nm}]^T, \quad \tilde{B} = [\tilde{\mu}^+, -\tilde{\mu}^-]^T \end{aligned} \tag{3C-3}$$

and solved directly as a QP using interior point methods.

An Alternative Solution The single equality constraint $\mathbf{1}^T\tilde{\mu} = 1$ can be removed by explicitly accounting for the unit sum constraint directly in the objective function. The optimization becomes

$$\min_{\tilde{\mu}} \{\tilde{\mu}^T \hat{A}\tilde{\mu} - \hat{B}\tilde{\mu} \mid \hat{A}'\tilde{\mu} \leq \hat{B}'\} \tag{3C-4}$$

where now $\tilde{\mu} = [\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_{N_m-1}]$ and

$$\begin{aligned} \hat{A}_{ij} &= \mathrm{Tr}[(\hat{x}^i - \hat{x}^N)(\hat{x}^j - \hat{x}^N)^T], \quad \forall i,j = 1,\ldots,N_m - 1 \\ \hat{B}_j &= \mathrm{Tr}[(\hat{x}^j - \hat{x}^N)(\hat{x}^j - \hat{x}^N)^T + P^j - P^N], \quad \forall j = 1,\ldots,N_m - 1 \\ \hat{A}' &= [I^{N_m-1\times N_m-1}, -I^{N_m-1\times N_m-1}]^T, \\ \hat{B}' &= [\tilde{\mu}^+, -\tilde{\mu}^-]^T \end{aligned}$$

This is also solvable solvers that allow quadratic programs (i.e., AMPL/CPLX). If a QP solver is not available, other techniques are available to solve this problem using decomposition methods for quadratic programs (using a modified simplex algorithm).

Ellipsoidal Uncertainty Model The ellipsoidal uncertainty model assumes the existence of a maximum likelihood estimate for the uncertain likelihood $\hat{\mu}$. This model is equivalent to a second order approximation to the log-likelihood function and the uncertainty set can be constructed as

$$\sum_j \tilde{\mu}_j \log(\tilde{\mu}_j/\hat{\mu}_j) \leq \sum_j \tilde{\mu}_j \left(\frac{\tilde{\mu}_j}{\hat{\mu}_j} - 1\right) \leq \sum_j (\tilde{\mu}_j - \hat{\mu}_j)^2/\hat{\mu}_j \tag{3C-5}$$

using the approximation $\log p \leq (1 + p)$ (Iyengar [43]). The optimization for the uncertain $\tilde{\mu}$ is given by the following optimization

$$\min_{\tilde{\mu}} \{\tilde{\mu}^T A\tilde{\mu} - B\tilde{\mu} \mid 1^T\tilde{\mu} = 1, \sum_j (\tilde{\mu}_j - \hat{\mu}_j)^2/\hat{\mu} \leq \kappa^2\}$$

where $\kappa = 2(\beta_{\max} - \beta)$ and $\beta_{\max} = \sum_j \hat{\mu}_j \log \hat{\mu}_j$. β is a tuning parameter chosen by the filter designer to select how much uncertainty is desired in the filter design. The resulting optimization is a quadratically constrained quadratic program (QCQP), which can be solved with CPLEX 10.

Note that the particular constraint $\sum_j (\tilde{\mu}_j - \hat{\mu}_j)^2/\hat{\mu}_j$ can be simplified to

$$\begin{aligned}\sum_j (\tilde{\mu}_j - \hat{\mu}_j)^2/\hat{\mu}_j &= \sum_j 1/\hat{\mu}_j(\tilde{\mu}_j^2 - 2\tilde{\mu}_j\hat{\mu}_j + \hat{\mu}_j^2) \\ &= \sum_j 1/\hat{\mu}_j\tilde{\mu}_j^2 - 1\end{aligned}$$

since $\sum_j \tilde{\mu}_j = 1$ and $\sum_j \hat{\mu}_j = 1$. Hence the quadratic constraint is simply of the form

$$\tilde{\mu}^T M_k \tilde{\mu} \leq N_k \tag{3C-6}$$

with $M_k = \text{diag}(1/\hat{\mu}_1, 1/\hat{\mu}_2, \ldots, 1/\hat{\mu}_{N_m})$ and $N = \kappa^2 + 1$.

Chapter 4

Markov Chain Adaptation

This chapter addresses the important issue of identification of the uncertain transition probabilities. Whereas in the previous chapters we have been concerned with *mitigating* the impact of the uncertainty on the performance of the control and estimation algorithms, this chapter addresses the question of model updating: given some sequence of observed state transitions of the system, how do we update the estimate of the transition probabilities, and their uncertainty description? In particular, how do we update the Dirichlet Sigma Points if the transition probabilities are time-varying?

Our primary contribution in this chapter is an algorithm that can adapt to online observations more efficiently by reducing the overall adaptation time of the Markov Chain. We first derive recursive forms for the the first two moments of the Dirichlet, and obtain recursive expressions for the Dirichlet Sigma Points. It turns out that this mean-variance recursion (used as a synonym for the recursive Dirichlet Sigma Points) can be slow in responding to changes in the transition probability. This mean-variance estimation is then improved by adding what amounts to an effective process noise term to the covariance term of the Dirichlet Sigma Points. We present the details of this simple, but effective algorithm, as well as some theoretical justification for the process noise addition as an effective measurement fading technique of the Dirichlet counts.

4.1 Introduction

4.1.1 Previous work

In the earlier chapters, we have seen that many decision processes, such as Markov Decision Processes (MDPs) and Jump Markov Linear systems, are modeled as a probabilistic process driven by a Markov Chain. The true parameters of the Markov Chain are frequently unavailable to the modeler, and many researchers have recently addressed the issue of robust performance in these decision systems [12, 59, 69, 86]. However, a large body of research has also been devoted to the identification of the Markov Chain using available observations. With few exceptions (such as the signal processing community [50, 76]), most of this research has addressed the case of a unique, stationary model.

When the transition matrix Π of a Markov Chain is stationary, classical maximum likelihood (ML) schemes exist ([46, 76]) that can recursively obtain the best estimate, $\hat{\Pi}$, and covariance of the transition matrix. Typical Bayesian methods assume a prior Dirichlet distribution on each row of the transition matrix, and exploit the conjugacy property of the Dirichlet distribution with the multinomial distribution to recursively compute $\hat{\Pi}$. This technique amounts to evaluating the empirical frequency of the transitions to obtain a ML or Maximum A Posteriori (MAP) estimate of the transition matrix. In the limit of an infinite observation sequence, this method converges to the true transition matrix, Π. Jilkov and Li [46] discuss the identification of the transition matrices in the context of Markov Jump systems, providing multiple algorithms that can identify Π using noisy measurements that are indirect observations of the transitions. Jaulmes et al. [44, 45] study this problem in an active estimation context using Partially Observable Markov Decision Processes (POMDPs). Marbach [60] considers this problem, when the transition probabilities depend on a parameter vector. Borkar and Varaiya [16] treat the adaptation problem in terms of a single parameter as well; namely, the true transition probability model is assumed to be a function a single parameter a belonging to a finite set $\mathcal{A}$. The adaptation algorithm recursively computes the maximum likelihood estimate of the parameter $\hat{\alpha}$ and Borkar and Varaiya's

adaptive algorithm is shown to converge (though their estimator may not converge to the true parameter). If the true parameter is not in the set $\mathcal{A}$, however, some examples were shown where their adaptive controller could not converge at all.

Konda and Tsitsiklis [49] consider the problem of slowly-varying Markov Chains in the context of reinforcement learning. Sato [80] considers this problem and shows asymptotic convergence of the probability estimates also in the context of dual control. Kumar [53] also considered the adaptation problem. Ford and Moore [30] consider the problem of estimating the parameters of a non-stationary Hidden Markov Model.

If the Markov Chain, Π_t, is changing over time, classical estimators will generally fail to respond quickly to changes in the model. The intuition behind this is that since these estimators keeps track of all the transitions that have occurred, a large number of new transitions will be required for the change detection, and convergence to the new model. Hence, new estimators are required to compensate for the inherent delay that will occur in classical techniques. Note that if the dynamics of the transition matrix were available to the estimator designer, they could be embedded directly in the estimator. For example, if the transition matrix were known to switch between two systems according to a probabilistic switching schedule, or if the switching time were a random variable with known statistics, these pieces of information could enhance the performance of any estimator. However, in a more general setting, it is unlikely that this information would be available to the estimator designer.

4.1.2 Outline

This chapter proposes a new technique to speed up the estimator response that does not require information about the dynamics of the uncertain transition model. First, recursions for the mean and variance of the Dirichlet distribution are derived; this is a mean-variance interpretation of classical MAP estimation techniques. These are recursions for the Dirichlet Sigma Points, as the Sigma Points introduced in the earlier chapter are calculated using the first two moments of the Dirichlet distribution.

Importantly, however, we use the similarity of these recursions to filter-based parameter estimation techniques to notice that the mean-variance estimator does not

incorporate any knowledge of the parameter (or transition matrix) dynamics, and therefore results in a stationary prediction step. To compensate for this, the responsiveness of the estimator is improved by adding an effective artificial pseudonoise to the variance that is implemented by scaling the variance. Scaling the variance leads to a very natural interpretation for updating the Dirichlet parameters, which, as we show, amounts to nothing more than progressively fading the impact of older transitions. This result provides an intuition for measurement fading applied to Hidden Markov Models [50]. This insight, and the resulting benefits of faster estimation when applied to decision systems, are the core results of this chapter.

4.2 Markov Chain and the Dirichlet Distribution

As before, when the transition matrix Π is uncertain, we take a Bayesian viewpoint and assume a prior Dirichlet distribution on each row of the transition matrix, and recursively update this distribution with observations.[1]

The mean and the variance of the Dirichlet distribution can then be calculated directly as

$$\bar{p}_i = \alpha_i/\alpha_0 \tag{4-1}$$

$$\Sigma_{ii} = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)} \tag{4-2}$$

These are the mean and the variance of each column of the transition model, and need to be evaluated for all rows (recalling $p_i = \pi(m, i)$). We have shown that the Dirichlet Sigma Points only rely on these first two moments.

4.2.1 Derivation of Mean-Variance Estimator

It is well known that the Dirichlet distribution is conjugate to the multinomial distribution; therefore, performing a Bayesian measurement update step on the Dirichlet

[1]Since each row of the transition matrix satisfies the properties of a probability vector, the following description of the Dirichlet distribution is interpreted to apply to *each row* of the transition matrix.

amounts to a simple addition of currently observed transitions to the previously observed counts $\alpha(k)$. Here, we define $\mathbf{p}_k = [p_1, p_2, \ldots, p_N]^T$ as the parameter at time k. The posterior distribution $f_D(\mathbf{p}_{k+1}|\alpha(k+1))$ is given in terms of the prior $f_D(\mathbf{p}_k|\alpha(k))$ as

$$\begin{aligned} f_D(\mathbf{p}_{k+1}|\alpha(k+1)) &\propto f_D(\mathbf{p}_k|\alpha(k)) f_M(\beta(k)(|\mathbf{p}_k) \\ &= \prod_{i=1}^{N} p_i^{\alpha_i - 1} p_i^{\beta_i} = \prod_{i=1}^{N} p_i^{\alpha_i + \beta_i - 1} \end{aligned}$$

where $f_M(\beta(k)|\mathbf{p}_k)$ is a multinomial distribution with hyperparameters $\beta(k) = [\beta_1, \ldots, \beta_N]$. Each β_i is the total number of transitions observed from state i to a new state i': mathematically $\beta_{i'} = \sum_i \delta_{i,i'}$ and

$$\delta_{i,i'} = \begin{cases} 1 & \text{if } i = i' \\ 0 & \text{else} \end{cases}$$

indicates how many times transitions were observed from state i to state i'. For the next derivations, we assume that only a single transition can occur per time step, $\beta_i = \delta_{i,i'}$.

Upon receipt of the observations $\beta(k)$, the parameters $\alpha(k)$ are thus updated in the following manner

$$\alpha_i(k+1) = \begin{cases} \alpha_i(k) + 1 & \text{Transition } i \text{ to } i' \\ \alpha_i(k) & \text{Else} \end{cases}$$

The mean and the variance can then be calculated by using Eqs. 4-1 and 4-2.

Instead of calculating the mean and variance from the transitions at each time step, we can directly find recursions for the mean $\bar{p}_i(k)$ and variance $\Sigma_{ii}(k)$ of the Dirichlet distribution by deriving the Mean-Variance Estimator with the following proposition.

Proposition 10 *The posterior mean $\bar{p}_i(k+1)$ and variance $\Sigma_{ii}(k+1)$ of the Dirichlet distribution can be found in terms of the prior mean $\bar{p}_i(k)$ and variance $\Sigma_{ii}(k)$ by using*

the following recursion for the Mean-Variance Estimator:

$$\bar{p}_i(k+1) = \bar{p}_i(k) + \Sigma_{ii}(k)\frac{\delta_{i,i'} - \bar{p}_i(k)}{\bar{p}_i(k)(1-\bar{p}_i(k))}$$
$$\Sigma_{ii}^{-1}(k+1) = \gamma_{k+1}\,\Sigma_{ii}^{-1}(k) + \frac{1}{\bar{p}_i(k+1)(1-\bar{p}_i(k+1))}$$

where $\gamma_{k+1} = \frac{\bar{p}_i(k)(1-\bar{p}_i(k))}{\bar{p}_i(k+1)(1-\bar{p}_i(k+1))}$.

Proof: Since the prior mean $\bar{p}_i(k) = \alpha_i/\alpha_0$ and the posterior mean is given by $\bar{p}_i(k+1) = (\alpha_i + \delta_{i,i'})/(\alpha_0 + 1)$, the difference between the two means is given by

$$\begin{aligned}\bar{p}_i(k+1) - \bar{p}_i(k) &= \frac{\alpha_i + \delta_{i,i'}}{\alpha_0 + 1} - \frac{\alpha_i}{\alpha_0} \\ &= \frac{\delta_{i,i'} - \bar{p}_i(k)}{\alpha_0 + 1}\end{aligned} \tag{4-3}$$

The variance $\Sigma_{ii}(k)$ is given by

$$\Sigma_{ii}(k) = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)} = \frac{\bar{p}_i(k)(1 - \bar{p}_i(k)}{\alpha_0 + 1}. \tag{4-4}$$

Eq. 4-4 can be inverted to solve for $\alpha_0 + 1$ and substitute in Eq. 4-3 to obtain the desired result

$$\bar{p}_i(k+1) = \bar{p}_i(k) + \Sigma_{ii}(k)\frac{\delta_{i,i'} - \bar{p}_i(k)}{\bar{p}_i(k)(1 - \bar{p}_i(k))} \tag{4-5}$$

An equivalent argument follows for the variance, $\Sigma_{ii}(k)$. Since

$$\begin{aligned}\Sigma_{ii}(k+1) &= \frac{(\alpha_i + \delta_{i,i'})(\alpha_0 + 1 - (\alpha_i + \delta_{i,i'}))}{(\alpha_0 + 1)^2(\alpha_0 + 2)} \\ &= \frac{\bar{p}_i(k+1)(1 - \bar{p}_i(k+1))}{\alpha_0 + 2}\end{aligned}$$

Table 4.1: Mean variance recursion shown in prediction and update step

	Mean-variance
Prediction	$\bar{p}_i(k+1\|k) = \bar{p}_i(k\|k)$ $\Sigma_{ii}(k+1\|k) = \Sigma_{ii}(k\|k)$
Measurement update	$\bar{p}_i(k+1\|k+1) = \bar{p}_i(k+1\|k) + \Sigma_{ii}(k+1\|k)\frac{\delta_{i,i'} - \bar{p}_i(k+1\|k)}{\bar{p}_i(k+1\|k)(1-\bar{p}_i(k+1\|k))}$ $\Sigma_{ii}^{-1}(k+1\|k+1) = \gamma_{k+1}\Sigma_{ii}^{-1}(k+1\|k) + \frac{1}{\bar{p}_i(k+1\|k)(1-\bar{p}_i(k+1\|k))}$

then we can see that the inverse variance $\Sigma_{ii}^{-1}(k+1)$ satisfies the following recursion

$$\Sigma_{ii}^{-1}(k+1) = \frac{\bar{p}_i(k)(1-\bar{p}_i(k))}{\bar{p}_i(k+1)(1-\bar{p}_i(k+1))}\Sigma_{ii}^{-1}(k) + \frac{1}{\bar{p}_i(k+1)(1-\bar{p}_i(k+1))}$$

but given the definition of γ_{k+1}, then

$$\Sigma_{ii}^{-1}(k+1) = \gamma_{k+1}\Sigma_{ii}^{-1}(k) + \frac{1}{\bar{p}_i(k+1)(1-\bar{p}_i(k+1))}$$

■

Remark 1: The recursion for the mean is actually the maximum a posteriori (MAP) estimator of the Dirichlet distribution, expressed in terms of prior mean and variance. If the updated counts are $\alpha'(k+1)$, then the posterior distribution is given by

$$f_D(\mathbf{p}_{k+1}|\alpha'(\mathbf{k}+\mathbf{1})) = K\prod_{i=1}^{N} p_i^{\alpha_i'}, \quad \sum_i p_i = 1 \tag{4-6}$$

and the MAP estimate is $\hat{p}_i = \arg\max f_D(\mathbf{p}|\alpha'(\mathbf{k}+\mathbf{1}))$.

Remark 2: This mean-variance estimator explicitly guarantees that the updated Dirichlet Sigma Points sum to unity, $\sum_i \bar{p}_i(k|k) = 1, \quad \forall k$, since they are calculated directly from the MAP estimate. Other mean-variance approaches [46] only enforce the unit sum constraint at the end of each estimator cycle, through some form of ad-hoc renormalization, which is not exact. However, in the mean-variance form for the Dirichlet, no approximations are needed to ensure that the estimates remain within the unit simplex.

Remark 3: The convergence of the mean-variance estimator is guaranteed since the MAP estimator is guaranteed to converge [44]. After a large number of observations, the MAP estimate of the probability $\bar{p}_i = \alpha_i/\alpha_0$ will be equal to the true probability p_i, and the variance asymptotically approaches 0.

This is immediately clear from the mean-variance formulation as well. From proposition 1, the estimate $\bar{p}_i(k)$ will converge if $\lim_{k\to\infty} \bar{p}_i(k+1) - \bar{p}_i(k) = 0$, which implies that for any arbitrary measurement $\delta_{i,i'}$, that this will be true if the variance asymptotically approaches 0, $\lim_{k\to\infty} \Sigma_{ii}(k) = 0$.

The steady-state covariance can be found explicitly in the mean-variance estimator by rearranging the expression in Proposition 1, and taking the limit.

$$\lim_{k\to\infty} \Sigma_{ii} = \lim_{k\to\infty} (1 - \gamma_{k+1})\, \bar{p}_i(k+1)(1 - \bar{p}_i(k+1)) = 0$$

Note that we have used the fact that, since the estimate converges, then by definition of γ_k, $\lim_{k\to\infty} \gamma_{k+1} = 1$.

Remark 4: The mean-variance estimator can also be expressed more explicitly in a prediction step and a measurement update step, much like in conventional filtering. The prior distribution is given by $f_D(\mathbf{p}_{k|k}|\alpha(k))$ where the prior estimate is now written as $\bar{p}_i(k|k)$. The propagated distribution is $f_D(\mathbf{p}_{k+1|k}|\alpha(k))$ and the propagated estimate is denoted as $\bar{p}_i(k+1|k)$. The posterior distribution is $f_D(\mathbf{p}_{k+1|k+1}|\alpha(k+1))$, where $\alpha(k+1)$ are the updated counts, and the updated estimate is written as $\bar{p}_i(k+1|k+1)$. These steps are shown in Table 4.1. In the (trivial) prediction step, the mean and the variance do not change, while the measurement update step is the proposition we just derived.

4.3 Discounted Mean Variance Estimator Derivation

The general limitation of applying this estimation technique to a non-stationary problem is that the variance of the estimator decreases to **0** fairly rapidly after $N_m \ll \infty$

measurements, which in turn implies that new observations $\delta_{i,i'}$ will not be heavily weighted in the measurement update. This can be seen in the measurement update step of Table 4.1: as the variance Σ_{ii} approaches zero, then new measurements have very little weighting.

This covariance can be thought of as the measurement *gain* of classical Kalman filtering recursions. A way to modify this gain is by embedding transition matrix dynamics. If transition matrix dynamics were available, these could be embedded in the estimator by using the Chapman-Kolmogorov equation $\int P(\pi_{k+1}|\pi_k)P(\pi_k|\alpha(k)d\pi_k$ in the prediction step. However, in general, the dynamics of the parameter may not be well known or easily modeled.

In parameter estimation, well known techniques are used to modify this prediction step for a time-varying unknown parameters, such as through the addition of artificial pseudonoise [4], or scaling the variance by a (possibly time-varying) factor greater than unity [66]. Both pseudonoise addition or covariance scaling rely on the fundamental idea of increasing the covariance of the estimate in the prediction step.

In Miller [66], and in the context of Kalman filtering, Miller artificially scales the predicted covariance matrix $\Sigma_{k+1|k}$ by a time-varying scale factor ω_k ($\omega_k > 1$) and shows that the Kalman filter recursions remain virtually unchanged, except that that predicted variance $\Sigma_{k+1|k}$ is modified to $\Sigma'_{k+1|k} = \omega_k \Sigma_{k+1|k}$. Since $\omega_k > 1$, this has the effect of increasing the covariance, thereby reducing the estimator's confidence and changing the Kalman gain to be more responsive to new measurements.

4.3.1 Adaptation for Dirichlet Sigma Points

This similar intuition is used to derive a modified mean-variance estimator for the case of the Dirichlet distribution; define $\lambda_k = 1/\omega_k$ (where now $\lambda_k < 1$), modify the prediction steps in a similar way to Miller, and obtain the direct analog for the modified mean-variance estimator. The new update step for the variance is given by

$$\Sigma_{ii}^{-1}(k+1|k) = \lambda_k \Sigma_{ii}^{-1}(k|k) \tag{4-7}$$

The variance is now scaled by a factor $1/\lambda_k > 1$ at each iteration. This discounted mean-variance estimator is shown in Table 4.2. We will remark further on the choice of λ_k in the numerical simulations, but there is an implicit tradeoff between speeding up estimator response, and overall estimation error.

Table 4.2: Kalman filter recursion and using scaling

	Conventional KF	Scaled form [66]
Prediction	$\Sigma_{k+1\|k}$	$\Sigma'_{k+1\|k} = \omega_k \Sigma_{k+1\|k}, \quad \omega_k > 1$

The complete recursion for the **Discounted Mean-Variance Estimator** is as follows (the prediction and measurement update step have been combined)

$$\bar{p}_i(k+1|k+1) = \bar{p}_i(k|k) + 1/\lambda_k \Sigma_{ii}(k|k) \frac{\delta_{i,i'} - \bar{p}_i(k|k)}{\bar{p}_i(k|k)(1-\bar{p}_i(k|k))}$$
$$\Sigma_{ii}^{-1}(k+1|k+1) = \lambda_k \gamma_{k+1} \Sigma_{ii}^{-1}(k|k) + \frac{1}{\bar{p}_i(k|k)(1-\bar{p}_i(k|k))}$$

Note that since the posterior mean $\bar{p}_i(k+1|k+1)$ is directly dependent on $\Sigma_{ii}(k+1|k+1)$, scaling the variance by $1/\lambda_k$ will result in faster changes in the mean than if no scaling were applied. Table 4.3 shows this estimator also in terms of the individual prediction and measurement update steps.

Finally, this provides an explicit update for the Sigma Points, as the Dirichlet Sigma Points at the next time are calculated as

$$\begin{aligned}
\mathcal{Y}_0 &= \mathbf{E}[\mathbf{p}(k+1|k+1)] \\
\mathcal{Y}_i &= \mathbf{E}[\mathbf{p}(k+1|k+1)] + \beta_{\max} \left(\boldsymbol{\Sigma}(k+1|k+1)^{1/2}\right)_i \; \forall i = 1, \ldots, N \qquad (4\text{-}8) \\
\mathcal{Y}_i &= \mathbf{E}[\mathbf{p}(k+1|k+1)] - \beta_{\max} \left(\boldsymbol{\Sigma}(k+1|k+1)^{1/2}\right)_i \; \forall i = N+1, \ldots, 2N \qquad (4\text{-}9)
\end{aligned}$$

4.3.2 Intuition on the Dirichlet model

There is a fairly natural counts-based interpretation of covariance scaling for the Dirichlet distribution.

Proposition 11 *The discounted mean-variance recursion is equivalent to updating the Dirichlet counts as* $\alpha(k+1) = \lambda_k \alpha(k) + \delta_{i,i'}$.

Table 4.3: Discounted Mean variance recursion

	Mean-variance
Prediction	$\bar{p}_i(k+1\|k) = \bar{p}_i(k\|k)$ $\Sigma_{ii}^{-1}(k+1\|k) = \lambda_k\ \Sigma_{ii}^{-1}(k\|k)$
Measurement update	$\bar{p}_i(k+1\|k+1) = \bar{p}_i(k+1\|k) + \Sigma_{ii}(k+1\|k)\frac{\delta_{i,i'} - \bar{p}_i(k+1\|k)}{\bar{p}_i(k+1\|k)(1-\bar{p}_i(k+1\|k))}$ $\Sigma_{ii}^{-1}(k+1\|k+1) = \gamma_{k+1}\Sigma_{ii}^{-1}(k+1\|k) + \frac{1}{\bar{p}_i(k+1\|k)(1-\bar{p}_i(k+1\|k))}$
Combined updates	$\bar{p}_i(k+1\|k+1) = \bar{p}_i(k\|k) + 1/\lambda_k\Sigma_{ii}(k\|k)\frac{\delta_{i,i'} - \bar{p}_i(k\|k)}{\bar{p}_i(k\|k)(1-\bar{p}_i(k\|k))}$ $\Sigma_{ii}^{-1}(k+1\|k+1) = \lambda_k\gamma_{k+1}\Sigma_{ii}^{-1}(k\|k) + \frac{1}{\bar{p}_i(k\|k)(1-\bar{p}_i(k\|k))}$

Proof: Note that the variance of the Dirichlet implies that the following holds,

$$\begin{aligned} 1/\lambda_k\ \Sigma_{ii}(k+1|k) &= 1/\lambda_k\ \frac{\alpha_i(\alpha_0-\alpha_i)}{\alpha_0^2(\alpha_0+1)} \\ &= \frac{\lambda_k\alpha_i(\lambda_k\alpha_0-\lambda_k\alpha_i)}{\lambda_k^3\alpha_0^2(\alpha_0+1)} \end{aligned} \tag{4-10}$$

When $\alpha_0 \gg 1$ (this holds true very early in the estimation process), the above expression is approximately equal to

$$\frac{\lambda_k\alpha_i(\lambda_k\alpha_0-\lambda_k\alpha_i)}{\lambda_k^3\alpha_0^2(\alpha_0+1)} \approx \frac{\alpha_i(\alpha_0-\lambda_k\alpha_i)}{\alpha_0^2(\lambda_k\alpha_0+1)} \tag{4-11}$$

But this is nothing more than the variance of a Dirichlet distribution where the parameters are chosen as $\alpha'(k) = \lambda_k\ \alpha(k)$ instead of $\alpha(k)$. In fact, if the distribution is given by $f_D(\mathbf{p}|\alpha'(\mathbf{k})) = K\ \prod_{i=1}^N p_i^{\lambda_k\alpha_i}$, the first two moments are given by

$$\begin{aligned} \bar{p}_i &= \lambda_k\alpha_i/\lambda_k\alpha_0 = \alpha_i/\alpha_0 \\ \Sigma_{ii} &= \frac{\lambda_k^2\alpha_i(\alpha_0-\alpha_i)}{\lambda_k^2\alpha_0^2(\lambda_k\alpha_0+1)} = \frac{\alpha_i(\alpha_0-\alpha_i)}{\alpha_0^2(\lambda_k\alpha_0+1)} \end{aligned} \tag{4-12}$$

Hence, the discounted mean variance formulation can be interpreted as updating the counts in the following manner

$$\alpha_i(k+1) = \begin{cases} \lambda_k\alpha_i(k)+1 & \text{Transition from } i \text{ to } i' \\ \lambda_k\alpha_i(k) & \text{Else} \end{cases}$$

rather than $\alpha_i(k+1) = \alpha_i(k) + \delta_{i,i'}$ in the undiscounted version. ■

4.3.3 Switching Models

Now, consider a specialized case of a time-varying transition matrix: the case when the matrix switches at a single (but unknown) time T_{sw} from a model Π^- to a model Π^+. In this case, the Mean-Variance estimator will eventually converge to the true model.

The discounted mean-variance estimator does not exhibit the same convergence properties as the undiscounted estimator for arbitrary $\lambda_k < 1$; this includes the case of constant λ_k, where $\lambda_k = \lambda < 1$. This is because the estimator has been modified to always maintains some level of *uncertainty* by rescaling the uncertainty. This can be seen in Figure 4-1 where the estimator gain is plotted as a function of time for a simple adaptation example, for different values of (constant) λ.

In particular, it can be shown that the estimator will constantly be responding to the most recent observations, and will only converge if the following proposition holds.

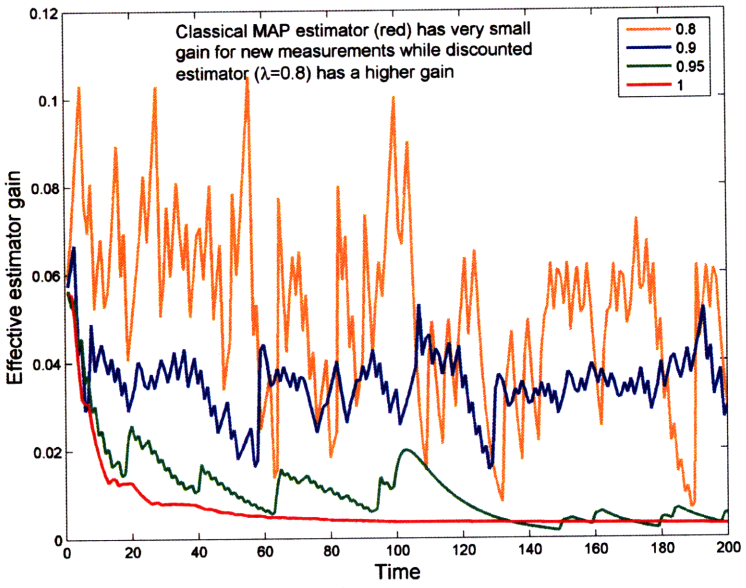
Proposition 12 *The discounted estimator converges if* $\lim_{k\to\infty} \lambda_k = 1$.

Proof: The asymptotic variance, $\Sigma_{ii}(\infty) = \lim_{k\to\infty} \Sigma_{ii}(k)$ is given by

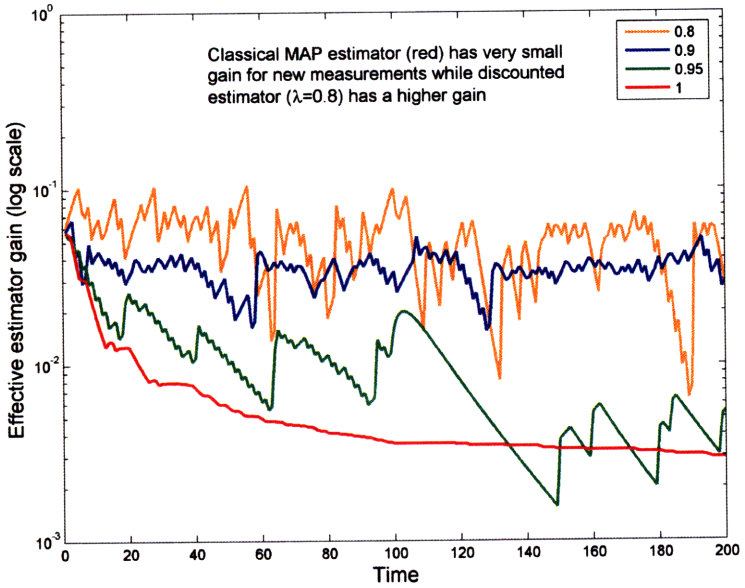
$$\Sigma_{ii}(\infty) = \lim_{k\to\infty} \frac{(1 - \lambda_k\, \gamma_{k+1})}{2 - \lambda_k} \bar{p}_i(k+1)(1 - \bar{p}_i(k+1))$$

and will asymptotically reach zero if both $\lim_{k\to\infty} \gamma_{k+1} = 1$ and $\lim_{k\to\infty} \lambda_k = 1$. If $\lambda_k = \lambda < 1$, the variance will not converge to 0; however, if $\lim_{k\to\infty} \lambda_k = 1$, the discounted mean estimator will converge to the undiscounted form, and hence the estimator will converge to the true parameter.

It is shown in the next simulations that using a constant λ_k still provides good estimates of the true parameter, but we caution that to achieve convergence, λ_k should be chosen such that $\lim_{k\to\infty} \lambda_k = 1$. Such a choice could be $\lambda_k = 1 - \lambda^k$, where $\lambda < 1$.



(a) Estimator gain does not converge for constant $\lambda < 1$



(b) Logarithm of estimator gain

Fig. 4-1: Estimator gain constantly responds to new observations for constant $\lambda < 1$

4.4 Robust Replanning

While the discounted form of the estimator results in a much faster response to the change in the parameter, and is a useful result in its own right, it is also important to remember that the outputs of this estimator will also be the inputs to some decision-making system. This is where the ultimate benefits of a faster estimator will be most apparent. We revisit the Markov Decision Process framework of the first chapter in order to demonstrate the utility of these faster updates on the overall control solution.

Recall that the first chapter emphasized the robustness issue associated with the uncertainty in the transition probability of a Markov Decision Process, and by using the Dirichlet Sigma Points, we were able to approximate the uncertainty set of the transition probability, and generate robust solutions. By updating the Dirichlet Sigma Points with the result from the previous section, one can robustly replan using the latest information. For small to medium problems, MDP solutions can be found in reasonable time.

There are many choices for re-planning efficiently using model-based methods, such as Real Time Dynamic Programming (RTDP) [5, 35]. These papers assumed that the transition probabilities were unknown, and were continually updated through an agent's actions in the state space. The complete value iteration was not repeated at each update, however. For computational considerations, only a single sweep of the value iteration was performed at each measurement update, and the result of Gullapalli [35] shows that if each state and action are executed infinitely often, then the (asynchronous) value iteration algorithm converges to the true value function. As long as the estimator that was updating the estimates of the transition probabilities was convergent, the optimal policy was guaranteed to converge.

In this section, we consider the full re-planning problem (though the re-planning problem as mentioned above could also be implemented), but the re-planning is done *robustly*, by taking into account the residual uncertainty in the transition probabilities. This results in the robust replanning algorithm (Algorithm 7).

Algorithm 6 Robust Replanning

Initialize Dirichlet Sigma Points
while Not finished **do**
Using discounted estimator, update estimates

$$\bar{p}_i(k+1|k+1) = \bar{p}_i(k|k) + 1/\lambda_k \Sigma_{ii}(k|k) \frac{\delta_{i,i'} - \bar{p}_i(k|k)}{\bar{p}_i(k|k)(1-\bar{p}_i(k|k))}$$
$$\Sigma_{ii}^{-1}(k+1|k+1) = \lambda_k \gamma_{k+1} \Sigma_{ii}^{-1}(k|k) + \frac{1}{\bar{p}_i(k|k)(1-\bar{p}_i(k|k))}$$

For each uncertain row of the transition probability matrix, provide mean and covariance

$$\begin{aligned} \mathcal{Y}_0 &= \mathbf{E}[\mathbf{p}] \\ \mathcal{Y}_i &= \mathbf{E}[\mathbf{p}] + \beta_{\max} \left(\boldsymbol{\Sigma}^{1/2}\right)_i \; \forall i = 1, \ldots, N \\ \mathcal{Y}_i &= \mathbf{E}[\mathbf{p}] - \beta_{\max} \left(\boldsymbol{\Sigma}^{1/2}\right)_i \; \forall i = N+1, \ldots, 2N \end{aligned} \tag{4-13}$$

Solve robust MDP

$$\min_u \max_{\Pi \in \mathcal{Y}} \mathbf{E}\left[J_u\right] \tag{4-14}$$

Return
end while

4.4.1 Convergence

We can use a similar argument from Theorem 1 of Gullapalli [35] to note that because the discounted estimator in fact converges in the limit of a large number of observations (with appropriate choice of λ), and the covariance Σ can eventually be driven to 0, then each of the Dirichlet Sigma Points will collapse to the singleton, unbiased estimate of the true transition probabilities. This means that the model will have converged, and that the robust solution will in fact have converged to the optimal value function. We address the implementation of this algorithm in the flight experiments.

4.4.2 Nominal Replan

Note that in the case that a user is not interested in robustness issues, then the above algorithm can also be implemented in a RTDP-like framework, where only the estimates (and not the covariances) are used in the optimization. In such a way, the algorithm returns the optimal policy at each time step given the current information.

We demonstrate this in the next numerical simulations.

4.5 Numerical Simulations

This section presents some numerical simulations showing the responsiveness of the discounted mean-variance estimator. In the first set of examples, we show a set of runs showing the identification of an underlying (non-stationary transition matrix) that switches from Π_1^- to Π_1^+ at some unknown time T_{sw}. We also show that the discounted mean-variance estimator responds quicker to the change than other estimators, such as the undiscounted version or a finite memory estimator. In the second set of examples, we show an implementation of the discounted mean-variance formulation in an infinite horizon Markov Decision Process, where at each time that the transition matrix is identified, a new control policy is calculated. The optimal performance of each policy converges quicker when the discounted mean-variance approach is used to identify the transition matrix.

4.5.1 Transition Matrix Identification

This first example has an underlying transition matrix that switches at some unknown time T_{sw}. First, we show the benefit of using the discounted version of the estimator over the undiscounted estimator. This is shown in Figure 4-2 where the discouned estimator (blue) responds to the change in transition matrix almost instantly at $t = 50$ seconds, and after 20 seconds from the switch, has a 50% error ($\hat{p} = 0.7$) from the true parameter $p = 0.8$. The undiscounted estimator (red) has a 50% error after 50 seconds, and is much slower.

Next, compare the identification of this model with a finite memory estimator which calculated by storing all observed transitions in previous M time steps,

$$\hat{\alpha}_i(k) = \sum_{j=k-M+1}^{k} \delta_{i,i'}^j \tag{4-15}$$

where $\delta_{i,i'}^j$ is unity if a transition occurred from state i to state i' at time j. The mean

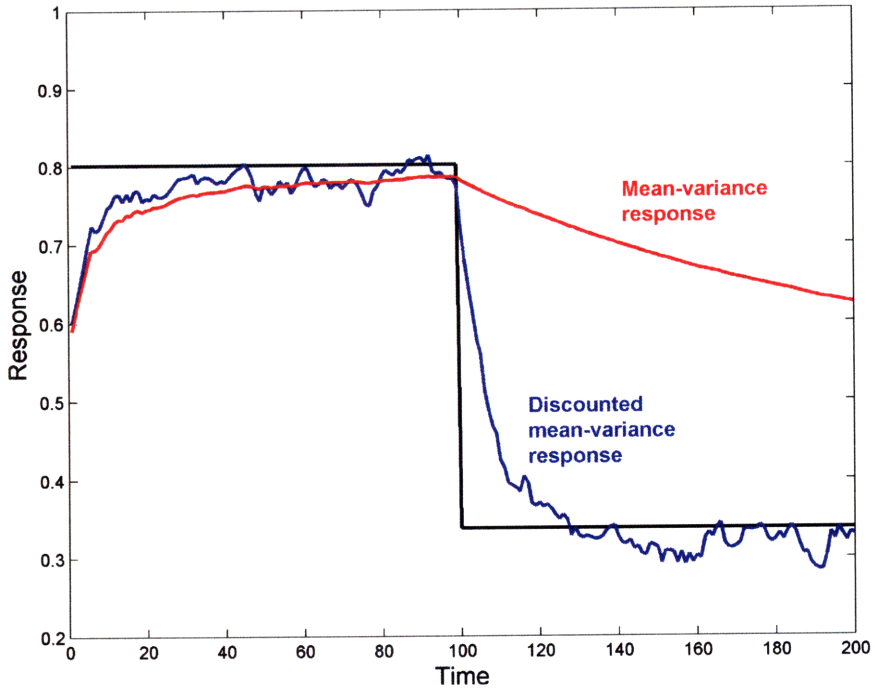


Fig. 4-2: Discounted estimator (blue) has a faster response at the switch time than undiscounted estimator (red).

and variance are calculated using

$$\begin{aligned} \bar{p}_i(k) &= \frac{\hat{\alpha}_i}{\hat{\alpha}_0} \\ \Sigma_{ii}(k|k) &= \frac{\hat{\alpha}_i(\hat{\alpha}_0 - \hat{\alpha}_i)}{\hat{\alpha}_0^2(\hat{\alpha}_0 + 1)} \end{aligned}$$

where $\hat{\alpha}_0 = \sum_i \hat{\alpha}_i(k)$. Note that the finite memory estimator does not include information that is older than M time steps. The three estimators compared in the next simulations are

- Estimator #1: Undiscounted estimator
- Estimator #2: Discounted estimator (varying λ_k)
- Estimator #3: Finite memory estimator (varying M)

Figure 4-4 shows the average of 200 different simulations for a sample problem, and compares the response of the finite memory estimator with the discounted estimator.

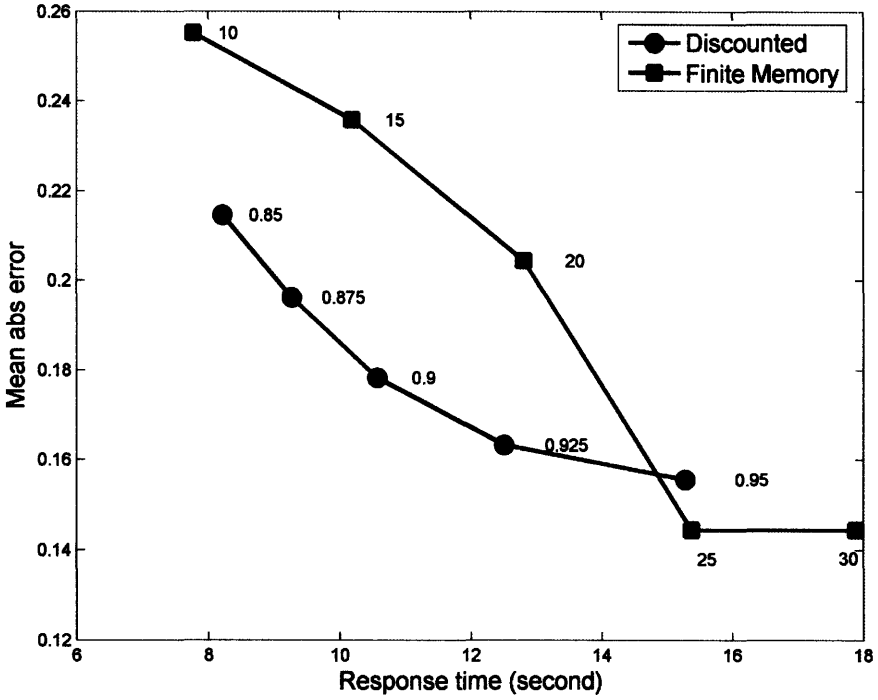
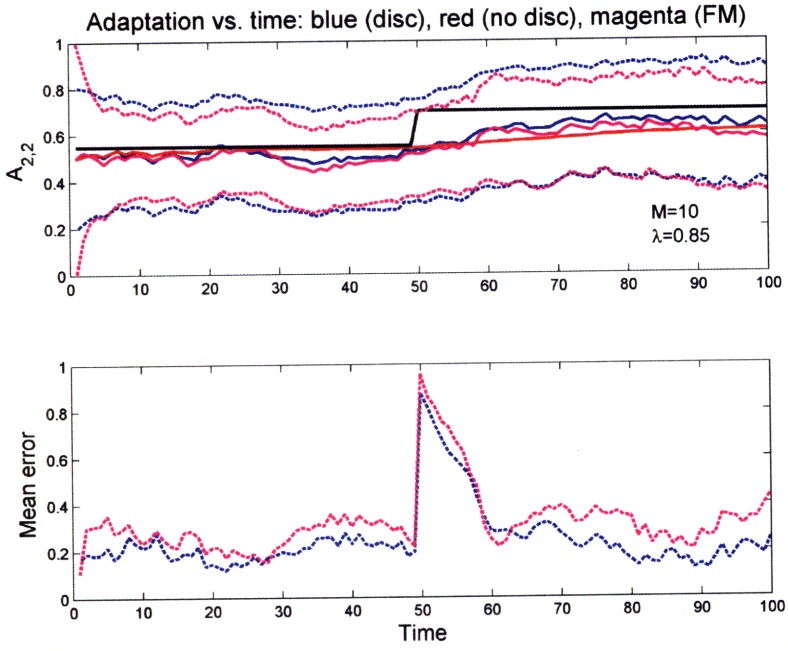


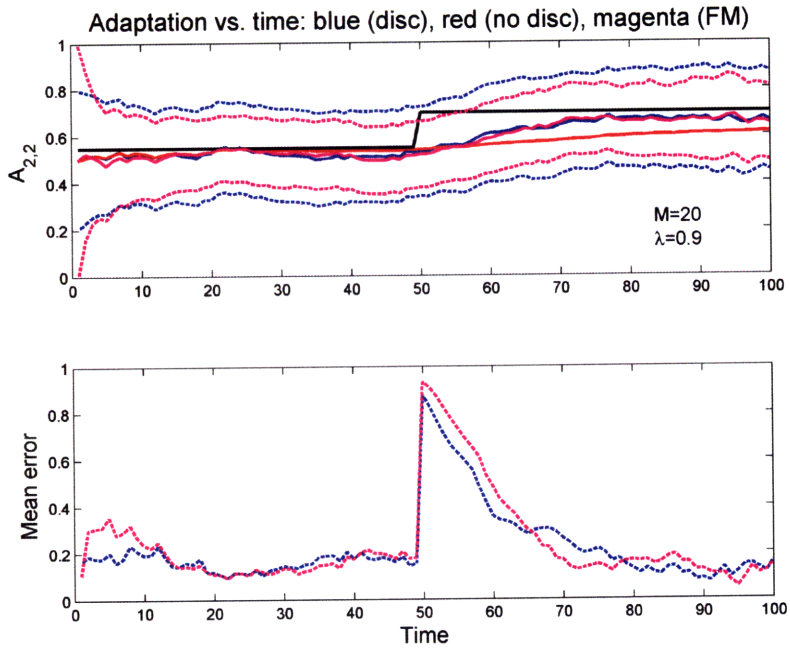
Fig. 4-3: Mean absolute error vs. response delay plot showing the discounted estimator (square) has better performance than finite memory estimator. Each data point is an average of 200 simulations with random transition matrices for a 2x2 case.

Figure 4-4(a) (top) shows the the time response required to detect the change in the model, for $\lambda = 0.85$ and $M = 10$, while Figure 4-4(a) (bottom) shows the mean absolute error of the two estimators. The discounted estimator has a smaller absolute error. Figure 4-4(b) shows the analogous figure for a case of $\lambda = 0.90$ and $M = 20$. Note that the discretizations for the finite horizon M and the discount factor λ are not related in any obvious manner, and are only used to discretize the parameter spaces for the two estimators.

Figure 4-3 presents a summary of the results for different M and λ values, where each data point corresponds to 200 different simulations of random transition matrices for a 2×2 identification problem. The plot compares mean absolute error of the estimator at the switch time to the response time of the estimator to achieve a 20% estimation error. The finite horizon length was varied in $M = \{10, 15, 20, 25, 30\}$ while the (constant) discount was varied in $\lambda = \{0.85, 0.875, 0.90, 0.925, 0.95\}$.



(a) $\lambda = 0.85$, finite memory estimator $M = 10$



(b) $\lambda = 0.90$, finite memory estimator $M = 20$

Fig. 4-4: Adaptation comparison of finite memory estimator and discounted estimator.

Table 4.4: Mean / Standard Deviation of Absolute Error

λ	Mean	Variance	Min	Max
0.85	0.215	0.099	0.018	0.632
0.875	0.196	0.096	0.011	0.601
0.90	0.178	0.094	0.005	0.577
0.925	0.163	0.094	0.013	0.563
0.95	0.156	0.096	0.011	0.587
M	Mean	Variance	Min	Max
10	0.255	0.119	0.014	0.659
15	0.236	0.108	0.017	0.777
20	0.204	0.102	0.004	0.586
25	0.144	0.084	0.009	0.463
30	0.144	0.084	0.009	0.463

The results clearly show the benefit of using the discounted estimator because for most reasonable values of the response time, the mean absolute error of the discounted estimator is lower on average than the finite memory estimator. Table 4.4 presents the summary statistics of these simulations in terms of mean absolute error, standard deviation of absolute error, and min/max of the absolute error. A two-sided T-test showed that the difference between the discounted estimator and the finite memory estimator up to $\lambda = 0.925$ and $M = 20$ was statistically significant at $p < 0.01$. Also note that the use of a finite memory estimator generally requires that all the observed transitions in the previous M steps be stored. For large M and a large system, this may in fact be impractical; this memory storage is not required in the discounted mean-variance formulation, where only storing the $\alpha_i(k)$ is required (if using the counts-based formulation).

4.5.2 Online MDP Replanning

This section considers a machine repair/replacement problem [10] driven by a time-varying transition matrix, posed as a Markov Decision Process (MDP). Similar to the previous example, the transition model is assumed to switch from model Π_1^- to model Π_1^+ at an unknown time T_{sw}. The estimate of the transition matrix is updated

at each time step with the most recent observations, and the optimal policy for the DP is re-calculated using the current estimate.[2]

Problem Statement A machine can take on one of two states x_k at time k: i) the machine is either *running* ($x_k = 1$), or ii) broken (not running, $x_k = 0$). If the machine is running, a profit of \$100 is made. The control options available to the user are the following: if the machine is running, a user can choose to either i) perform maintenance (abbreviated as $u_k = m$) on the machine (thereby decreasing the likelihood the machine failing in the future), or ii) leave the maching running without maintenance ($u_k = n$). The choice of maintenance has cost, C_{maint}, e.g., the cost of a technician to maintain the machine.

If the machine is broken, two choices are available to the user: i) repair the machine ($u_k = r$), or ii) completely replace the machine ($u_k = p$). Both of these two options come at a price, however; machine repair has a cost C_{repair}, while machine replacement is $C_{replace}$, where for any sensible problem specification, the price of replacement is greater than the repair cost $C_{replace} > C_{repair}$. If the machine is replaced, it is *guaranteed* to work for at least the next stage.

For the case of the machine running at the current time step, the state transitions are governed by the following model

$$\Pr\ (x_{k+1} = \text{fails} \mid x_k\text{=running}, u_k\text{=m}) = \gamma_1$$
$$\Pr\ (x_{k+1} = \text{fails} \mid x_k\text{=running}, u_k\text{=n}) = \gamma_2$$

For the case of the machine not running at the current time step, the state transition are governed by the following model

$$\Pr\ (x_{k+1} = \text{fails} \mid x_k\text{=fails}, u_k\text{=r}) = \gamma_3$$
$$\Pr\ (x_{k+1} = \text{fails} \mid x_k\text{=fails}, u_k\text{=p}) = 0$$

[2]This problem is sufficiently small that the policy can be quickly recalculated. For larger problems, this may not be the case, and one might have to resort to Real-Time Dynamic Programming (RTDP) techniques, such as in Barto et al [5].

Note that, consistent with our earlier statement that machine replacement guarantees machine function at the next time step, the transition matrix for the replacement is deterministic. From these two models, we can completely describe the transition matrix if the machine is running or not running at the current time step:

$$\text{Machine Running } (x_k = 1), \Pi_1 : \begin{bmatrix} 1-\gamma_1 & \gamma_1 \\ 1-\gamma_2 & \gamma_2 \end{bmatrix}$$

$$\text{Machine Not Running } (x_k = 0), \Pi_2 : \begin{bmatrix} 1-\gamma_3 & \gamma_3 \\ 1 & 0 \end{bmatrix}$$

The objective is to find an optimal control policy such that $u_k(x_k = 0) \in \{$ r, p $\}$ if the machine is not running, and $u_k(x_k = 1) \in \{$ m, n $\}$ if the machine is running, for each time step. The state of the machine is assumed to be perfectly observable, and this can be solved via dynamic programming.

Results The transition matrix for time $t < T_{sw}$ was

$$\Pi_1^- = \begin{bmatrix} 0.05 & 0.95 \\ 0.3 & 0.7 \end{bmatrix},$$

while for $t \geq T_{sw}$, the transition matrix was

$$\Pi_1^+ = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}.$$

The response speeds of the two types of estimators can be calculated by evaluating the difference in the mean objective function and. The optimal policy $u^*(k, s)$ and optimal cost $J^*(k, s)$ are calculated at each time step k and simulation s using i) the discounted estimator $(u_d^*(k, s), J_d^*(k, s))$ and the undiscounted estimator $(u_u^*(k, s), J_u^*(k, s))$. The mean of the objective function is calculated as follows

$$\bar{J}_u(k) = \frac{1}{N_s} \sum_{s=1}^{N_s} J_u^*(k, s),$$

The mean of the objective function for $\lambda_k = 0.90$ is shown in Figure 4-5 while for $\lambda_k = 0.95$ is shown in Figure 4-6. The discounted estimator response (blue) is shown to be much faster than the undiscounted response (red) at the switch time of T_{sw} 10 seconds.

4.6 Conclusions

This chapter has presented a formulation for the identification on non-stationary Markov Chains that uses filtering insight to speed up the response of classical ML-based estimator. We have shown that the addition of an artificial pseudo-noise like term is equivalent to a fading of the transition observations using the Dirichlet model; this fading of the observations is similar to fading mechanisms proposed in time-varying parameter estimation techniques, but our pseudo-noise-based derivation provides an alternative motivation for actually fading these Dirichlet counts in a perfectly observable system. Additional work will account for measurement noise addition, and the sensitivity of the overall estimator to the discount factor.

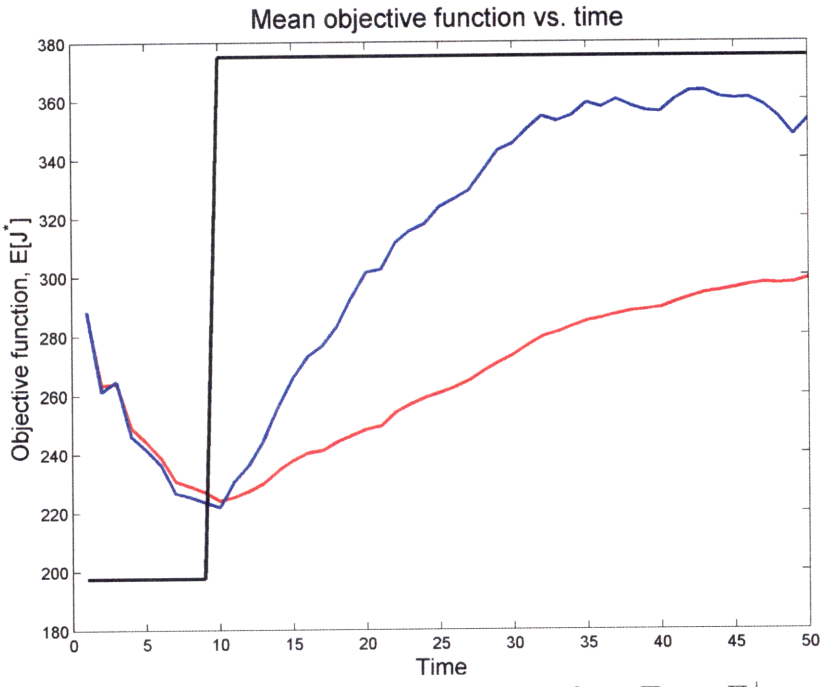


Fig. 4-5: At $t = 10$, the transition matrix changes from Π_1^- to Π_1^+, and the MDP solution (after replanning at each observation) using the discounted estimator ($\lambda = 0.90$, blue) converges in the neighborhood of the optimal objective J^* quicker than with using the undiscounted estimator (red)

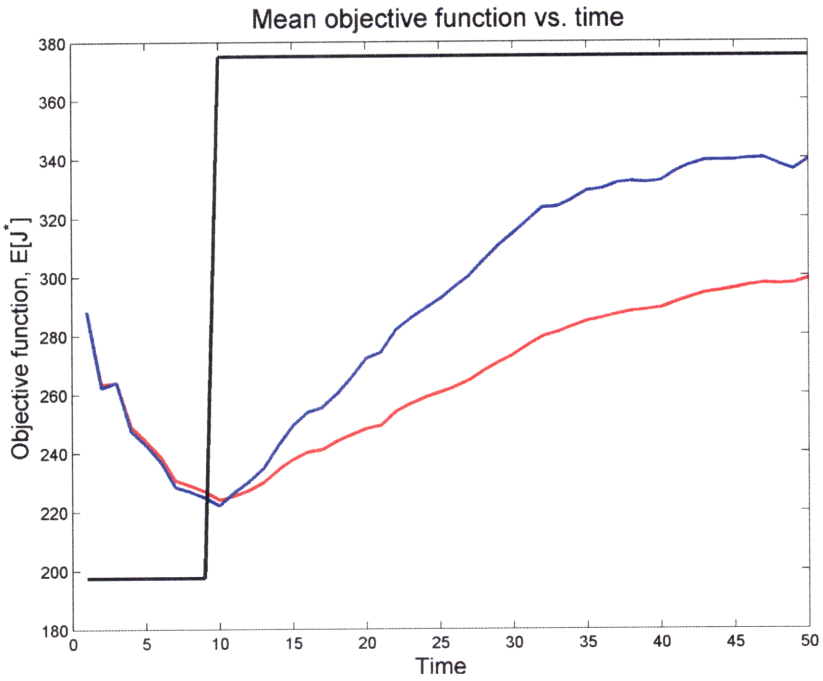


Fig. 4-6: At $t = 10$, the transition matrix changes from Π_1^- to Π_1^+, and the MDP solution (after replanning at each observation) using the discounted estimator ($\lambda = 0.95$, blue) converges in the neighborhood of the optimal objective J^* quicker than with using the undiscounted estimator (red)

Chapter 5

Persistent Surveillance Implementation

5.1 Introduction

This chapter describes the hardware implementation using the robust replanning MDP formulation in the application of multiple unmanned aerial vehicles persistent surveillance missions. Experimental validation of the proposed algorithms is critical in verifying that the algorithms truly are implementable in real-time with actual hardware. Recent work by other authors has formulated UAV persistent surveillance missions as an MDP. The essence of the surveillance mission is the time maximization of UAV's coverage of a desired region in the environment, while accounting for fuel constraints and random vehicle failures. In particular, the fuel flow of the vehicles is governed by a probabilistic process that stochastically characterizes how much fuel will be burned at each time step. We also account for the time-variability of this fuel flow probability, to more realistically model real-life effects, such as vehicle degradation over time, as well as adversarial effects.

In this section, we apply the Dirichlet Sigma Points with replanning to the persistent surveillance mission when the fuel flow probability is uncertain. The effect of this uncertainty can range from slightly degraded total coverage to an increased level of vehicle crashes as the UAVs run out of fuel prematurely. Use of the Dirichlet Sigma

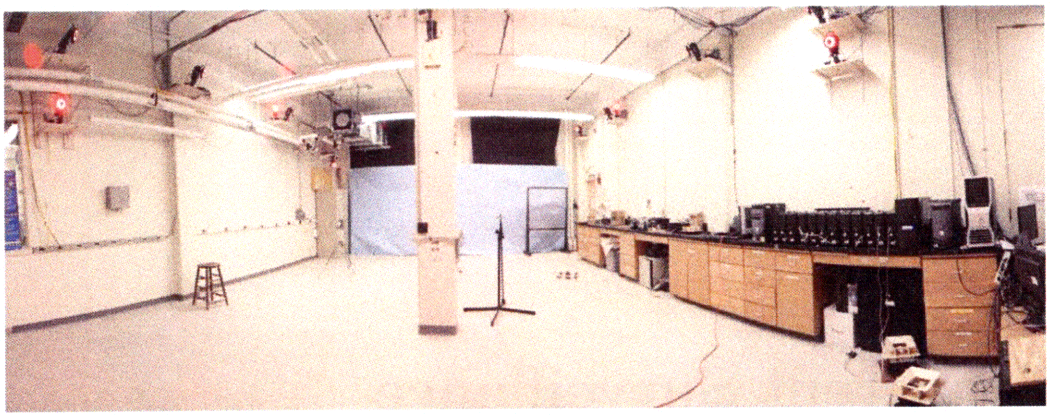


Figure 5-1: RAVEN Testbed

Points with the replanning mechanism is shown to mitigate the effect of vehicle failures by conservatively replanning with new observations as the fuel flow probability varies over time.

5.2 RAVEN Testbed

The Real-Time indoor Autonomous Vehicle test ENvironment (RAVEN) testbed is an advanced testbed that is used for rapid prototyping of new control and estimation algorithms, ranging from aggressive flight control [64] to multi-vehicle coordinated control [14, 83]. At the heart of RAVEN is the precise positioning of a Vicon MX camera system [83] that can accurately position a vehicle to within tenths of millimeters. The RAVEN testbed is composed of a wide variety of flight vehicles, but the ones used in this chapter were quadrotor UAVs. For a much more detailed description of the testbed, the reader is referred to the work of Valenti and Bethke [39, 54, 83, 84].

5.3 Persistent Surveillance Problem

The problem description of the persistent surveillance mission is as follows. A desired number N_{des} of UAVs is required to maintain coverage of a surveillance area (see Figure 5-8). The vehicles start from a base location, traverse one or more intermediate

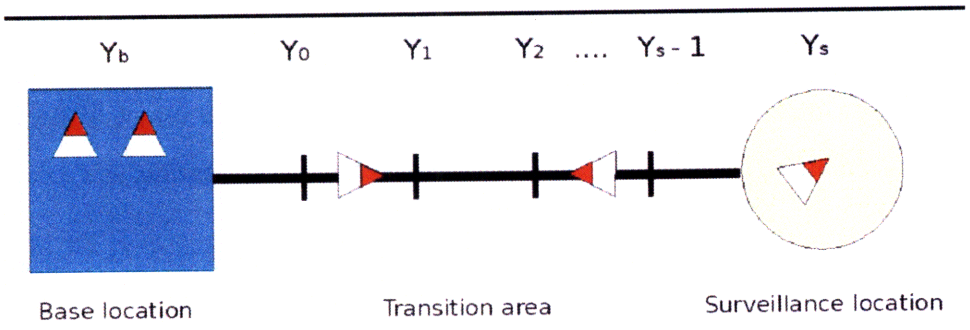


Fig. 5-2: Persistent surveillance mission: vehicles must take off from a base location, Y_b, fly through intermediate locations, and finally reach the desired surveillance area Y_s [14].

areas (modeling the finite amount of time it takes a vehicle to reach the surveillance location), and finally arrive to the surveillance area.

At each time step, the vehicles have three actions available to them: they can either 1) return close to base, 2) approach the surveillance area, or 3) do nothing, at which point the vehicle remains in its current location. Once the vehicle take off from the base area, they lose fuel in a stochastic manner. With probability p_{nom}, the vehicle will lose fuel at a rate of $\dot{F} = 1$ unit per time step. With probability $1 - p_{nom}$, the vehicles will lose fuel at an off-nominal rate of $\dot{F} = 2$ units per time step.

5.4 MDP Formulation

Given the qualitative description of the persistent surveillance problem, an MDP can now be formulated [13, 14]. The MDP is specified by $(\mathcal{S}, \mathcal{A}, P, g)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P_{\mathbf{xy}}(\mathbf{u})$ gives the transition probability from state $\mathbf{x}$ to state $\mathbf{y}$ under action $\mathbf{u}$, and $g(\mathbf{x}, \mathbf{u})$ gives the cost of taking action $\mathbf{u}$ in state $\mathbf{x}$. Future costs are discounted by a factor $0 < \alpha < 1$. A policy of the MDP is denoted by $\mu : \mathcal{S} \rightarrow \mathcal{A}$. Given the MDP specification, the problem is to minimize the so-called cost-to-go function J_μ over the set of admissible policies Π:

$$\min_{\mu \in \Pi} J_\mu(\mathbf{x}_0) = \min_{\mu \in \Pi} \mathbb{E}\left[\sum_{k=0}^{\infty} \alpha^k g(\mathbf{x}_k, \mu(\mathbf{x}_k))\right].$$

5.4.1 State Space $\mathcal{S}$

The state of each UAV is given by two scalar variables describing the vehicle's flight status and fuel remaining. The flight status y_i describes the UAV location,

$$y_i \in \{Y_b, Y_0, Y_1, \ldots, Y_s, Y_c\}$$

where Y_b is the base location, Y_s is the surveillance location, $\{Y_0, Y_1, \ldots, Y_{s-1}\}$ are transition states between the base and surveillance locations (capturing the fact that it takes finite time to fly between the two locations), and Y_c is a special state denoting that the vehicle has crashed.

Similarly, the fuel state f_i is described by a discrete set of possible fuel quantities,

$$f_i \in \{0, \Delta f, 2\Delta f, \ldots, F_{max} - \Delta f, F_{max}\}$$

where Δf is an appropriate discrete fuel quantity. The total system state vector $\mathbf{x}$ is thus given by the states y_i and f_i for each UAV, along with r, the number of requested vehicles:

$$\mathbf{x} = (y_1, y_2, \ldots, y_n; f_1, f_2, \ldots, f_n; r)^T$$

5.4.2 Control Space $\mathcal{A}$

The controls u_i available for the i^{th} UAV depend on the UAV's current flight status y_i.

- If $y_i \in \{Y_0, \ldots, Y_s - 1\}$, then the vehicle is in the transition area and may either move away from base or toward base: $u_i \in \{\text{“}+\text{”}, \text{“}-\text{”}\}$
- If $y_i = Y_c$, then the vehicle has crashed and no action for that vehicle can be taken: $u_i = \emptyset$
- If $y_i = Y_b$, then the vehicle is at base and may either take off or remain at base: $u_i \in \{\text{“take off”}, \text{“remain at base”}\}$

- If $y_i = Y_s$, then the vehicle is at the surveillance location and may loiter there or move toward base: $u_i \in \{\text{“loiter”}, \text{“}-\text{”}\}$

The full control vector $\mathbf{u}$ is thus given by the controls for each UAV:

$$\mathbf{u} = (u_1, \ldots, u_n)^T \tag{5-1}$$

5.4.3 State Transition Model P

The state transition model P captures the qualitative description of the dynamics given at the start of this section. The model can be partitioned into dynamics for each individual UAV.

The dynamics for the flight status y_i are described by the following rules:

- If $y_i \in \{Y_0, \ldots, Y_s - 1\}$, then the UAV moves one unit away from or toward base as specified by the action $u_i \in \{\text{“}+\text{”}, \text{“}-\text{”}\}$.
- If $y_i = Y_c$, then the vehicle has crashed and remains in the crashed state forever afterward.
- If $y_i = Y_b$, then the UAV remains at the base location if the action “remain at base” is selected. If the action “take off” is selected, it moves to state Y_0.
- If $y_i = Y_s$, then if the action “loiter” is selected, the UAV remains at the surveillance location. Otherwise, if the action “−” is selected, it moves one unit toward base.
- If at any time the UAV's fuel level f_i reaches zero, the UAV transitions to the crashed state ($y_i = Y_c$).

The dynamics for the fuel state f_i are described by the following rules:

- If $y_i = Y_b$, then f_i increases at the rate $\dot{F}_{refuel}$ (the vehicle refuels).
- If $y_i = Y_c$, then the fuel state remains the same (the vehicle is crashed).

- Otherwise, the vehicle is in a flying state and burns fuel at a stochastically modeled rate: f_i decreases by $\dot{F}_{burn}$ with probability p_{nom} and decreases by $2\dot{F}_{burn}$ with probability $(1 - p_{nom})$.

5.4.4 Cost Function g

The cost function $g(\mathbf{x}, \mathbf{u})$ penalizes three undesirable outcomes in the persistent surveillance mission. First, any gaps in surveillance coverage (i.e. times when fewer vehicles are on station in the surveillance area than were requested) are penalized with a high cost. Second, a small cost is associated with each unit of fuel used. This cost is meant to prevent the system from simply launching every UAV on hand; this approach would certainly result in good surveillance coverage but is undesirable from an efficiency standpoint. Finally, a high cost is associated with any vehicle crashes. The cost function can be expressed as

$$g(\mathbf{x}, \mathbf{u}) = C_{loc} \max\{0, (r - n_s(\mathbf{x}))\} + C_{crash} n_{crashed}(\mathbf{x}) + C_f n_f(\mathbf{x})$$

where:

- $n_s(\mathbf{x})$: number of UAVs in surveillance area in state $\mathbf{x}$,
- $n_{crashed}(\mathbf{x})$: number of crashed UAVs in state $\mathbf{x}$,
- $n_f(\mathbf{x})$: total number of fuel units burned in state $\mathbf{x}$,

and C_{loc}, C_{crash}, and C_f are the relative costs of loss of coverage events, crashes, and fuel usage, respectively.

5.5 Robustness

In this first section, we address the issue of sensitivity of the persistent surveillance mission to the nominal fuel transition flow probability. At first glance, it may not be surprising that the mission is in fact sensitive to p_{nom}, the actual sensitivity of

the coverage time of the mission is fairly dramatic. To empirically determine this sensitivity, p_{nom} was discretized into a finite number of values. These values were chosen under the assumption that p_{nom} was an uncertain quantity with a nominal value of 0.80, and we used the Dirichlet Sigma Points of Chapter 2 to generate a set of different p_{nom} for different values of β. The discretizations resulted in the values of $p_{nom} : \{0.605, 0.7, \ldots, 0.995\}$.

The optimal policy and other characteristics of the persistent surveillance mission are very sensitive to the precise value of the parameter p_{nom}. Figure 5-3 (top) demonstrates the sensitivity of the coverage time of the mission (the total number of time steps in which a single UAV was at the surveillance location) as a function of p_{nom}. For values of $p_{nom} < 0.9$, typical coverage times for a 50-time step mission can range from 25 to 30 time steps, while for values of $p_{nom} > 0.9$, the coverage times can increase to almost 47 time steps.

Figure 5-3 (bottom) shows the impact of a mismatched transition model on the overall mission coverage times. The modeled value for p_{nom} is shown on the "Modeled" axis, while the true system operated under a value of p_{nom} on the "Actual" axis. When the modeled p_{nom} is less than the actual p_{nom}, this results in more conservative policies, where the control policy recalls the UAVs to base well before they were out of fuel, because it assumes they will use a lot of fuel on the flight back to base. This results in fewer crashes, but also led to decreased surveillance coverage since the vehicles spend less time in the surveillance area. Conversely, riskier policies are the result when the modeled p_{nom} is greater than the actual p_{nom}, since the control policy assumes the UAVs can fly for longer than they actually are capable of. This leads to significant coverage losses, since the UAVs tend to run out of fuel and crash more frequently.

A seemingly counter-intuitive result is that the optimal coverage time need not occur along the diagonal "Actual=Modeled". This is however easily resolved as the off-nominal fuel transitions are simply occurring less frequently, and even though the vehicles return to base with residual fuel, they immediately take-off again and return to the surveillance area. The almost immediate refueling therefore partially mitigates

the performance loss that is suffered by the mismatched estimates.

5.6 Adaptation Flight Experiments

The prior results showed that value of the parameter p_{nom} has a strong effect on the optimal policy, and in particular, how mismatches between the true parameter value and the value used to compute the optimal policy can lead to degraded performance when implemented in the real system. Therefore, in order to achieve better performance in the real system, some form of adaptation mechanism is necessary to enable the planner to adjust the policy based on observations of the true parameter values. These observations cannot be obtained prior to the start of operation of the real system, so this adaptation must be done online.

Flight experiments were flown on RAVEN to demonstrate the advantage of an adaptation mechanism. Multiple tests were implemented, involving step changes to the probability of nominal fuel flow. This reflected the more realistic set of scenarios where a vehicle could suffer damage throughout the course of the mission.

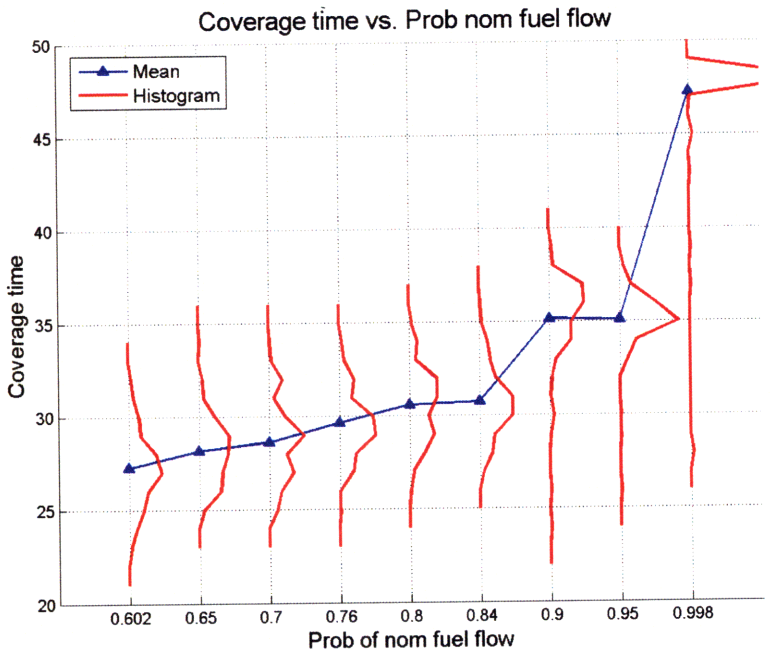
The estimator of Chapter 4 was implemented for estimating this probability. For our hardware implementation, the estimator was implemented in terms of the fading memory interpretation of the Dirichlet counts. Namely, we initialized the estimator with parameters $(\alpha(0), \beta(0))$, and our a priori density was given by

$$f_B(p \mid \alpha(0), \beta(0)) = K\, p^{\alpha(0)-1}(1-p)^{\beta(0)-1} \tag{5-2}$$

For the undiscounted estimator, the parameters were updated as

$$\begin{aligned} \alpha(k+1) &= \alpha(k) + \delta \\ \beta(k+1) &= \beta(k) + (1-\delta) \end{aligned}$$

where $\delta = 1$ if a nominal transition was observed (incrementing $\alpha(k)$ by 1), and $\delta = 0$



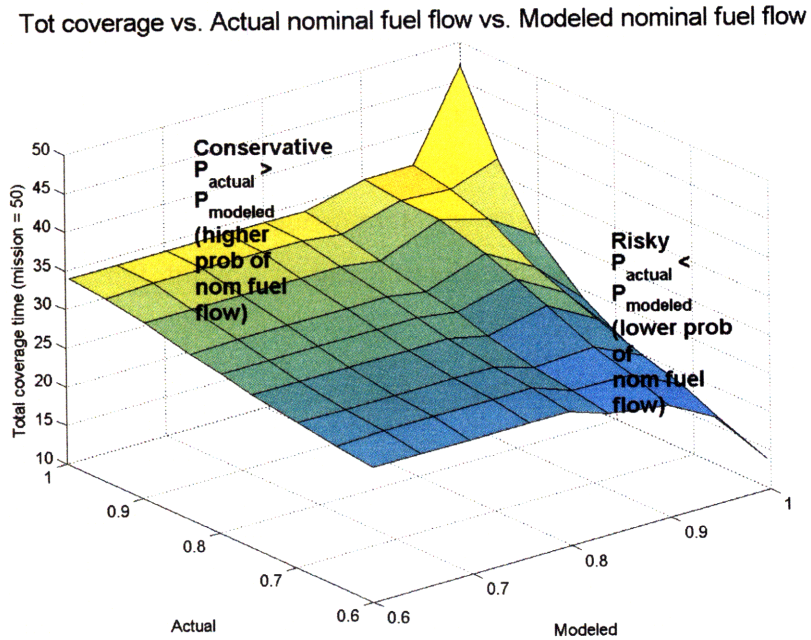


Fig. 5-3: Sensitivity of total coverage to nominal probability of fuel flow (left) and mismatched probability of nominal fuel flow (right)

if an off-nominal transition was observed (incrementing $\beta(k)$ by 1).

$$\alpha(k+1) = \lambda\, \alpha(k) + \delta$$
$$\beta(k+1) = \lambda\, \beta(k) + (1-\delta)$$

such that at each iteration, the parameters were faded by the factor $\lambda < 1$. As described in Chapter 4, this tuning parameter is used to vary the response speed of the estimator.

The following tests were performed to validate the effectiveness of adaptation in the hardware testbed:

- Test 1: The probability was changed in mid-mission from $p_{nom} = 1$ to $p_{nom} = 0$, and the estimators were anlayzed for responsiveness.
- Test 2: Incremental step changes, where the probability was initialized to $p_{nom} = 1$ and decreased by 0.3 approximately every 5 time steps (corresponding to approximately 2 minutes of actual flight time).

In these tests, the optimal policy was *recomputed* within 4 time steps of the updated estimates. At each time step, the previous policy and optimal cost were used as initial condition for the new value iteration. More details about warm-starting the optimization using the previously calculated policy can be found in Ref. [13].

5.6.1 Test 1

The next scenario demonstrated the ability of the adaptation mechanism to adjust to actual model changes during the mission, such as might be observed if the vehicles were damaged in flight. In this scenario, the vehicles were initialized with a $p_{nom} = 1$ and the model was changed to $p_{nom} = 0$ after approximately 2 minutes (5 time steps), mimicking adversarial actions (such as anti-aircraft fire) and/or system degradation over time. The change in the probability estimate is shown in Figure 5-4 for three different choices of $\lambda = \{0.6, 0.8, 1\}$. It can be seen that the classical estimation ($\lambda = 1$) results in a very slow change in the estimate, while $\lambda = 0.8$ is within 20% of

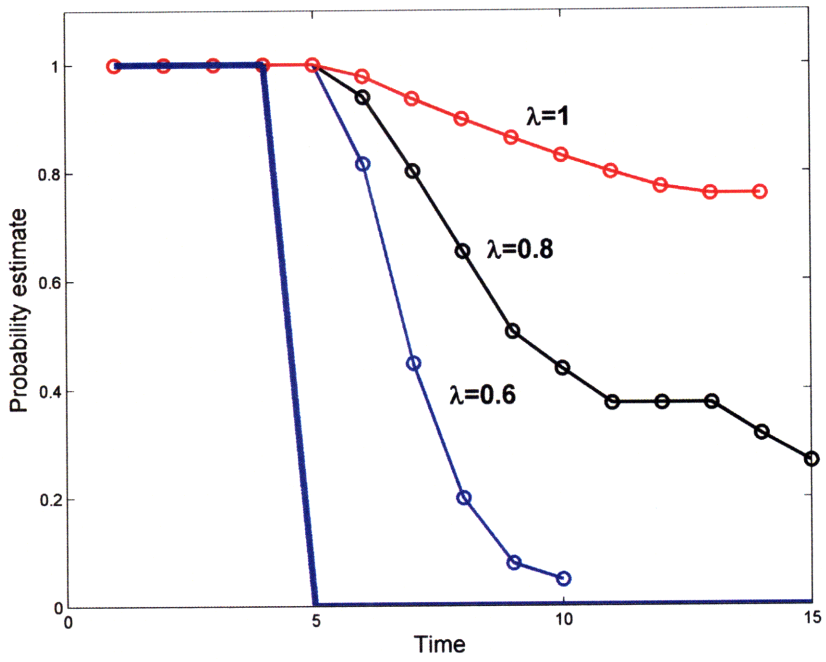


Fig. 5-4: Step response from $p_{nom} = 1$ to $p_{nom} = 0$ for three different values of λ, showing that $\lambda = 0.6$ has a response time of approximately 5 times steps, while $\lambda = 1$ has a very slow response time.

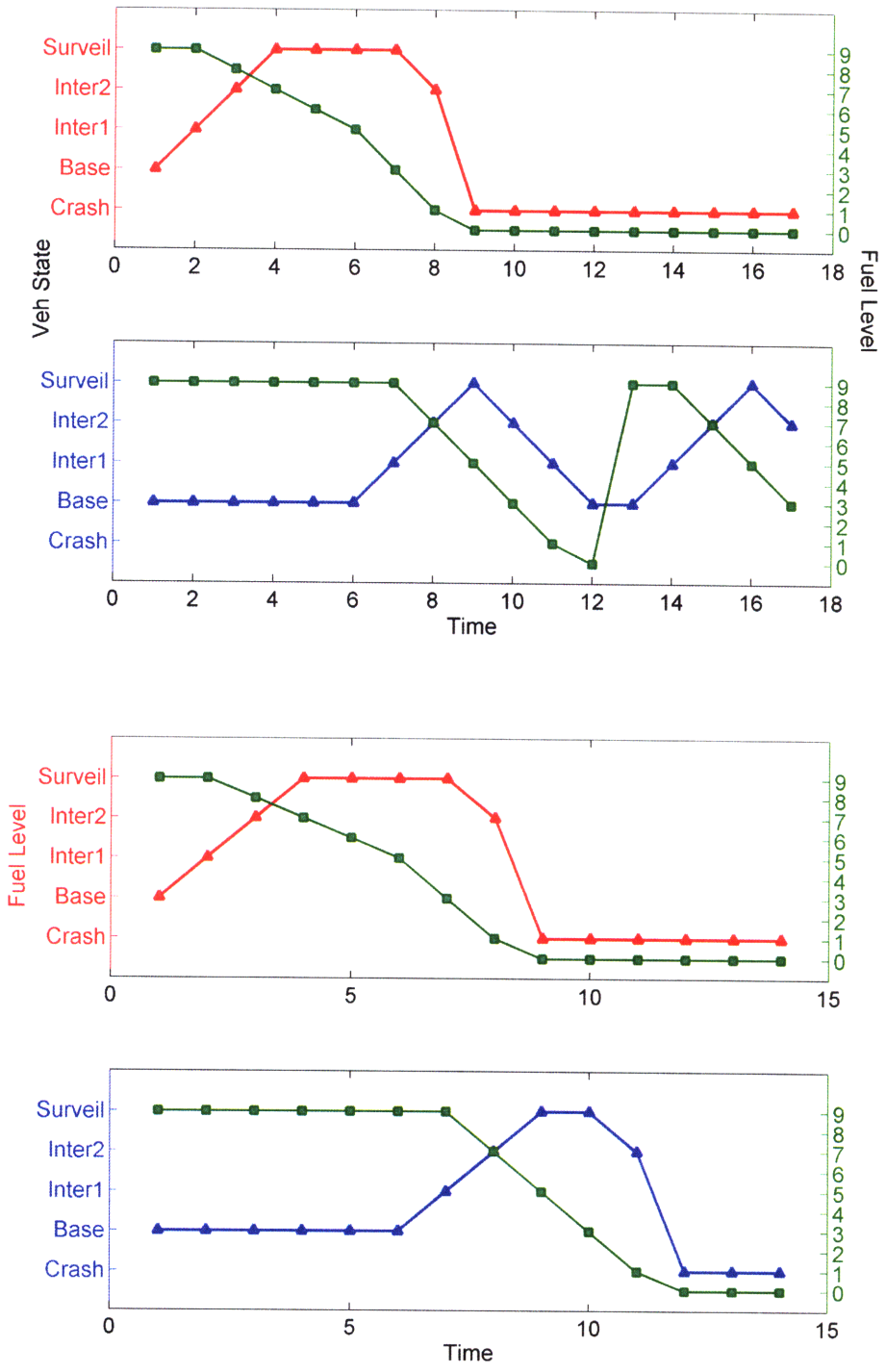


Fig. 5-5: For $\lambda = 0.8$ (top), the faster estimator response causes the second vehicle to only stay out for a single time step. For $\lambda = 1$ (bottom), the second vehicle stays on surveillance location for an extra time step, and that results in vehicle failure.

the true estimate after 10 time steps, while $\lambda = 0.6$ is within 20% after only 3 time steps, resulting in a significantly faster response. The variation of λ resulted in an interesting set of vehicle behaviors that can be seen in Figure 5-5. For $\lambda = 1$ (top), the estimate converges too slowly, resulting in an extremely slow convergence to the optimal policy. The convergence is so slow that both vehicles crash (vehicle 1 at time step 9, and vehicle 2 at time step 12), because the estimator was not capable of detecting the change in the value of p_{nom} quickly, and these vehicle were still operating under an optimistic value of $p_{nom} \approx 0.8$. Due to the physical dynamics of the fuel flow switch for this scenario, it turns out that the first vehicle will inevitably crash, since the switch occurs when the vehicle does not have sufficient fuel to return to base. However, if the estimator were responsive enough to detect the switch, this could results in a much decrease surveillance time for the second vehicle. This does not occur when $\lambda = 1$. The benefits of the more responsive estimator are seen in the bottom figure, where by selecting $\lambda = 0.8$, the second vehicle only spends one unit of time on surveillance, and then immediately returns to base to refuel, with only 1 unit of fuel remaining. Thus, the faster estimator is able to adapt in time to prevent the second vehicle from crashing.

5.6.2 Test 2

The final scenario was a slightly different test of the adaptation mechanism in tracking a series of smaller step changes to p_{nom}. In the earlier flight tests, under a nominal fuel flow, $p_{nom} = 1$, the fuel transitions were always of 1 unit of fuel. Likewise, when $p_{nom} = 0$, the fuel transitions were always of 2 units of fuel. In this test, the transition probability p_{nom} was decreased in steps of 0.3, and the estimators saw both nominal and off-nominal fuel transitions in the estimator updates at each time step (unlike the earlier tests where they *either* saw nominal transitions or off-nominal transitions). As a result, this test was perhaps a more realistic implementation of a gradual temporal degradation of vehicle health. Figure 5-6 is shown for two different choices of $\lambda = \{0.8, 1\}$. The first item of note is the step decreases in p_{nom}, that unlike the earlier flight results, are much more subtle. Next, note that the initial response of

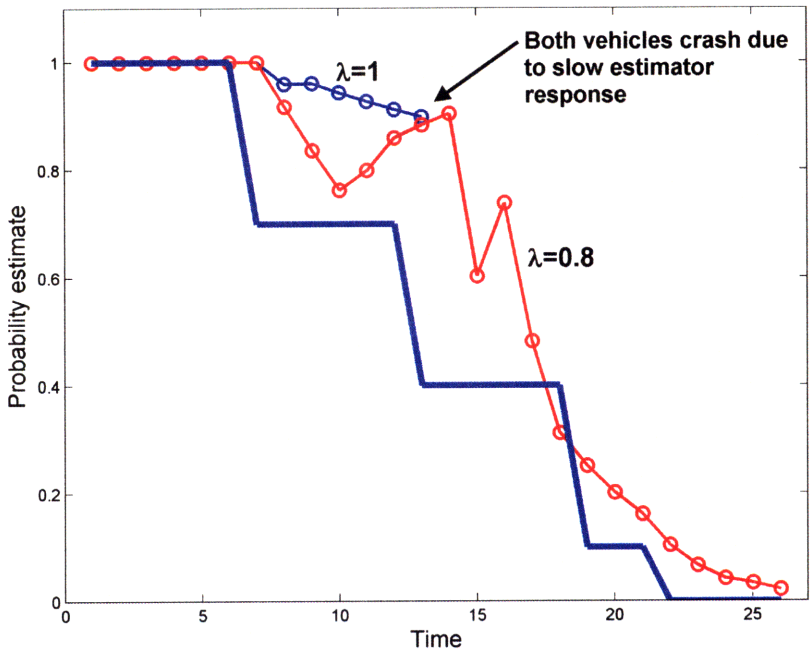
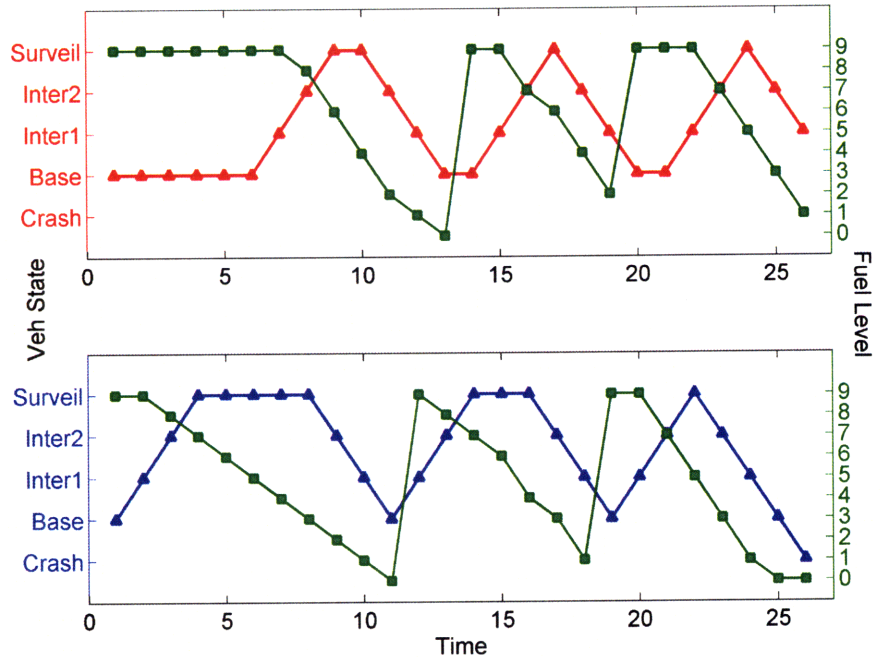


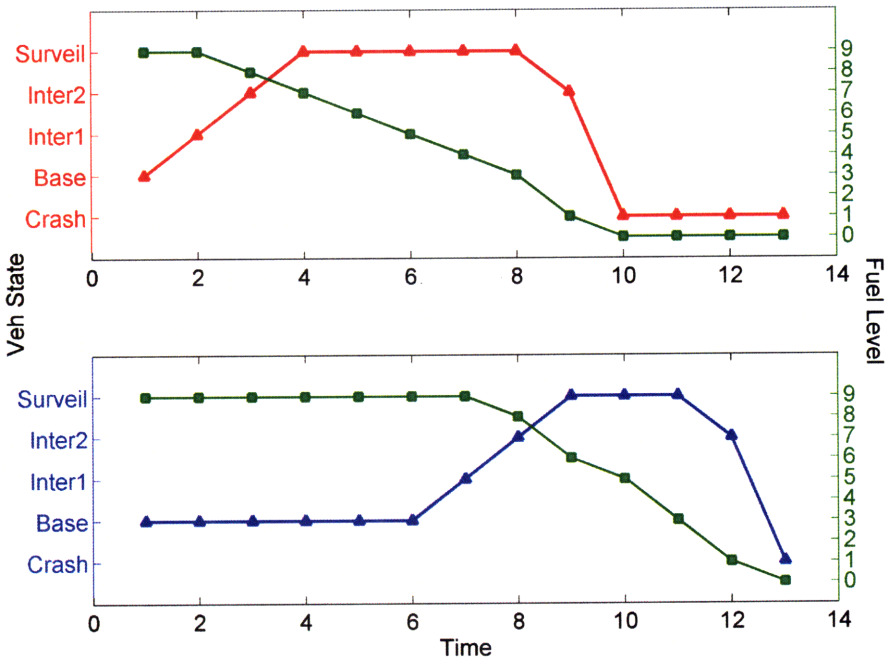
Fig. 5-6: Probability estimates of p_{nom} for $\lambda = 0.8$ and $\lambda = 1$. Due to the slow response of the $\lambda = 1$ estimator, both vehicles crash by time step 13, and no further adaptation is possible. Estimator with $\lambda = 0.8$ shows faster response, and ultimately converges to the true value.

the undiscounted estimator (blue) is extremely slow. In this flight test, the adaptation was so slow that the significant mismatch between the true and estimated system resulted in a mismatched policy that ultimately resulted in the loss of both vehicles.

Note that using an undiscounted estimator, both vehicles end up crashing in 13 time steps, while by using the discounted estimator, the vehicle lifetime is extended to 25 time steps, at which point one of the vehicle crashes due to the off-nominal sequence of transitions corresponding to the new $p_{nom} = 0$, which the estimator had not yet converged to. Further decreasing the λ parameter emphasizes the new observations significantly, and thus was less desirable venue to pursue. This observation served as an appropriate motivation for the next section, which accounted for vehicle failure by merging both robust and adaptive planning.



(a) $\lambda = 0.8$



(b) $\lambda = 1$

Fig. 5-7: The slower estimator $\lambda = 1$ (bottom) does not detect the fuel flow transition sufficiently quickly, causing both vehicles to run out of gas.

5.6.3 Further remarks

Remark 1 (Same model for all vehicles): These experiments assumed that the same transition model was used for all vehicles. This assumption is valid if the vehicles were in fact impacted by the same adversarial effects, but the estimators used in this paper can be applied to situations where individual vehicles have unique transition models.

Remark 2 (Information updates and cooperation): Since each vehicle was assumed to have the same transition model, the vehicles could update the models with their individual observations during the flights. Furthermore, if neither vehicle crashed, then twice the information was available to the estimators to update the probability estimates of $\hat{p}_{nom}$. This implies an indirect (but unintentional) cooperation among the vehicles for estimating this unknown, time-varying parameter.

5.7 Robust and Adaptive Replanning

In this section, the adaptive replanning was implemented by explicitly accounting for the residual uncertainty in the probability estimate $\hat{p}_{nom}$. Since this was a scalar estimation problem and the counts-based approach was used, at each time step the estimator output the updated $\alpha(k+1)$ and $\beta(k+1)$, and calculated the mean and variance as

$$\begin{aligned}\hat{p}_{nom} &= \frac{\alpha(k+1)}{(\alpha(k+1)+\beta(k+1)} \\ \sigma_p^2 &= \frac{\alpha(k+1)\beta(k+1)}{(\alpha(k+1)+\beta(k+1))^2\ (\alpha(k+1)+\beta(k+1)+1)}\end{aligned}$$

The Dirichlet Sigma Points were then formed using this mean and variance

$$\begin{aligned}\mathcal{Y}_0 &= \hat{p}_{nom} \\ \mathcal{Y}_1 &= \hat{p}_{nom} + \beta\ \sigma_p \\ \mathcal{Y}_2 &= \hat{p}_{nom} - \beta\ \sigma_p\end{aligned}$$

and used to find the robust policy. Using the results from the earlier chapters, appropriate choices of β could range from 1 to 5, where $\beta \approx 3$ corresponds to a 99% certainty region for the Dirichlet (in this case, the Beta density). For this scalar problem, the robust solution of the MDP corresponded to using a value of $\hat{p}_{nom} - \beta\sigma_p$ in place of the nominal probability estimate $\hat{p}_{nom}$, as this corresponded to a more cautious policy.

Flight experiments were repeated for a case when the transition probability estimate $\hat{p}_{nom}$ was varied in mid-mission, and compared three different replanning strategies

- **Adaptive only:** The first replan strategy involved only an adaptive strategy, with $\lambda = 0.8$, and using only the estimate $\hat{p}_{nom}$ (equivalent to setting $\beta = 0$ for the Dirichlet Sigma Points)

- **Robust replan, undiscounted adaptation:** This replan strategy used the undiscounted mean-variance estimator $\lambda = 1$, and set $\beta = 4$ for the Dirichlet Sigma Points

- **Robust replan, discounted adaptation:** This replan strategy used the undiscounted mean-variance estimator $\lambda = 0.8$, and set $\beta = 4$ for the Dirichlet Sigma Points

In all cases, the vehicle takes off from base, travels through 2 intermediate areas, and then reaches the surveillance location. In the nominal fuel flow setting losing 1 unit of fuel per time step, the vehicle can safely remain at the surveillance region for 4 time steps, but in the off-nominal fuel flow setting (losing 2 units), the vehicle can only remain on surveillance for only 1 time step.

The main results are shown in Figure 5-8, where the transition in p_{nom} occurred at $t = 7$ time steps. At this point in time, one of the vehicles is just completing the surveillance, and is initiating the return to base to refuel, as the second vehicle is heading to the surveillance area. The key to the successful mission, in the sense of avoiding vehicle crashes, is to ensure that the change is detected sufficiently quickly,

and that the planner maintains some level of cautiousness in this estimate by embedding robustness. The successful mission will detect this change rapidly, and leave the UAVs on target for a shorter time.

The result of Figure 5-8(a) ignores any uncertainty in the estimate but has a fast adaptation (since it uses the factor $\lambda = 0.8$). However, by not embedding the uncertainty, the estimator detects the change in p_{nom} quickly, but allocates the second vehicle to remain at the surveillance. Consequently, one of the vehicles runs out of fuel, and crashes. At the second cycle of the mission, the second vehicle remains at the surveillance area for only 1 time step.

The result of Figure 5-8(a) accounts for uncertainty in the estimate but has a slow adaptation (since it uses the factor $\lambda = 1$). However, while embedding the uncertainty, the replanning is not done quickly, and for this different reason from the adaptive, non-robust example, one of the vehicle runs out of fuel, and crashes. At the second cycle of the mission, the second vehicle remains at the surveillance area for only 1 time step.

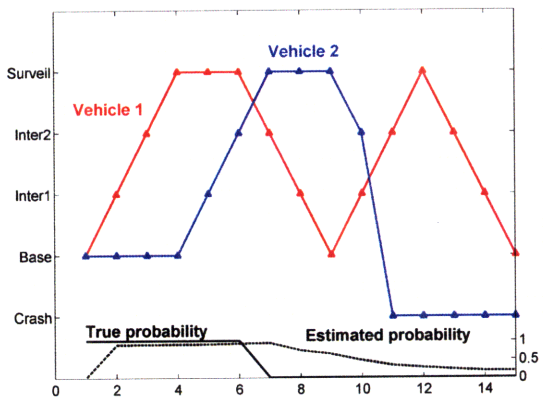
Figure 5-8(c) shows the robustness and adaptation acting together to cautiously allocate the vehicles, while responding quickly to changes in p_{nom}. The second vehicle is allocated to perform surveillance for only 2 time steps (instead of 3), and safely returns to base with no fuel remaining. At the second cycle, both vehicles only stay at the surveillance area for 1 time step. Hence, the robustness and adaptation have together been able to recover mission efficiency by bringing in their relative strengths: the robustness by accounting for uncertainty in the probability, and the adaptation by quickly responding to the changes in the probability.

5.8 Summary

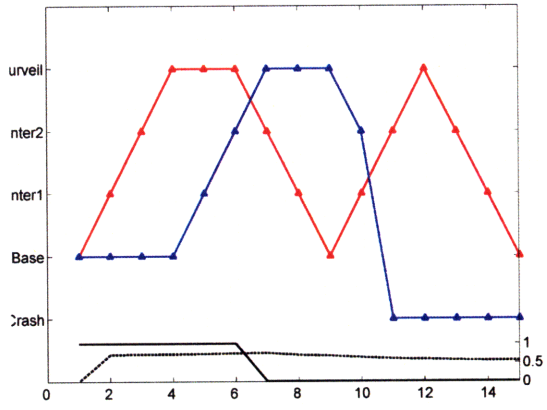
This chapter presented hardware implementation that demonstrate the detrimental impact of modeling mismatches, and show that the adaptation approach can mitigate these effects even in the presence of poorly known initial model and model changes. Furthermore, the adaptive approach yields better performance over offline, minimax

type approaches, which must trade-off performance versus robustness.

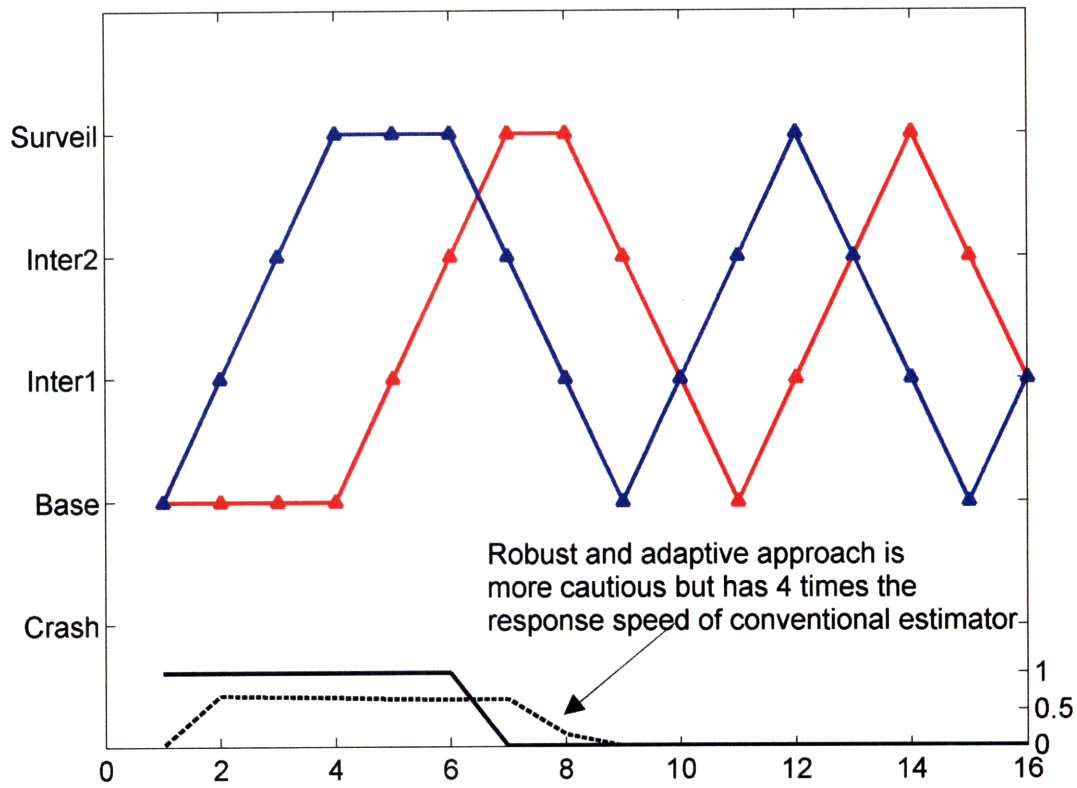
The flight experiments demonstrate the effectiveness of the adaptive architecture. With this architecture in place, there are a number of interesting future research areas that could be explored. First, in the flight experiments done to date, the same fuel model was assumed for all vehicles. A minor, but interesting modification would be to run a separate fuel model estimator for every vehicle, allowing for the possibility that vehicles degrade at different rates, for example. Another area would be modification of the system cost function to explicitly reward exploration, where vehicles would be rewarded for taking actions that reduce the uncertainty in the system parameters.



(a) Adaptive mission, with $\lambda = 0.8$



(b) Robust and adaptive mission, with $\lambda = 1$ and $\beta = 4$



(c) Robust and adaptive mission, with $\lambda = 0.8$ and $\beta = 4$

Fig. 5-8: Robust and adaptive flight test underscoring the importance of integrating the two technologies

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis has presented new contributions in the area of robust decision-making and robust estimation. In the area of robust decision-making, this thesis has presented:

- An algorithm that precisely defines the model uncertainty in terms of *credibility regions*, using the Dirichlet prior to model the uncertain transition probabilities. This bisection algorithm is used in conjunction with Monte Carlo sampling, and can efficiently find the credibility region used in the robust MDP;
- A new sampling-based algorithm using Dirichlet Sigma Points for finding approximate solutions to robust MDPs in a computationally tractable manner. We prove that the Dirichlet Sigma Points are proper samples of a probability vector (summing to unity, and between 0 and 1) and can therefore be used in general sampling-based algorithms. By using Dirichlet Sigma Points, we significantly reduce the total number of samples required to find the robust solution, while achieving near optimal performance;
- Guidelines for choosing the tuning parameter used in the Dirichlet Sigma Points, and numerical results demonstrating the reduction in samples required for the robust solution. In particular we show results in a machine repair problem, and autonomous agent planning.

In the area on multiple model estimation, this thesis has

- Addressed the issue of uncertain transition probabilities in multiple model estimators. In particular, we have extended the work of Refs. [27, 46] and identified the problem of covariance mismatch due to the uncertain Markov Chain;

- Provided a framework for generating robust estimates and covariances. In tracking applications, one of the main problems of covariance mismatch is the problem of covariance underestimation, in which the estimator is more confident about its state estimates than it should be, and can result in an increased estimation error. Our robustness framework ensures that the covariance is not underestimated, and is able to maintain a low estimation error;

- Shown reduction in estimation error in two aerospace tracking problems: the first one is a UAV multi-target tracking problem, and the second an agile target tracking problem.

Finally, the work on Markov Chain adaptation provided a method for learning the transition probabilities of the Markov Chain when these transition probabilities are time varying. In particular,

- An explicit recursion is derived for the mean and variance of the transition probabilities under a Dirichlet prior, making the Dirichlet Sigma Points amenable to real-time adaptation.

- This recursive formulation has been used to identify the cause of the slow learning of the Markov Chain, namely that the effective estimator gain drives to zero too quickly;

- A new estimator is derived that introduces the notion of an effective process noise to speed up the transition probability identification problem;

- Numerical examples are presented that demonstrate the faster adaptation of the transition probabilities using the new estimator. This new estimator is

also demonstrated in the context of real-time MDP re-planning where the optimal reward is collected almost twice as quickly as conventional adaptation algorithms.

This robust and adaptive group of algorithms has then been implemented in the RAVEN testbed. The benefits of using the proposed algorithms has been demonstrated in extended mission times with reduced vehicle crashes.

6.2 Future Work

There are many interesting venues for future work in this problem. At a high level, this thesis has presented new algorithms for applications to sequential decision-making applications that can cope with the uncertainty in the transition probabilities, and can efficiently adapt to changes in these transition probabilities. This of course draws a stronsg parallel to alternative learning techniques that do not explicitly use a model for the optimal control problem, such as Reinforcement Learning (RL) methods.

RL-like methods, for example, allow an agent to incorporate an "exploration" set of actions, where the agent actively tries actions that are not necessarily optimal ("exploitation" actions), but serve to find out more about the uncertain world. The methods proposed in Chapter 4 adapt passively only based on transitions that have been observed. Being able to extend the proposed model-based robust adaptive methods to account for an active uncertainty reduction mechanism such as those proposed in the RL literature would be an important extension of this work.

When dealing with multi-agent problems, an additional set of important research questions addresses the tradeoffs of decentralization and centralized decision-making. In particular, if all agents share the same transition model (as in the example using the RAVEN testbed), and each agent is experiencing a different set of transitions, how do the agents share their own information with other members of their group? With a much larger number of dispersed vehicles, possibly in a non-fully connected network, recent ideas from consensus for uncertain probabilities will also present interesting areas of future research [70].

Bibliography

[1] M. Alighanbari. *Robust and Decentralized Task Assignment Algorithms for UAVs.* PhD thesis, MIT, 2007. 17, 18

[2] M. Alighanbari, L. F. Bertuccelli, and J. P. How. Filter-Embedded UAV Task Assignment Algorithms For Dynamic Environments. *AIAA Conference on Guidance, Navigation and Control*, 2004. 17

[3] A. Bagnell, A. Ng, and J. Schneider. Solving Uncertain Markov Decision Processes. *NIPS*, 2001. 19, 24, 25, 26

[4] Y. Bar Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation.* Wiley Interscience, 2001. 80, 82, 83, 106, 121

[5] A. Barto, S. Bradtke, and S. Singh. Learning to Act using Real-Time Dynamic Programming. *Artificial Intelligence*, 72:81–138, 2001. 126, 133

[6] J. Bellingham, M. Tillerson, M. Alighanbari, and J. P. How. Cooperative Path Planning for Multiple UAVs in Dynamic and Uncertain Environments. *IEEE Conference on Decision and Control*, 2002. 17

[7] J. Bellingham, M. Tillerson, A. Richards, and J. P. How. *Multi-Task Allocation and Path Planning for Cooperative UAVs.* Kluwer Academic Publishers, 2003. 17

[8] R. Bellman. *Dynamic Programming.* Princeton University Press, 1957. 18, 19

[9] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer, 1985. 33

[10] D. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, 2005. 27, 56, 132

[11] L. F. Bertuccelli, M. Alighabari, and J. P. How. Robust Planning For Coupled Cooperative UAV Missions. *IEEE Conference on Decision and Control*, 2004. 17

[12] L. F. Bertuccelli and J. P. How. Robust Decision-Making for Uncertain Markov Decision Processes Using Sigma Point Sampling. *IEEE American Controls Conference*, 2008. 114

[13] B. Bethke, L. Bertuccelli, and J. P. How. Experimental Demonstration of MDP-Based Planning with Model Uncertainty. In *AIAA Guidance Navigation and Control Conference*, Aug 2008. AIAA-2008-6322. 139, 146

[14] B. Bethke, J. How, and J. Vian. Group Health Management of UAV Teams With Applications to Persistent Surveillance. *IEEE American Controls Conference*, 2008. 17, 18, 138, 139

[15] H. Blom and Y. Bar-Shalom. The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients. *IEEE Trans. on Automatic Control*, 33(8), 1988. 80

[16] P. Borkar and P. Varaiya. Adaptive Control of Markov Chains, I: Finite Parameter Set. *IEEE Trans. on Automatic Control*, AC–24(6), 1979. 114

[17] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte. Decentralized bayesian negotiation for cooperative search. *IEEE/RSJ International Conf. on Intelligent Robots and Systems*, 2004. 17

[18] J. Buckley and E. Eslami. Fuzzy markov chains: Uncertain probabilities. *Mathware and Soft Computing*, 9:33–41, 2002. 25

[19] M. H. Chen and Q. M. Shao. Monte Carlo Estimation of Bayesian Credible and HPD Intervals. *Journal of Computational and Graphical Statistics.*, 7:193–107, 1999. 33, 35, 36

[20] A. Chinchuluun, D. Grundels, and P. Pardalos. Searching for a Moving Target: Optimal Path Planning. *IEEE International Conference on Networking, Sensing and Control*, 2005. 17

[21] S. Coraluppi. *Optimal Control of Markov Decision Processes for Performance and Robustness.* PhD thesis, Univ of Maryland, 1997. 29, 30

[22] E. Craparo, E. Modiano, and J. How. Optimization of Mobile Backbone Networks: Improved Algorithms and Approximation. *IEEE American Controls Conference*, 2008. 18

[23] E. Delage and S. Mannor. Percentile Optimization for Markov Decision Processes with Parameter Uncertainty. *subm to Operations Research*, 2007. 25, 26

[24] A. S. Dif. *Advanced Matrix Theory for Scientist and Engineers.* Abacus Press, 1991. 108

[25] F. Doshi, J. Pineau, and N. Roy. Reinforcement Learning with Limited Reinforcement: Using Bayes Risk for Active Learning in POMDPs. *International Conference on Machine Learning*, 2008. 26, 32

[26] F. Doshi and N. Roy. Efficient Model Learning for Dialog Management. *Proc of Human Robot Interaction*, 2007. 26, 32

[27] A. Doucet and B. Ristic. Recursive State Estimation for Multiple Switching Models with Unknown Transition Probabilities. *IEEE Trans. on Aerospace and Electronic Systems*, 38(3):1098–1104, 2003. 21, 30, 80, 81, 94, 100, 158

[28] C.A. Earnest. Dynamic Action Space for Autonomous Search Operations. Master's thesis, MIT, 2005. 17

[29] G. S. Fishman. *A First Course in Monte Carlo.* Duxbury Advanced Series, 2006. 35

[30] J. Ford and J. Moore. Adaptive Estimation of HMM Transition Probabilities. *IEEE Transactions on Signal Processing*, 46(5), 1998. 115

[31] D. Fox. Adapting the Sample Size in Particle Filters Through KLD-Sampling. *International Journal of Robotics Research*, 22(12), 2003. 41

[32] A. Gelb. *Applied Optimal Estimation.* MIT Press, 1974. 90

[33] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis.* Chapman and Hall, 1995. 51, 88

[34] L. El Ghaoui and G. Calafiore. Robust filtering for discrete-time systems with structured uncertainty. *IEEE Trans. on Automatic Control*, 46(7), 2001. 90

[35] V. Gullapalli and A. Barto. Convergence of Indirect Adaptive Asynchronous Value Iteration Algorithms. *Advances in NIPS*, 1994. 126, 127

[36] A. Hero, D. Castanon, D. Cochran, and K. Kastella. *Foundations and Applications of Sensor Management.* Springer-Verlag, 2008. 80

[37] M. Hofbaur and B. Williams. Hybrid Estimation of Complex Systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 34(5), 2004. 80

[38] L. Hong, N. Cui, M. Bakich, and J.R. Layne. Multirate interacting multiple model particle filter for terrain-based ground target tracking. *IEEE Proc of Control Theory Applications*, 153(6), 2006. 80

[39] J. P. How, B. Bethke, A. Frank, D. Dale, and J. Vian. Real-time indoor autonomous vehicle test environment. *IEEE Control Systems Magazine*, 28(2):51–64, April 2008. 138

[40] I. Hwang, H. Balakrishnan, and C. J. Tomlin. Observability Criteria and Estimator Design for Stochastic Linear Hybrid Systems. *IEE European Control Conference*, 2003. 80

[41] I. Hwang, H. Balakrishnan, and C. J. Tomlin. Performance Analysis of Hybrid Estimation Algorithms. *IEEE Conference on Decision and Control*, 2003. 80

[42] R.B. Israel, J. S. Rosenthal, and J. Z. Wei. Finding Generators for Markov Chains via Empirical Transition Matrices with Applications to Credit Ratings. *Mathematical Finance*, 11(2), 2001. 25, 30

[43] G. Iyengar. Robust Dynamic Programming. *Math. Oper. Res.*, 30(2):257–280, 2005. 19, 24, 25, 26, 39, 40, 111

[44] R. Jaulmes, J. Pineau, and D. Precup. Active Learning in Partially Observable Markov Decision Processes. *European Conference on Machine Learning (ECML)*, 2005. 25, 26, 32, 114, 120

[45] R. Jaulmes, J. Pineau, and D. Precup. Learning in Non-Stationary Partially Observable Markov Decision Processes. *ECML Workshop on Reinforcement Learning in Non-Stationary Environments*, 2005. 25, 26, 32, 41, 47, 114

[46] V. Jilkov and X. Li. Online Bayesian Estimation of Transition Probabilities for Markovian Jump Systems. *IEEE Trans. on Signal Processing*, 52(6), 2004. 21, 30, 80, 81, 85, 94, 100, 114, 119, 158

[47] S. Julier and J. Uhlmann. Unscented Filtering and Nonlinear Estimation. *Proc. of IEEE*, 92(3), 2004. 42

[48] T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati, I. Kadar, B. Abrams, and E. Eadan. Tracking ground targets with road constraints using an IMM estimator. *IEEE Aerospace Conference*, 1998. 80

[49] V. Konda and J. Tsitsiklis. Linear stochastic approximation driven by slowly varying Markov chains. *Systems and Control Letters*, 50, 2003. 115

[50] V. Krishnamurthy and J. B. Moore. On-Line Estimation of Hidden Markov Model Parameters Based on the Kullback-Leibler Information Measure. *IEEE Trans on Signal Processing*, 41(8), 1993. 114, 116

[51] P. R. Kumar and A. Becker. A New Family of Optimal Adaptive Controllers for Markov Chains. *IEEE Trans. on Automatic Control*, AC-27(1), 1982. 24

[52] P. R. Kumar and W. Lin. Optimal Adaptive Controllers for Unknown Markov Chains. *IEEE Trans. on Automatic Control*, AC-27(4), 1982. 24

[53] P. R. Kumar and W. Lin. Simultaneous Identification and Adaptive Control of Unknown Systems over Finite Parameters Sets. *IEEE Trans. on Automatic Control*, AC-28(1), 1983. 24, 115

[54] Aerospace Controls Laboratory. UAV swarm health management project. Online `http://vertol.mit.edu`, 2008. 138

[55] B. Li and J. Si. Robust Dynamic Programming for Discounted Infinite-Horizon Markov Decision Processes with Uncertain Stationary Transition Matrices. *IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning*, 2007. 24, 25

[56] X.-R. Li and Y. Bar-Shalom. Multiple Model Estimation with Variable Structure. *IEEE Trans. on Automatic Control*, 41(4), 1996. 80

[57] X. R. Li and V. Jilkov. A survey of Maneuvering Target tracking. Part V: Multiple Model Methods. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4), 2005. 80

[58] Y. Li, S. Kang, and Y. Paschalidis. Some Results on the Analysis of Stochastic Processes with Uncertain Transition Probabilities and Robust Optimal Control. *Allerton Conf.*, 2008. 26

[59] S. Mannor, D. Simester, P. Sun, and J. Tsitsiklis. Bias and Variance Approximation in Value Function Estimates. *Management Science*, 52(2):308–322, 2007. 25, 26, 29, 39, 47, 114

[60] P. Marbach. *Simulation-based methods for Markov Decision Processes.* PhD thesis, MIT, 1998. 114

[61] S. Marcus, E. Fernandez-Gaucherand, D. Hernandez-Hernandez, S. Coraluppi, and P. Fard. Risk-sensitive Markov decision processes. *Systems and Control in the Twenty-First Century*, 1997. 29

[62] P. Maybeck. *Stochastic Estimation and Control.* Academic Press, 1970. 106

[63] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1), 1998. 80

[64] J. McGrew. Real-Time Maneuvering Decisions for Autonomous Air Combat. Master's thesis, MIT, 2008. 138

[65] T. Menke and P. Maybeck. Sensor/actuator failure detection in the vista F-16 by multiple model adaptive estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 31(4), 1995. 80

[66] R. W. Miller. Asymptotic behavior of the Kalman filter with exponential aging. *AIAA Journal*, 9, 1971. 121, 122

[67] M. Morelande, C. Kreucher, and K. Kastella. A Bayesian Approach to Multiple Target Detection and Tracking. *IEEE Transactions on Signal Processing*, 55(5), 2007. 80

[68] J. Nascimento, J. S. Marques, and J. Sanches. Estimation of cardiac phases in echographic images using multiple models. *Proc. of IEEE Conf on Image Processing*, 2, 2003. 80

[69] A. Nilim and L. El Ghaoui. Robust Solutions to Markov Decision Problems with Uncertain Transition Matrices. *Operations Research*, 53(5), 2005. 19, 24, 25, 26, 31, 39, 40, 41, 50, 54, 109, 114

[70] R. Olfati-Saber, E. Franco, E. Frazzoli, and J. S. Shamma. Belief Consensus and Distributed Hypothesis Testing in Sensor Networks. *Workshop of Network Embedded Sensing and Control*, October, 2005. 18, 159

[71] R. Olfati-Saber and R. M. Murray. Consensus Problems in Networks of Agents with Switching Topology and Time-Delays. *IEEE Trans. on Automatic Control*, 49(9), 2004. 18

[72] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, 1991. 32, 43

[73] A. Plotnik. *Applied Estimation for Hybrid Dynamical Systems Using Perceptional Information.* PhD thesis, Stanford, 2007. 80

[74] P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An Analytic Solution to Discrete Bayesian Reinforcement Learning. *Intl. Conf. on Machine Learning (ICML)*, 2006. 25

[75] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, 2005. 27

[76] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *IEEE Trans.*, 77(2), 1990. 20, 30, 114

[77] D. Ruan and D. Castanon. Real-time Tumor Tracking with Interactive Multiple Model Filter. *CENSSIS 2003*, 2003. 80

[78] S. Russel and P. Norvig. *Artificial Intelligence.* 1995. 63

[79] J. K. Satia and R. E. Lave. Markovian Decision Processes with Uncertain Transition Probabilities. *Operations Research*, 21(3), 1973. 24, 26

[80] M. Sato, K. Abe, and H. Takeda. Learning Control of Finite Markov Chains with Unknown Transition Probabilities. *IEEE Trans. on Automatic Control*, AC-27(2), 1982. 25, 115

[81] D. D. Sworder and J. E. Boyd. *Estimation Problems in Hybrid Systems.* Cambridge University Press, 1999. 80

[82] J. K. Tugnait. Adaptive Estimation and Identification for Discrete Systems with Markov Jump Parameters. *IEEE Trans. on Automatic Control*, AC-27(5), 1982. 81

[83] M. Valenti. *Approximate Dynamic Programming with Applications in Multi-Agent Systems.* PhD thesis, MIT, 2007. 17, 18, 138

[84] M. Valenti, B. Bethke, G. Fiore, J. P. How, and E. Feron. Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, August 2006. 138

[85] J. vanDorp and T. Mazzuchi. Solving for the parameters of a Beta Distribution under two quantile constraints. *Journal of Statistical Computation and Simulation*, 67, 2000. 52

[86] C. C. White and H. K. Eldeib. Markov Decision Processes with Imprecise Transition Probabilities. *Operations Research*, 42(4), 1994. 25, 26, 114

[87] N. A. White, P. S. Maybeck, and S. L. DeVilbiss. Detection of interference/jamming and spoofing in a DGPS-aided inertial system. *IEEE Transactions on Aerospace and Electronic Systems*, 34(4), 1998. 80

[88] N. Wicker, J. Mullera, R. Kiran, R. Kalathura, and O. Pocha. A maximum likelihood approximation method for Dirichlet's parameter estimation. *Computational Statistics and Data Analysis*, 52(3), 2008. 44, 78

[89] A. Willsky. A Survey of Design Methods for Failure Detection in Dynamic Systems. *Automatica*, 12, 1976. 80

[90] K. Zhou, J. Doyle, and K. Glover. *Essentials of Robust Control.* Prentice-Hall, 1997. 30