TECHNISCHE
UNIVERSITÄT
DARMSTADT

Statistical Machine Learning for Modeling and Control of Stochastic Structured Systems

Vom Fachbereich Informatik an der
Technischen Universität Darmstadt

zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Dissertation

von

Hany Abdulsamad, M.Sc.

Erstgutachter: Prof. Jan Peters, Ph.D.
Zweitgutachter: Prof. Thomas Schön, Ph.D.

Darmstadt, 2021

Abdulsamad, Hany — Statistical Machine Learning for Modeling
and Control of Stochastic Structured Systems
Darmstadt, Technische Universität Darmstadt
Jahr der Veröffentlichung der Dissertation auf TUprints: 2022
URN: urn:nbn:de:tuda-tuprints-225737
Tag der mündlichen Prüfung: 04.11.2021

Erklärungen laut Promotionsordnung

§8 Abs. 1 lit. c PromO

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

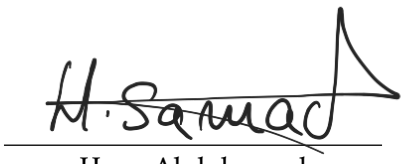
§9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 23. September 2021



Hany Abdulsamad

Abstract

Machine learning and its various applications have driven innovation in robotics, synthetic perception, and data analytics. The last decade especially has experienced an explosion in interest in the research and development of artificial intelligence with successful adoption and deployment in some domains. A significant force behind these advances has been an abundance of data and the evolution of simple computational models and tools with a capacity to scale up to massive learning automata. Monolithic neural networks with billions of parameters that rely on automatic differentiation are a prime example of the significant role efficient computation has had on supercharging the ability of well-established representations to extract intelligent patterns from unstructured data.

Nonetheless, despite the strides taken in the digital domains of vision and natural language processing, applications of optimal control and robotics significantly trail behind and have not been able to capitalize as much on the latest trends of machine learning. This discrepancy can be explained by the limited transferability of learning concepts that rely on full differentiability to the heavily structured physical and human interaction environments, not to mention the substantial cost of data generation on real physical systems. Therefore, these factors severely limit the application scope of loosely-structured over-parameterized data-crunching machines in the mechanical realm of robot learning and control.

This thesis investigates modeling paradigms of hierarchical and switching systems to tackle some of the previously highlighted issues. This research direction is motivated by insights into universal function approximation via local cooperating units and the promise of inherently regularized representations through explicit structural design. Moreover, we explore ideas from robust optimization that address model mismatch issues in statistical models and outline how related methods may be used to improve the tractability of state filtering in stochastic hybrid systems.

In Chapter 2, we consider hierarchical modeling for general regression problems. The presented approach is a generative probabilistic interpretation of local regression techniques that approximate nonlinear functions through a set of local linear or polynomial units. The number of available units is crucial in such models, as it directly balances representational power with the parametric complexity. This ambiguity is addressed by using principles from Bayesian nonparametrics to formulate flexible models that adapt their complexity to the data and can potentially encompass an infinite number of components. To learn these representations, we present two efficient variational inference techniques that scale well with data and highlight the advantages of hierarchical infinite local regression models, such as dealing with non-smooth functions, mitigating catastrophic forgetting, and enabling parameter sharing and fast predictions. Finally, we validate this approach on a set of large inverse dynamics datasets and test the learned models in real-world control scenarios.

Chapter 3 addresses discrete-continuous hybrid modeling and control for stochastic dynamical systems, which implies dealing with time-series data. In this scenario, we develop an automatic system identification technique that decomposes nonlinear systems into hybrid automata and leverages the resulting structure to learn switching feedback control via hierarchical reinforcement learning. In the process, we rely on an augmented closed-loop hidden Markov model architecture that captures time correlations over long horizons and provides a principled Bayesian inference framework for learning hybrid representations and filtering the hidden discrete states to apply control accordingly. Finally, we embed this structure explicitly into a novel hybrid relative entropy policy search algorithm that optimizes a set of local polynomial feedback controllers and value functions. We validate the overall switching-system perspective by benchmarking the open-loop predictive performance against popular black-box representations. We also provide qualitative empirical results for hybrid reinforcement learning on common nonlinear control tasks.

In Chapter 4, we attend to a general and fundamental problem in learning for control, namely robustness in data-driven stochastic optimization. The question of sensitivity has a strong priority, given the rising popularity of embedding statistical models into stochastic control frameworks. However, data from dynamical, especially mechanical, systems is often scarce due to a high extraction cost and limited coverage of the state-action space. The result is usually poor models with narrow validity and brittle control laws, particularly in an ill-posed over-parameterized learning example. We propose to robustify stochastic control by finding the worst-case distribution over the dynamics and optimizing a corresponding robust policy that minimizes the probability of catastrophic failures. We achieve this goal by formulating a two-stage iterative minimax optimization problem that finds the most pessimistic adversary in a trust region around a nominal model and uses it to optimize a robust optimal controller. We test this approach on a set of linear and nonlinear stochastic systems and supply empirical evidence of its practicality. Finally, we provide an outlook on how similar multi-stage distributional optimization techniques can be applied in approximate filtering of stochastic switching systems in order to tackle the issue of exponential explosion in state mixture components.

In summation, the individual contributions of this thesis are a collection of interconnected principles for structured and robust learning for control. Although many challenges remain ahead, this research lays a foundation for reflecting on future structured learning questions that strive to combine optimal control and statistical machine learning perspectives for the automatic decomposition and optimization of hierarchical models.

Kurzfassung

Maschinelles Lernen und seine verschiedenen Anwendungen haben Innovationen in der Robotik, der synthetischen Wahrnehmung und der Datenanalyse vorangetrieben. Vor allem in den letzten zehn Jahren ist das Interesse an der Erforschung und Entwicklung künstlicher Intelligenz explosionsartig gestiegen, und in einigen Bereichen wurden sie bereits erfolgreich eingeführt und eingesetzt. Eine wichtige Triebkraft hinter diesen Fortschritten war die Fülle an Daten und die Entwicklung einfacher Berechnungsmodelle und Werkzeuge, die bis zu massiven Lernautomaten skaliert werden können. Monolithische neuronale Netze mit Milliarden von Parametern, die auf automatischer Differenzierung beruhen, sind ein Paradebeispiel für die bedeutende Rolle, die effiziente Berechnungen bei der Verbesserung der Fähigkeit etablierter Darstellungen zur Extraktion intelligenter Muster aus unstrukturierten Daten gespielt haben.

Trotz der Fortschritte, die in den digitalen Bereichen der Bildverarbeitung und der Verarbeitung natürlicher Sprache gemacht wurden, hinken Anwendungen der optimalen Steuerung und der Robotik deutlich hinterher und waren nicht in der Lage, von den neuesten Trends des maschinellen Lernens in gleichem Maße zu profitieren. Diese Diskrepanz lässt sich durch die begrenzte Übertragbarkeit von Lernkonzepten, die auf vollständiger Differenzierbarkeit beruhen, auf stark strukturierte physische und menschliche Interaktionsumgebungen erklären, ganz zu schweigen von den erheblichen Kosten der Datengenerierung bei realen physikalischen Systemen. Diese Faktoren schränken daher den Anwendungsbereich von unstrukturierten, überparametrisierten Datenverarbeitungsmaschinen im mechanischen Bereich des Roboterlernens und der Robotersteuerung stark ein.

In dieser Arbeit werden Modellierungsparadigmen für hierarchische und schaltende Systeme untersucht, um einige der zuvor hervorgehobenen Probleme zu lösen. Diese Forschungsrichtung ist motiviert durch die Erkenntnisse der universellen Funktionsapproximation über lokal-kooperierende Einheiten und das Versprechen regularisierter Repräsentationen durch explizites Strukturdesign. Darüber hinaus erforschen wir Ideen aus der robusten Optimierung, die sich mit Problemen der Modellabweichung in statistischen Modellen befassen, und skizzieren, wie verwandte Methoden eingesetzt werden können, um die Traktabilität von Filterung in stochastischen Hybridsystemen zu verbessern.

In Kapitel 2 betrachten wir die hierarchische Modellierung für allgemeine Regressionsprobleme. Der vorgestellte Ansatz ist eine generative probabilistische Interpretation lokaler Regressionstechniken, die nichtlineare Funktionen durch einen Satz lokaler linearer oder polynomialer Einheiten approximieren. Die Anzahl der verfügbaren Einheiten ist bei solchen Modellen von entscheidender Bedeutung, da sie ein direktes Gleichgewicht zwischen der Repräsentationsfähigkeit und der parametrischen Komplexität herstellt. Diese Ambiguität wird durch die Anwendung von Prinzipien aus der Bayes'schen Nichtparametrik

angegangen, um flexible Modelle zu formulieren, die ihre Komplexität an die Daten anpassen und potenziell eine unendliche Anzahl von Komponenten umfassen können. Um diese Repräsentationen zu erlernen, stellen wir zwei effiziente Variationsinferenztechniken vor, die gut mit den Daten skalieren und die Vorteile hierarchischer lokaler Regressionsmodelle hervorheben, wie z.B. den Umgang mit nicht-kontinuierlichen Funktionen, die Abschwächung katastrophalen Vergessens und die Ermöglichung von Paramaterteilung und schnellen Vorhersagen. Schließlich validieren wir diesen Ansatz auf große Datensätze der inversen Dynamik und testen die gelernten Modelle in realen Kontrollszenarien.

Kapitel 3 befasst sich mit der diskret-kontinuierlichen hybriden Modellierung und Steuerung stochastischer dynamischer Systeme, was den Umgang mit Zeitreihendaten voraussetzt. In diesem Szenario entwickeln wir eine automatische Systemidentifikationstechnik, die nichtlineare Systeme in hybride Automaten zerlegt, und nutzen die resultierende Struktur, um eine schaltende Rückkopplungssteuerung über hierarchisches Bestärkungslernen zu erlernen. Dabei stützen wir uns auf eine erweiterte Markov-Modell-Architektur für geschlossene Regelkreise, die Zeitkorrelationen über lange Horizonte erfasst und einen grundlegenden Bayes'schen Inferenzrahmen für das Lernen hybrider Repräsentationen und die Filterung der verborgenen diskreten Zustände bietet, um die Steuerung entsprechend anzuwenden. Schließlich betten wir diese Struktur in einen neuartigen hybriden Suchalgorithmus mit relativer Entropie ein, der eine Reihe von lokalen polynomialen Rückkopplungsreglern und Wertfunktionen optimiert. Wir validieren den Gesamtansatz des Schaltsystems, indem wir die Vorhersageleistung mit gängigen Black-Box-Darstellungen vergleichen. Wir liefern auch qualitative empirische Ergebnisse für hybrides Bestärkungslernen bei gängigen nichtlinearen Steuerungsaufgaben.

In Kapitel 4 widmen wir uns einem allgemeinen und grundlegenden Problem des Lernens für die Steuerung, nämlich der Robustheit bei datengesteuerter stochastischer Optimierung. Die Frage der Sensitivität hat angesichts der zunehmenden Popularität der Einbettung statistischer Modelle in stochastische Kontrollsysteme hohe Priorität. Allerdings sind die Daten dynamischer, insbesondere mechanischer Systeme aufgrund der hohen Erhebungskosten und der begrenzten Abdeckung des Zustands-Aktions-Raums oft knapp. Das Ergebnis sind in der Regel schlechte Modelle mit enger Gültigkeit und brüchigen Kontrollgesetzen, insbesondere in einem schlecht gestellten, überparametrisierten Lernbeispiel. Wir schlagen vor, die stochastische Steuerung zu robustifizieren, indem wir die schlimmstmögliche Verteilung über die Dynamik finden und eine entsprechende robuste Strategie optimieren, die die Wahrscheinlichkeit von katastrophalen Fehlern minimiert. Wir erreichen dieses Ziel durch die Formulierung eines zweistufigen iterativen Minimax-Optimierungsproblems, das den pessimistischsten Gegner in einer Trust-Region um ein nominales Modell findet und zur Optimierung eines robusten optimalen Reglers verwendet. Wir testen diesen Ansatz an einer Reihe von linearen und nichtlinearen stochastischen Systemen und liefern empirische Beweise für seine Praxistauglichkeit. Schließlich geben

wir einen Ausblick darauf, wie ähnliche mehrstufige Optimierungstechniken bei der approximativen Filterung stochastischer Schaltsysteme angewendet werden können, um das Problem der exponentiellen Explosion von Zustandsmischungskomponenten zu lösen.

Zusammenfassend stellen die einzelnen Beiträge dieser Arbeit eine Sammlung von miteinander verbundenen Prinzipien für strukturiertes und robustes Lernen dar. Auch wenn noch viele Herausforderungen zu bewältigen sind, legt diese Arbeit den Grundstein, um über zukünftige Fragen des strukturierten Lernens nachzudenken, die darauf abzielen, die Perspektiven der optimalen Steuerung und des statistischen maschinellen Lernens für die automatische Dekomposition und Optimierung hierarchischer Modelle zu kombinieren.

Acknowledgments

I want to start by thanking my supervisor Jan Peters for his guidance and support, stretching from my undergrad studies and continuing throughout my Ph.D. years. I am grateful to Jan for sharing his time and insights, believing in me, and giving me the freedom to pursue the topics I was interested in. His mentoring and advice made me the researcher I am today. I will always cherish our regular meetings.

I have spent great years in the Intelligent Autonomous Systems Group, and I am thankful to every researcher, staff, and student member that contributed to that wonderful environment while I was there. I am especially grateful to Boris Belousov and Joe Watson for their friendship and for the endless and exciting scientific discussions that expanded my knowledge and understanding. To both Boris and Joe, I owe you a debt of gratitude.

I want to thank all my co-authors for their time, effort, and willingness to collaborate. I am grateful to Oleg Arenz, Riad Akrour, Christian Daniel, Abbas Abdolmaleki, Kianoosh Naveh, Samuele Tosatto, Joao Carvalho, Carlo D'Eramo, Joni Pajarinen, Jia-Jie Zhu, Debora Clever, Gerhard Neumann, and Rolf Findeisen. I am humbled by their dedication.

I am deeply thankful for the time and experience I gathered while interacting with students. I am honored that, over the years, many of them have become colleagues and co-authors of mine. I want to express my deep gratitude to Onur Celik, Pascal Klink, Mathias Schultheis, Peter Nickl, Tim Dorau, Kay Hansel, Janosch Moos, and Tim Schneider. Their contributions to my research cannot be overstated. I hope I have been a positive force for them on their paths as they have been for me on mine.

I would also like to thank Michael Lutter and Fabio Muratore for their friendship and companionship and for making the group a great place to be. I am also indebted to Rudolf Lioutikov, Gregor Gebhardt, Filipe Veiga, Simone Parisi, and Alexandros Paraschos for their help when I first joined the group and for the friendship that has since endured.

To Svenja, Leonie, and Tom, thank you from the bottom of my heart for your love, kindness, and support. Your friendship has made my years in Darmstadt unforgettable.

Finally, to my family, my mom and dad, my brother and sister, thank you, because of your support and patience, I am able to write these words.

Contents

Chapter 1
Introduction

1.1 Motivation

The inception of the core ideas discussed in this thesis occurred at the beginning of my doctoral study in 2016. During that time, deep learning approaches have already established an inevitable dominance in the machine learning community and have upended many years of hand-engineered solutions. However, the fields of optimal control and reinforcement learning were noticeably slower to adopt the same set of tools due to the unique conditions that apply in those areas and that set them apart from problems of visual perception, natural language understanding, and abstract large-data applications.

In our opinion, the most important distinction that separates learning for control from other data-driven machine intelligence domains is the data generation process. Typical supervised and unsupervised learning commonly rely on stationary datasets that can be aggregated from different sources, standardized, and benchmarked across algorithmic and model design choices. In contrast, optimizing intelligent systems in dynamic environments involves an interactive data generation process, posing problems of optimality and efficiency, inadvertently entangling data acquisition and learning.

This difference has significant implications for the general learning process and the role of poorly regularized over-parameterized representations. The nature of learning in physically interactive systems implies slow and expensive acquisition mechanisms that result in individual data distributions for every agent-environment combination, thus drastically limiting the possibility of aggregating large datasets. Moreover, the iterative accumulation and evolution of information as agents progress in learning leads to inherently non-stationary, distributionally shifting data streams. These aspects are serious challenges to large-parameter discriminative learning machines that commonly suffer from catastrophic forgetting, i.e., neural networks.

Regardless, deep representations made their way into reinforcement learning and optimal control and delivered outstanding never-before-observed results for a while. Neural networks became standard models for value and policy functions, and many algorithms were designed to accommodate them. However, with time, the limitations became evident. Developing reinforcement learning algorithms often degraded into an endeavor of endless tuning of gradient step sizes, random seeds, and implementations in order to compensate for scarce data and covariate shifts, despite relying on information-theoretic paradigms that were supposedly designed to deal with these problems automatically.

Our observation and understanding of these trends are the primary motivation for formulating a different research plan focusing on structured models. In this thesis, we seize the opportunity to study alternative representations that build up complexity from simple units without abstracting away or hiding their structure. We argue for a parametric regularization paradigm, *regularization through simplicity*. We adopt a switching-system view of control to highlight the redundancies of over-parameterized representations and present a powerful framework for dealing with discrete components of physical systems. In addition, we see potential in Bayesian nonparametrics, as it provides principled approaches to constructing flexible models that adapt according to the data. Finally, the Bayesian view allows for generative modeling that can deal with non-stationary information flows and offers flexibility for intelligent systems with continual learning objectives.

1.2 Contributions

In this section, we summarize the main chapters and state their primary contributions.

1.2.1 Infinite Bayesian Local Regression Mixtures

In Chapter 2, we look at hierarchical modeling for general regression problems. Using Bayesian nonparametric principles, we develop flexible models that approximate nonlinear functions using a set of local linear or polynomial units and adapt their complexity according to the data. These formulations are probabilistic generative interpretations of a wide range of local regression algorithms. We derive variational Bayes algorithms for efficient deterministic inference of these representations and highlight their properties on a set of toy examples and robotic control scenarios.

1.2.2 Hybrid Reinforcement Learning for Switching Systems

Chapter 3 focuses on the paradigm of hybrid systems for modeling and control. We propose using an augmented hidden Markov model to enable the automatic decomposition of general nonlinear dynamics by relying on Bayesian inference principles. Moreover, we leverage the resulting discrete-continuous structure in learning switching feedback control via a novel hybrid relative entropy policy search algorithm. Finally, we provide empirical validation on a set of popular examples of dynamical systems and highlight how this structured control view helps to drastically reduce parametric complexity.

1.2.3 Distributionally Robust Optimal Control

In Chapter 4, we address the problem of robustness of data-driven control with respect to statistical dynamics models. We propose an alternating minimax optimization problem that identifies the worst-case dynamics within a trust region and optimizes a conservative policy that lowers the risk of catastrophic failures. We provide empirical evaluation on probabilistic linear and nonlinear systems, emphasizing the advantages of robustification. Moreover, we outline how multi-stage distributional optimization strategies might

be used in approximate filtering of stochastic switching systems to address the problem of exponential explosion in state mixture components.

1.3 Foundations

The approaches presented in this thesis heavily build upon various central concepts in today's machine learning, reinforcement learning, and stochastic optimal control. In this section, we briefly introduce the foundations of conjugate Bayesian computation in exponential family distributions, expectation-maximization and variational inference of structured models, trust region policy search and stochastic optimal control formulations, and distributionally robust optimization.

1.3.1 Exponential Family

The upcoming chapters mainly consider random variables with probability density functions belonging to the exponential family. The unified minimal parameterization of this class of distributions lends itself to convenient and efficient posterior computation when paired with conjugate priors.

We assume the natural form for a probability density of a random variable $\mathbf{x}$

$$f(\mathbf{x}|\boldsymbol{\eta}) = h(\mathbf{x}) \exp\big[\boldsymbol{\eta} \cdot \mathbf{t}(\mathbf{x}) - a(\boldsymbol{\eta})\big],$$

where $h(\mathbf{x})$ is the base measure, $\boldsymbol{\eta}$ are the natural parameters, $\mathbf{t}(\mathbf{x})$ are the sufficient statistics and $a(\boldsymbol{\eta})$ is the log-partition function, or log-normalizer. Following the same notation, a conjugate prior $g(\boldsymbol{\eta}|\boldsymbol{\lambda})$ to the likelihood $f(\mathbf{x}|\boldsymbol{\eta})$ has the form

$$g(\boldsymbol{\eta}|\boldsymbol{\lambda}) = h(\boldsymbol{\eta}) \exp\big[\boldsymbol{\lambda} \cdot \mathbf{t}(\boldsymbol{\eta}) - a(\boldsymbol{\lambda})\big],$$

with prior sufficient statistics $\mathbf{t}(\boldsymbol{\eta}) = [\boldsymbol{\eta}, -a(\boldsymbol{\eta})]^\top$ and hyperparameters $\boldsymbol{\lambda} = [\boldsymbol{\alpha}, \boldsymbol{\beta}]^\top$. By applying Bayes' rule, we can directly infer the posterior $q(\boldsymbol{\eta}|\mathbf{x})$

$$\begin{aligned} q(\boldsymbol{\eta}|\mathbf{x}) &\propto f(\mathbf{x}|\boldsymbol{\eta}) g(\boldsymbol{\eta}|\boldsymbol{\lambda}) \\ &\propto \exp\big[\boldsymbol{\rho}(\mathbf{x}, \boldsymbol{\lambda}) \cdot \mathbf{t}(\boldsymbol{\eta}) - a(\boldsymbol{\rho})\big], \end{aligned}$$

where the posterior natural parameters $\boldsymbol{\rho}(\mathbf{x}, \boldsymbol{\lambda})$ are a function of the likelihood sufficient statistics $\mathbf{t}(\mathbf{x})$ and prior hyperparameters $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$

$$\boldsymbol{\rho}(\mathbf{x}, \boldsymbol{\lambda}) = \big[\boldsymbol{\alpha} + \mathbf{t}(\mathbf{x}),\ \boldsymbol{\beta} + \mathbf{1}\big]^\top.$$

The structure of the resulting posterior reveals a simple recipe for data-driven inference. By moving into the natural space, the posterior parameters are computed by combining the prior hyperparameters with the likelihood sufficient statistics and log-partition function.

By definition, every exponential family distribution has a minimal natural parameterization that leads to a unique decomposition of these quantities (Wainwright & Jordan, 2008).

1.3.2 Expectation-Maximization

The study of switching systems directly implies a mixture modeling paradigm that admits a set of hidden one-hot indicator states $\mathbf{z}$, which define a structure over the observed data $\mathbf{x}$. In statistical machine learning, the inference of such models is often tackled using expectation-maximization (EM) style algorithms (Dempster et al., 1977).

The objective in EM is to infer the parameters $\boldsymbol{\theta}$ of a mixture model

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}),$$

by maximizing the log-likelihood of a dataset consisting of N independent and identically distributed (i.i.d.) observations $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{maximize}} \quad \log p(\mathbf{X}|\boldsymbol{\theta}) &= \log \prod_{N} p(\mathbf{x}_n|\boldsymbol{\theta}) \\ &= \sum_{N} \log \sum_{\mathbf{z}} p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}). \end{aligned}$$

The log-sum operator in this optimization is hard to deal with because its maximization would require the consideration of all possible combinations of $\mathbf{z}$ for every $\mathbf{x}$. This problem can be side-stepped by introducing the variational distributions $q_n(\mathbf{z}_n)$ and using Jensen's inequality (Jensen, 1906) to optimize a lower bound of the log-likelihood instead

$$\begin{aligned} \sum_{N} \log \sum_{\mathbf{z}} p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}) &= \sum_{N} \log \sum_{\mathbf{z}} q_n(\mathbf{z}_n) \frac{p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta})}{q_n(\mathbf{z}_n)} \\ &\geq \sum_{N} \sum_{\mathbf{z}} q_n(\mathbf{z}_n) \log \frac{p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta})}{q_n(\mathbf{z}_n)}. \end{aligned} \tag{1.1}$$

By decomposing Equation (1.1), the lower bound can be reformulated

$$\begin{aligned} \log \prod_{N} p(\mathbf{x}_n|\boldsymbol{\theta}) &\geq \sum_{N} \sum_{\mathbf{z}} q_n(\mathbf{z}_n) \log \frac{p(\mathbf{x}_n|\boldsymbol{\theta}) p(\mathbf{z}_n|\mathbf{x}_n, \boldsymbol{\theta})}{q_n(\mathbf{z}_n)} \\ &= \sum_{N} \sum_{\mathbf{z}} q_n(\mathbf{z}_n) \log p(\mathbf{x}_n|\boldsymbol{\theta}) + \sum_{N} \sum_{\mathbf{z}} q_n(\mathbf{z}_n) \log \frac{p(\mathbf{z}_n|\mathbf{x}_n, \boldsymbol{\theta})}{q_n(\mathbf{z}_n)} \\ &= \sum_{N} \log p(\mathbf{x}_n|\boldsymbol{\theta}) - \sum_{N} \mathrm{KL}(q_n(\mathbf{z}_n) \,||\, p(\mathbf{z}_n|\mathbf{x}_n, \boldsymbol{\theta})). \end{aligned} \tag{1.2}$$

The Kullback-Leibler divergence (KL), in this instance an I-projection minimizing distance measure, is greater than or equal to zero, meaning that Equation (1.2) is always smaller than or equal to the log-likelihood $\sum_N \log p(\mathbf{x}_n|\boldsymbol{\theta})$. This lower bound is tight if variational distribution $q_n(\mathbf{z}_n)$ is chosen to have the same functional form as the true posterior $p(\mathbf{z}_n|\mathbf{x}_n, \boldsymbol{\theta})$ (Beal, 2003).

The general scheme of an expectation-maximization algorithm is to alternate between two procedures until convergence. In the expectation step, the KL divergence in Equation (1.2) is maximized with respect to the variational distributions $q_n(\mathbf{z}_n)$ by relying on an intermediate estimate of the parameters $\hat{\boldsymbol{\theta}}$

$$q_n(\mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{x}_n, \hat{\boldsymbol{\theta}}),$$

while the maximization step optimizes the lower bound in Equation (1.1) with respect to parameters $\boldsymbol{\theta}$ after substituting the intermediate solution of the expectation step

$$\begin{aligned} \log \prod^{N} p(\mathbf{x}_n|\boldsymbol{\theta}) &\geq \sum_N \sum_{\mathbf{z}} q_n(\mathbf{z}_n) \log \frac{p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta})}{q_n(\mathbf{z}_n)} \\ &= \sum_N \sum_{\mathbf{z}} p(\mathbf{z}_n|\mathbf{x}_n, \hat{\boldsymbol{\theta}}) \log p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}) + \text{const} \\ &= Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}). \end{aligned} \tag{1.3}$$

The term in Equation (1.3) is the expected complete-data log-likelihood. Notice that the sum over $\mathbf{z}$ is now outside the logarithm function. This new form makes it easy to plug in the mixture densities and maximize $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$ with respect to $\boldsymbol{\theta}$.

1.3.3 Variational Inference

Expectation-maximization techniques optimize a point estimate of the parameters $\boldsymbol{\theta}$. Unfortunately, these estimates are often only local optima considering the non-convex nature of the objective. In scenarios involving many states $\mathbf{z}$ and high dimensional parameters $\boldsymbol{\theta}$, these approaches often get stuck in shallow local minima that reflect low-quality solutions. One way to address this serious drawback is to consider a probabilistic paradigm to infer a posterior distribution over $\boldsymbol{\theta}$ and avoid catastrophic point estimates.

Markov chain Monte Carlo (MCMC) (Brooks et al., 2011) and variational inference (VI) (Blei et al., 2017) have become the two main approaches for approximate probabilistic inference in structured graphical models. While MCMC relies on constructing a stochastic sampling process that converges to the posterior, VI formulates the inference task as a deterministic optimization problem. Although VI usually relies on coarser functional posterior approximations, nonetheless, it is often preferred as it admits a clear convergence indicator. Moreover, deterministic optimization circumvents the issue of label switching

in sampled-based multi-modal posterior inference of mixture models.

In a nutshell, in variational inference, a typically intractable posterior is approximated by a tractable functional distribution $q(\boldsymbol{\beta})$ that minimizes the KL to true posterior $p(\boldsymbol{\beta}|\mathcal{D})$

$$q^*(\boldsymbol{\beta}) = \arg\min \quad \mathrm{KL}(q(\boldsymbol{\beta}) \,||\, p(\boldsymbol{\beta}|\mathcal{D})), \tag{1.4}$$

where $\mathcal{D}$ is observed data and the vector $\boldsymbol{\beta}$ subsumes both the parameters $\boldsymbol{\theta}$ and hidden indicators $\mathbf{z}$ in structured models. Note that the KL in Equation (1.4) is mode-seeking, meaning it will lock on one mode of a possibly multi-modal posterior. In general, the posterior $p(\boldsymbol{\beta}|\mathcal{D})$ is unknown, because the normalizer $p(\mathcal{D})$ is not tractable. In consequence, the KL cannot be minimized directly, but rather optimized via a related objective that is equal up to the constant term equivalent to the evidence $p(\mathcal{D})$

$$\begin{aligned}\mathrm{KL}(q(\boldsymbol{\beta}) \,||\, p(\boldsymbol{\beta}|\mathcal{D})) &= \mathbb{E}\big[\log q(\boldsymbol{\beta})\big] - \mathbb{E}\big[\log p(\boldsymbol{\beta}|\mathcal{D})\big] \\ &= \mathbb{E}\big[\log q(\boldsymbol{\beta})\big] - \mathbb{E}\big[\log p(\mathcal{D}, \boldsymbol{\beta})\big] + \text{const.}\end{aligned}$$

This modified objective is denoted as the negative evidence lower bound (ELBO) and can be reformulated to take the traditional form in VI algorithms

$$\begin{aligned}\mathrm{ELBO}(q) &= \mathbb{E}\big[\log p(\boldsymbol{\beta})\big] + \mathbb{E}\big[\log p(\mathcal{D}|\boldsymbol{\beta})\big] - \mathbb{E}\big[\log q(\boldsymbol{\beta})\big] \\ &= \mathbb{E}\big[\log p(\mathcal{D}|\boldsymbol{\beta})\big] - \mathrm{KL}(q(\boldsymbol{\beta}) \,||\, p(\boldsymbol{\beta})).\end{aligned} \tag{1.5}$$

The choice of the approximate posterior distribution $q(\boldsymbol{\beta})$ is open. In this paper, we focus on variational Bayes methods that rely on the (structured) mean-field assumption as a general recipe for maximizing the ELBO (Opper & Saad, 2001; Beal, 2003). This approximation requires that the posterior factorizes over the set of the hidden variables $q(\boldsymbol{\beta}) = \prod_{i=1}^{M} q_i(\beta_i)$. It is emphasized that no other assumptions are made about $q(\boldsymbol{\beta})$. The resulting posterior will be determined solely by the assumed likelihood and priors.

In this thesis, we follow the scheme of variational Bayes expectation-maximization (VBEM) as a probabilistic generalization of EM. This approach constitutes a coordinate ascent scheme that iteratively optimizes the ELBO for individual factors of the approximate posterior $q(\boldsymbol{\beta})$ while holding the others constant

$$\ln q_j(\beta_j) = \mathbb{E}_{i \neq j}\big[\log p(\mathcal{D}, \boldsymbol{\beta})\big] + \text{const.}$$

A more practical version of this optimization can be achieved by using stochastic variational inference (SVI), a batched stochastic gradient ascent approach. In the case of the conjugate exponential family, SVI not only facilitates scalability over large datasets but also resembles a natural gradient ascent algorithm on the ELBO with favorable convergence properties (Hoffman et al., 2013).

1.3.4 Relative Entropy Stochastic Control

We rely on trust region stochastic optimization principles to formulate switching and robust control frameworks. Our primary inspiration is model-free and model-based relative entropy policy search algorithms, which constrain the policy updates by using a KL trust region to limit the loss of information between iterations, and explicitly incorporate the system dynamics (Peters et al., 2010; Levine & Koltun, 2013).

The general relative entropy infinite-horizon stochastic optimal control objective, defined over a state space $\mathbf{X}$ and an action space $\mathbf{U}$, takes the following form

$$\underset{\pi,\mu}{\text{maximize}} \quad J = \iint r(\mathbf{x},\mathbf{u})\pi(\mathbf{u}|\mathbf{x})\mu(\mathbf{x})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}, \tag{1.6a}$$

$$\text{subject to} \quad \mu(\mathbf{x}') = \iint \pi(\mathbf{u}|\mathbf{x})\mu(\mathbf{x})p(\mathbf{x}'|\mathbf{x},\mathbf{u})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}, \tag{1.6b}$$

$$\mathrm{KL}(\pi(\mathbf{u}|\mathbf{x})\mu(\mathbf{x})\,||\,q(\mathbf{x},\mathbf{u})) \leq \epsilon, \tag{1.6c}$$

$$\iint \pi(\mathbf{u}|\mathbf{x})\mu(\mathbf{x})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x} = 1, \tag{1.6d}$$

where $r(\mathbf{x},\mathbf{u})$ is a deterministic reward function. Equation (1.6b) describes the evolution of the state distribution $\mu(\mathbf{x})$ according to the dynamics $p(\mathbf{x}'|\mathbf{x},\mathbf{u})$ and the state-conditional policy density $\pi(\mathbf{u}|\mathbf{x})$. The trust region in Equation (1.6c) constrains the state action distribution $p(\mathbf{x},\mathbf{u}) = \pi(\mathbf{u}|\mathbf{x})\mu(\mathbf{x})$ to a KL-ball of size ϵ around a reference distribution $q(\mathbf{x},\mathbf{u})$. Finally, Equation (1.6d) guarantees that $\pi(\mathbf{u}|\mathbf{x})$ and $\mu(\mathbf{x})$ are proper densities.

This optimization problem is generally not tractable for arbitrary nonlinear dynamics and reward functions. In model-free reinforcement learning, a sampled-based approach can be adopted to iteratively find the optimal policy $\pi^*(\mathbf{u}|\mathbf{x})$. However, problems that satisfy the assumption of (time-variant) linear dynamics with quadratic rewards admit a closed-form solution that leads to a regularized forward-backward algorithm closely related to the Riccati equation (Arenz et al., 2016).

1.3.5 Distributional Robustness

Robustness analysis studies the sensitivity of an optimization objective

$$\min_{\mathbf{x}\in\mathbf{X}} \max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} J(\mathbf{x},\boldsymbol{\theta})$$

with a decision variable $\mathbf{x} \in \mathbf{X}$ with respect to to a parameter set $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. The solution $\mathbf{x}^*$ is a conservative point that minimizes the worst-case objective with respect to $\boldsymbol{\theta}$ and delivers an upper-bound on the objective J (Rahimian & Mehrotra, 2019). Furthermore, robust optimization assumes all parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ are equally probable.

In contrast, stochastic optimization assumes the parameters $\boldsymbol{\theta}$ are random variables drawn from a known distribution $p(\boldsymbol{\theta})$. Thus, the objective J can be minimized under a distributional risk measure or an expectation operator, for example,

$$\min_{\mathbf{x}\in\mathbf{X}} \mathbb{E}_{p(\boldsymbol{\theta})}\Big[J(\mathbf{x},\boldsymbol{\theta})\Big].$$

Distributionally robust optimization is a paradigm that combines the concepts of worst-case solutions and distributional uncertainty in one general framework. As in stochastic optimization, the parameters $\boldsymbol{\theta}$ are assumed to be random variables, however, the knowledge about the distribution $p(\boldsymbol{\theta})$ is uncertain. An example of a distributionally robust optimization can be written as

$$\min_{\mathbf{x}\in\mathbf{X}} \max_{p\in\mathbf{P}} \mathbb{E}_{p(\boldsymbol{\theta})}\Big[J(\mathbf{x},\boldsymbol{\theta})\Big],$$

where $\mathbf{P}$ is a set over distributions, commonly referred to as the *ambiguity set*, and contains the worst-case distribution $p^*(\boldsymbol{\theta})$ that upper-bounds the expected loss. The generality of this formulation becomes evident when we consider two different scenarios. In one scenario, the set $\mathbf{P}$ may contain a single distribution, which recovers the stochastic optimization problem. In another, $\mathbf{P}$ may contain all possible distributions with support on $\boldsymbol{\theta}$, thus delivering the classical robust optimization formulation.

The motivation behind distributional robustness considerations pertains to data-driven stochastic learning applications, where the distribution $p(\boldsymbol{\theta})$ is hard to estimate due to limited data. Defining an ambiguity set and optimizing for the worst case is a robust approach to combat statistical learning biases and avoid catastrophic results.

Finally, the choice of the ambiguity set remains important, as it directly influences the overall solution and its usefulness. Sets that are very broadly defined can lead to over-powered biases that cripple the optimization, while a restrictive set definition can undermine the robustness objective. Related literature includes a wide spectrum of possible definitions (Rahimian & Mehrotra, 2019). We focus on ambiguity sets defined using discrepancy measures with respect to a nominal distribution $\hat{p}(\boldsymbol{\theta})$

$$\mathcal{B}_\delta(\hat{p}) = \{p \,|D(p,\hat{p}) \leq \delta\},$$

where D is a measure of the distance or divergence between an arbitrary distribution p and the reference $\hat{p}$. The parameter δ is a set radius around $\hat{p}(\boldsymbol{\theta})$, which effectively bounds the worst-case scenario or the strength of the worst possible case. More specifically, in this thesis, we rely on the Kullback-Leibler divergence as a measure due to its tractable computational properties and its compatibility with trust region stochastic optimization.

Chapter 2
Probabilistic Infinite Local Regression Mixtures

Well-calibrated probabilistic regression models are a crucial learning component in robotics applications as datasets grow rapidly and tasks become more complex. Classical regression models are usually either probabilistic kernel machines with a flexible structure that does not scale gracefully with data or deterministic and vastly scalable automata, albeit with a restrictive parametric form and poor regularization.

In this chapter, we consider a probabilistic hierarchical modeling paradigm that combines the benefits of both worlds to deliver computationally efficient representations with inherent complexity regularization. The presented approaches are probabilistic interpretations of local regression techniques that approximate nonlinear functions through a set of local linear or polynomial units. Importantly, we rely on principles from Bayesian nonparametrics to formulate flexible models that adapt their complexity to the data and can potentially encompass an infinite number of components. We derive two efficient variational inference techniques to learn these representations and highlight the advantages of hierarchical infinite local regression models, such as dealing with non-smooth functions, mitigating catastrophic forgetting, and enabling parameter sharing and fast predictions. Finally, we validate this approach on a set of large inverse dynamics datasets and test the learned models in real-world control scenarios.

2.1 Introduction

Principled data-driven, adaptive and incremental learning is a desirable property in domains in which datasets are dynamic and accumulate slowly over time. For example, robots have to build models of their dynamics and the environment as they interact with the world. Moreover, these models have to be computationally efficient during both the learning and evaluation process. In the case of general-purpose robots, these models must incorporate different modalities of continuous and discrete stochastic random variables and possibly incorporate heteroscedastic noise (Todorov, 2005; Büchler et al., 2018).

Predominant and successful regression techniques, such as Gaussian process regression (GPR) (Rasmussen & Williams, 2006), artificial neural networks (ANNs) (Goodfellow et al., 2016), and local regression (LR) (Wasserman, 2006), have a diverse set of properties that are useful in different scenarios.

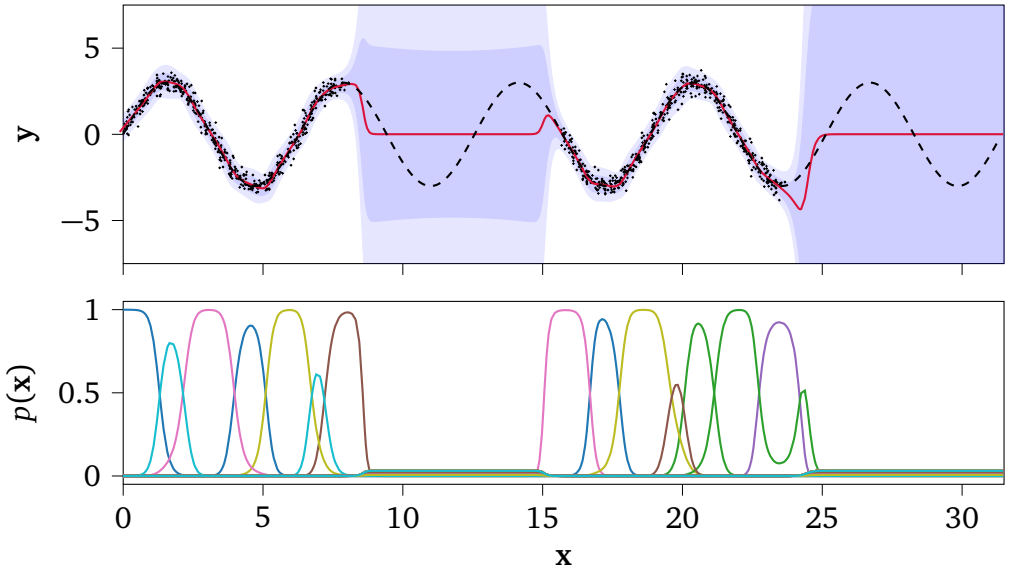


Figure 2.1: Gap data learned with infinite local regression (ILR). The top plot depicts the mean prediction (red) on the training data (dots) and the true mean function (dashed). The shaded blue area represents the predictive uncertainty of two standard deviations. This example highlights how ILR deals with out-of-distribution uncertainty. In areas lacking training data, the predictive uncertainty of ILR is large, the mean prediction falls back to the prior. The bottom plot shows the activation of the local regression models over the input space.

Gaussian process regression offers a principled Bayesian treatment that enables continual and incremental learning. Nonetheless, the *vanilla* formulation of GPR (Rasmussen & Williams, 2006) suffered from many drawbacks that have been gradually addressed by recent research. Some of these drawbacks are the functional smoothness assumption (Calandra et al., 2016; Wilson et al., 2016; Salimbeni & Deisenroth, 2017), limitations when scaling to large datasets (Herbrich et al., 2003; Titsias, 2009; Cao & Fleet, 2014; Deisenroth & Ng, 2015; Bauer et al., 2016; Matthews, 2017) and difficulties modeling heteroscedasticity (Le et al., 2005; Kersting et al., 2007; Liu et al., 2020).

On the other hand, artificial neural networks have proven themselves as very powerful, easy-to-train universal approximators. They are, however, still susceptible to overparameterization (Frankle & Carbin, 2019) and catastrophic forgetting (McCloskey & Cohen, 1989). Moreover, despite major progress on the front of Bayesian neural networks (BNNs) (Neal, 1994; Blundell et al., 2015; Lakshminarayanan et al., 2017; Khan et al., 2018; Sun et al., 2019b; Watson et al., 2021; Daxberger et al., 2021), new evidence suggests that issues regarding the accuracy of uncertainty quantification still need to be tackled (Wenzel et al., 2020; Foong et al., 2020).

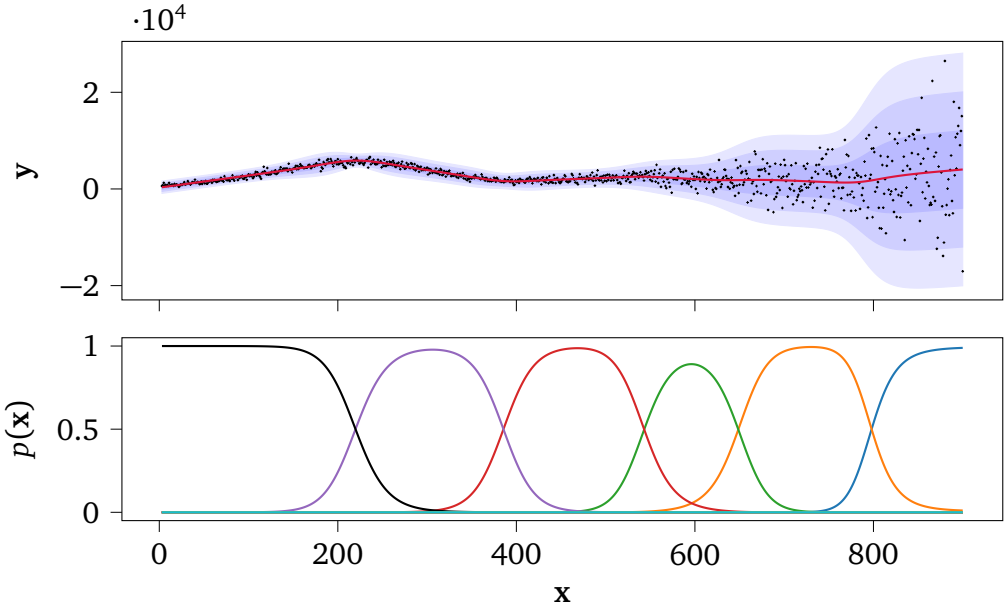


Figure 2.2: The cosmic microwave background (CMB) dataset learned by infinite local regression (ILR). The top figure depicts the mean prediction (red) with three standard deviations predictive uncertainty (shaded blue). ILR captures the heteroscedastic spread of the data with a handful of local regression models. The bottom plot shows the activation of the models over the input space.

Finally, local regression methods have had great success in the domain of robotics and control (Atkeson et al., 1997a; Schaal & Atkeson, 1998; Schaal et al., 2002; Vijayakumar et al., 2005), because of their flexibility, ability to model hard nonlinearities and to incorporate new data online naturally. More generally, local regression is a family of generative mixture of experts (MoE) techniques that take a basis-function approach to model the input density and automatically induces local model responsibilities (Moody & Darken, 1989; Xu et al., 1994; Nelles & Isermann, 1996; Moerland, 1999), see Figures 2.1 and 2.2. In contrast, discriminative MoEs rely on an explicitly input-conditioned gating to choose the local expert (Jacobs et al., 1991; Jordan & Jacobs, 1994; Rasmussen & Ghahramani, 2002).

Two categories of LR exist (Ting et al., 2010), *lazy* learners, that maintain all seen data points in memory (Atkeson et al., 1997a; Schaal et al., 2002), and *memoryless* learners that compress data by constructing basis functions in the input space and fitting and storing locally parameterized regression models (Nelles & Isermann, 1996). Prominent examples of the latter include receptive field weighted regression (RFWR) (Schaal & Atkeson, 1998) and locally weighted projection regression (LWPR) (Vijayakumar et al., 2005). However, these methods are often difficult to tune as they possess many hyperparameters.

A limited attempt at a Bayesian treatment of LR is made in (Ting et al., 2009) by con-

structing local nonparametric kernels and placing gamma priors on the kernel widths to alleviate the need to tune the basis functions. This approach leads to a localized GP formulation that needs to retain the training data in memory, again leading to the computational issues of vanilla GPR. Local Gaussian regression (LGR) is a further Bayesian generalization of LR (Meier et al., 2014). The authors treat the local models in a Bayesian framework and couple them via the loss function that reinforces global coordination. Nonetheless, both approaches rely on heuristics and thresholds for adding and pruning local models and fall short of formulating a generative model over input and output.

Following this introduction, it is our opinion that local regression with a generative Bayesian treatment has the potential to serve as a powerful general-purpose function approximator. Moreover, as a probabilistic and efficient representation, it can drive many low-level applications in robotics that favor fast predictions and do not require deep representations.

In the upcoming sections, we introduce two probabilistic graphical mixture models for local regression. The first, infinite local regression (ILR) (Abdulsamad et al., 2021), is a generative formulation that relies on the paradigm of Bayesian nonparametrics (BNP) (Hjort et al., 2010) to automatically grow the mixture size based on observed data. This technique ultimately results in a general formulation of related methods that alleviates the need for any heuristic considerations. However, despite the effectiveness of ILR, and like other local regression techniques that rely on locally linear or polynomial approximations, it maintains a one-to-one correspondence between the activations and local regression units. This effect limits the model's capacity to share parameters across the input space and often forces the generation of duplicate components, needlessly increasing the overall number of parameters. To address this limitation, we introduce hierarchical infinite local regression (HILR), a multi-level development of ILR that enables multi-modal activations of the same regression component, giving the model a structure that allows sharing of regression parameters across repeating local patterns in the data. This architecture increases the flexibility of the representation and contributes towards its compression.

For learning these models, we derive two general variational Bayes (VB) schemes (Beal, 2003) that efficiently infer the posterior parameters and overcome the need for computationally heavy sampling methods. We benchmark the models on a range of toy tasks that highlight their strengths, such as dealing with heteroscedasticity, non-continuous functions, and multi-modal activation. Additionally, we test on large real-world high-dimensional datasets for learning the inverse dynamics of robotic manipulators. Most importantly, we finally deploy an instance of ILR to perform inverse dynamics control on a real Barrett-WAM robotic manipulator.

Previously mentioned LR methods, including ILR and HILR, mainly rely on locally linear regression models. On the one hand, linear and polynomial components offer a natural unit of approximation. On the other hand, they satisfy real-time computation and memory

requirements in robotics. Nonetheless, multiple Bayesian extensions of generative mixture of Gaussian process experts exist with inference techniques based on Markov chain Monte Carlo (MCMC) (Meeds & Osindero, 2006), and variational inference (VI) (Yuan & Neubauer, 2009). Given that a single Gaussian process is very effective in capturing nonlinear trends, the motivation of constructing such experts is not to increase the quality of approximation but rather to reduce the computational complexity and memory usage during inference and deployment. Furthermore, the infinite mixture paradigm used in ILR and HILR is based on seminal work in Bayesian nonparametrics. We reference influential MCMC sampling techniques for Bayesian nonparametric density estimation (Escobar & West, 1995; Neal, 2000; Ishwaran & James, 2001; Rasmussen, 1999), which developed the first seeds of Bayesian inference for Dirichlet processes (Ferguson, 1973) under the Pólya-urn sampling scheme (Blackwell et al., 1973).

Finally, comparable infinite mixture regression models have been proposed. Prior attempts exclusively rely on expensive Gibbs sampling algorithms for inference (Mueller et al., 1996; Shahbaba & Neal, 2009; Hannah et al., 2011; Gadd et al., 2020). We focus on developing efficient deterministic VI algorithms that improve the practical aspects of training and deploying Bayesian finite and infinite mixture models (Attias, 2000; Blei & Jordan, 2006).

2.2 Preliminaries

In this section, we introduce some related concepts, such as Bayesian linear regression, Bayesian finite mixture models, and the Dirichlet process.

2.2.1 Bayesian Linear Regression

We start by discussing the Bayesian treatment of a single component of a Bayesian local regression model, namely Bayesian linear regression (Minka, 2000a). The conditional data likelihood takes a feature vector $\mathbf{x} \in \mathbb{R}^m$ as a random input variable and returns a response random variable $\mathbf{y} \in \mathbb{R}^d$ according to a linear mapping $\mathbf{A} : \mathbb{R}^m \to \mathbb{R}^d$, a bias vector $\mathbf{c}$, and additive zero-mean noise with a precision matrix $\mathbf{V}$

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{c} + \mathbf{e}, \quad \mathbf{e} \sim \mathrm{N}(\mathbf{0}, \mathbf{V}).$$

For a fully Bayesian treatment, we consider all parameters of this model to be random variables on which we place proper conjugate or semi-conjugate priors. In this case, we place matrix-normal (MN) and normal-Wishart (NW) priors on the matrix $\mathbf{A}$, the bias coefficient $\mathbf{c}$, and precision matrix $\mathbf{V}$

$$p(\mathbf{A}, \mathbf{c}, \mathbf{V}) = \mathrm{MN}(\mathbf{A}|\mathbf{M}, \mathbf{V}, \mathbf{K})\,\mathrm{N}(\mathbf{c}|\boldsymbol{\theta}, \rho\mathbf{V})\,\mathrm{W}(\mathbf{V}|\boldsymbol{\Phi}, \eta),$$

where $\mathbf{M}$, the mean of $\mathbf{A}$, is a $d \times m$ matrix and $\mathbf{V}$ and $\mathbf{K}$ are $d \times d$ and $m \times m$ that serve as row and column precision matrices of $\mathbf{A}$, respectively. The mean $\boldsymbol{\theta}$ is an m-dimensional

vector, and the scalar ρ modulates the amplitude of the precision. Finally, the parameters of the Wishart distribution are the $d \times d$ positive definite scale matrix $\mathbf{\Phi}$ and the degrees of freedom η. Due to the conjugate nature of the priors, the joint posteriors $p(\mathbf{A}, \mathbf{c}, \mathbf{V}|\mathcal{D})$ are matrix-normal and normal-Wishart distributions, conditioned on the data of N independent and identically distributed data pairs $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$.

2.2.2 Bayesian Finite Mixture Models

Gaussian mixture models (GMMs) are hierarchical latent variable models with universal approximation capabilities for arbitrary continuous densities. This insight is of central interest when connected to density estimation for local regression models, which are themselves universal nonlinear function approximators (Wasserman, 2006). A finite K-component Gaussian mixture of a random variable $\mathbf{x}$ is a weighted combination of densities

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} p(\mathbf{z} = k|\boldsymbol{\pi}) p(\mathbf{x}|\boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k \, \mathrm{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k),$$

with K unique mean vectors $\boldsymbol{\mu}_k$ and precision matrices $\boldsymbol{\Lambda}_k$. The latent quantity $\mathbf{z}$ is a one-hot random variable distributed according to a categorical distribution $p(\mathbf{z}) = \mathrm{Cat}(\boldsymbol{\pi})$, governed by the weights $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$ that satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.

The Bayesian extension of this model (Attias, 2000) introduces a conjugate normal-Wishart prior on the means and precision matrices $(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \sim \mathrm{NW}(\boldsymbol{\lambda})$, where $\boldsymbol{\lambda}$ contains the hyperparameters. Furthermore, a conjugate Dirichlet prior, with a concentration parameter α, is placed on the mixing weights $\boldsymbol{\pi} \sim \mathrm{Dir}(\boldsymbol{\alpha})$.

This Bayesian approach has proven effective in regularizing GMMs by allowing superfluous components to fall back onto their priors instead of severely overfitting to small clusters. This effect can be understood as sparsification bias over K (Beal & Ghahramani, 2006; Rousseau & Mengersen, 2011).

2.2.3 Dirichlet Process and Stick-Breaking

A Dirichlet process (DP) is a distribution over probability measures G. We write $\mathrm{G} \sim \mathrm{DP}(\alpha, \mathrm{H})$, where α is the concentration parameter and H is the base measure (Murphy, 2012; Teh, 2010). Intuitively, a Dirichlet process is a distribution over distributions, meaning each draw G is a distribution. The base distribution H is the mean of the DP, and the concentration parameter α is the inverse variance. The larger α is, the smaller the variance, and the process concentrates more of its mass around the mean distribution H.

We will rely on the stick-breaking construction (Sethuraman, 1994) of a DP as an algorithmic realization. Stick-breaking delivers an infinite sequence of mixture weights π_k of

an infinite mixture model from the stochastic process

$$\pi_k = s_k \prod_{l=1}^{k-1} \big(1 - s_l\big), \quad s_k \sim \text{Beta}(1, \alpha).$$

This process is sometimes denoted as $\boldsymbol{\pi} \sim \text{GEM}(\alpha)$ (Murphy, 2012). The stick-breaking procedure describes how the random variables s_k, representing stick lengths, are drawn from a beta distribution and combined to obtain the mixture weights π_k. If the concentration parameter α increases, the magnitude of the mixing weights π_k decreases on average, and the number of possible active components increases. This representation of DPs can be used to replace the priors placed on the finite Gaussian mixture model (Blei & Jordan, 2006). In such a setting, the base H is a normal-Wishart distribution, and the sampled measure $\text{G} \sim \text{DP}(\alpha, \text{NW})$ is a draw of an unbounded number of parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \sim \text{NW}(\boldsymbol{\lambda})$ for an infinite number of clusters, associated with an infinite number of weights π_k generated by the stick-breaking process. These draws from a Dirichlet process are discrete with probability one, which leads to the clustering effect of the DP. Eventually, the same parameters will be sampled over and over, forcing the associated data points to cluster.

2.3 Infinite Mixture of Local Regression

Using the previously presented concepts of Bayesian linear regression, Bayesian mixture models, and Dirichlet processes, we now construct the Bayesian infinite local regression (ILR) model. Our approach to solving the regression task is mainly a Bayesian joint density estimation task. We assume a generative process as depicted in Figure 2.3. A Dirichlet process is sampled to generate the categorical weights $\boldsymbol{\pi}$, mixture activations $\{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\}_{k=1}^{\infty}$, and regression parameters $\{\mathbf{A}_k, \mathbf{c}_k, \mathbf{V}_k\}_{k=1}^{\infty}$

$$\begin{gathered}
\boldsymbol{\pi}(\mathbf{s}) \sim \text{GEM}(\alpha), \\
\boldsymbol{\Lambda}_k \sim \text{W}(\boldsymbol{\Psi}, \nu), \;\; \boldsymbol{\mu}_k \sim \text{N}(\mathbf{m}, \kappa\boldsymbol{\Lambda}_k), \\
\mathbf{V}_k \sim \text{W}(\boldsymbol{\Phi}, \eta), \;\; \mathbf{A}_k \sim \text{MN}(\mathbf{M}, \mathbf{K}, \mathbf{V}_k), \;\; \mathbf{c}_k \sim \text{N}(\boldsymbol{\theta}, \rho\mathbf{V}_k),
\end{gathered}$$

and those parameters generate the one-hot labels $\mathbf{z}_n$ and data pairs $\mathbf{x}_n, \mathbf{y}_n$

$$\mathbf{z}_n \sim \text{Cat}(\boldsymbol{\pi}(\mathbf{s})), \quad \mathbf{x}_n \sim \text{N}(\boldsymbol{\mu}_{z_n}, \boldsymbol{\Lambda}_{z_n}), \quad \mathbf{y}_n \sim \text{N}(\mathbf{A}_{z_n}\mathbf{x}_n + \mathbf{c}_{z_n}, \mathbf{V}_{z_n}).$$

Notice that the densities over the input space naturally play the role of basis functions or so-called receptive fields as in the receptive field weighted regression (Schaal et al., 2002) and locally weighted projection regression (Vijayakumar et al., 2005) algorithms.

The next step is to infer the joint posterior $p(\mathbf{Z}, \mathbf{s}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{A}, \mathbf{c}, \mathbf{V} | \mathbf{X}, \mathbf{Y})$ over labels and parameters from data, where $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$, and $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$. Our method of choice, variational Bayes expectation-maximization (VBEM), alternates

between a variational expectation step (E-step) and maximization step (M-step), see Section 1.3.3. Deriving such an algorithm requires pinning down the following definitions of the likelihood, prior and posterior.

2.3.1 Complete Data Likelihood

We assume the following form of the joint likelihood For the general case of multivariate regression with m inputs and d outputs, we assume the following structured joint likelihood over data and indicators

$$\begin{aligned} p(\mathbf{X},\mathbf{Y},\mathbf{Z}|.) &= p(\mathbf{Z})\,p(\mathbf{X}|\mathbf{Z})\,p(\mathbf{Y}|\mathbf{Z},\mathbf{X}) \\ &= \prod_{n=1}^{N} \mathrm{Cat}(\mathbf{z}_n|\boldsymbol{\pi}(\mathbf{s})) \\ &\quad \times \prod_{n=1}^{N}\prod_{k=1}^{\infty} \mathrm{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k)^{z_{nk}} \\ &\quad \times \prod_{n=1}^{N}\prod_{k=1}^{\infty} \mathrm{N}(\mathbf{y}_n|\mathbf{A}_k\mathbf{x}_n+\mathbf{c}_k,\mathbf{V}_k)^{z_{nk}}. \end{aligned}$$

2.3.2 Infinite Conjugate Prior

We assume the factorized conjugate infinite mixture prior

$$p(\mathbf{s},\boldsymbol{\mu},\boldsymbol{\Lambda},\mathbf{A},\mathbf{c},\mathbf{V}) = p(\mathbf{s})\,p(\boldsymbol{\mu}|\boldsymbol{\Lambda})\,p(\boldsymbol{\Lambda})p(\mathbf{A}|\mathbf{V})\,p(\mathbf{c}|\mathbf{V})\,p(\mathbf{V}).$$

This prior samples the cluster means $\boldsymbol{\mu}_k$ and precision matrices $\boldsymbol{\Lambda}_k$ from a normal-Wishart

$$p(\boldsymbol{\mu}|\boldsymbol{\Lambda})\,p(\boldsymbol{\Lambda}) = \prod_{k=1}^{\infty} \mathrm{N}(\boldsymbol{\mu}_k|\mathbf{m}_0,\kappa_0\boldsymbol{\Lambda}_k)\,\mathrm{W}(\boldsymbol{\Lambda}_k|\boldsymbol{\Psi}_0,\nu_0),$$

while matrix-normal-Wishart and a normal-Wishart priors are placed on the regression coefficients $(\mathbf{A}_k,\mathbf{c}_k)$ and the precision matrices $\mathbf{V}_k$

$$p(\mathbf{A}|\mathbf{V})\,p(\mathbf{c}|\mathbf{V})\,p(\mathbf{V}) = \prod_{k=1}^{\infty} \mathrm{MN}(\mathbf{A}_k|\mathbf{M}_0,\mathbf{K}_0,\mathbf{V}_k)\,\mathrm{N}(\mathbf{c}_k|\boldsymbol{\theta}_0,\rho_0\mathbf{V}_k)\,\mathrm{W}(\mathbf{V}_k|\boldsymbol{\Phi}_0,\eta_0).$$

The parameters π_k are generated by a stick-breaking process $\pi_k(\mathbf{s}) = s_k\prod_{l=1}^{k-1}(1-s_l)$. The parameters $\mathbf{s} = \{s_i,\ldots,s_K\}$ are independently beta distributed

$$p(\mathbf{s}) = \prod_{k=1}^{\infty} \mathrm{Beta}(s_k|1,\alpha_0).$$

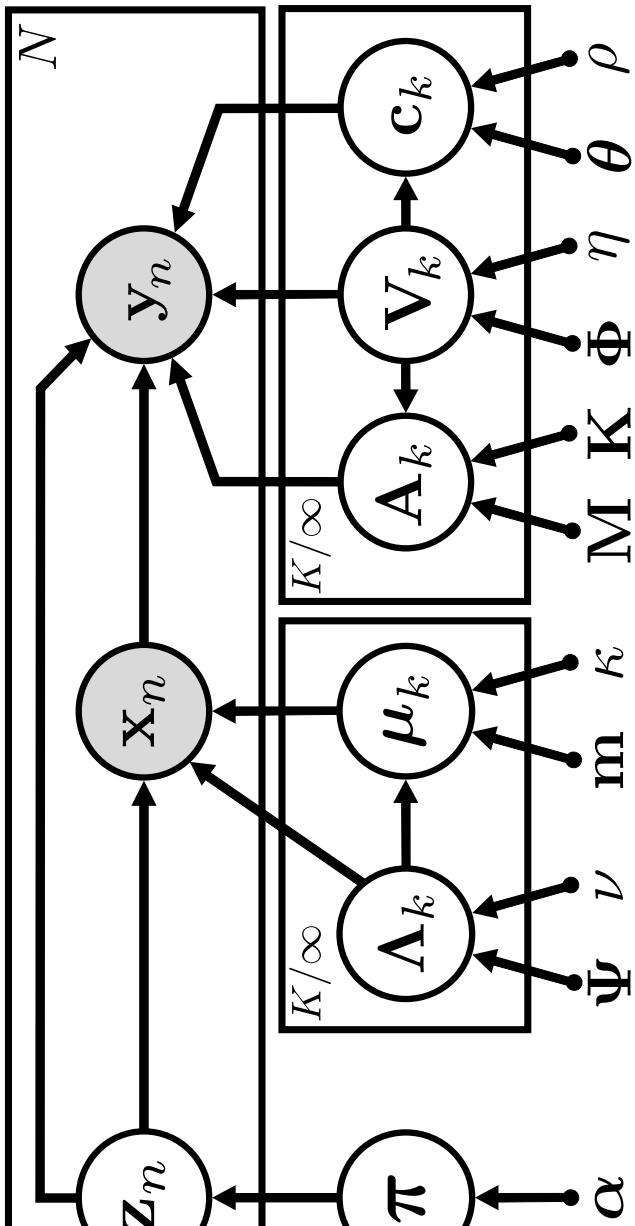


Figure 2.3: A unified plate notation for infinite mixtures of Bayesian local regression. Assuming Gaussian and linear Gaussian densities, the basis parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$ are sampled from a normal-Wishart distribution, while the regression parameters $(\mathbf{A}_k, \mathbf{c}_k)$, and precision matrices $\mathbf{V}_k$, are sampled from a matrix-Normal-Wishart, for every component k. The latent variables $\mathbf{z}_n$ assign every $\mathbf{x}_n$ and $\mathbf{y}_n$ to a component and are drawn from a categorical distribution parameterized by $\boldsymbol{\pi}$. The mixture weights $\boldsymbol{\pi}$ are generated by a stick-breaking process with a concentration parameter α.

2.3.3 Truncated Mean-Field Factorization

We rely on a structured mean-field approximation of the posterior (Opper & Saad, 2001) that factorizes between the labels $q(\mathbf{Z})$ and the remaining parameters $q(\mathbf{s}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{A}, \mathbf{c}, \mathbf{V})$, thus automatically leading to the following decomposition

$$p(\mathbf{Z}, \mathbf{s}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{A}, \mathbf{c}, \mathbf{V} | \mathcal{D}) \approx q(\mathbf{Z})\, q(\mathbf{s})\, q(\boldsymbol{\mu}, \boldsymbol{\Lambda})\, q(\mathbf{A}, \mathbf{c}, \mathbf{V}).$$

Further, we follow (Blei & Jordan, 2006) by allowing a truncation of the posterior while maintaining an infinite prior, so that $q(s_K = 1) = 1$, implying that $\pi_k = 0$ for $k > K$

$$\begin{aligned}
q &= q(\mathbf{Z})\, q(\mathbf{s})\, q(\boldsymbol{\mu}, \boldsymbol{\Lambda})\, q(\mathbf{A}, \mathbf{c}, \mathbf{V}) \\
&= \prod_{n=1}^{N} \mathrm{Cat}(\mathbf{z}_n | \mathbf{r}_n) \prod_{k=1}^{K-1} \mathrm{Beta}(s_k | \gamma_k, \alpha_k) \\
&\quad \times \prod_{k=1}^{K} \mathrm{N}(\boldsymbol{\mu}_k | \mathbf{m}_k, \kappa_k \boldsymbol{\Lambda}_k)\, \mathrm{W}(\boldsymbol{\Lambda}_k | \boldsymbol{\Psi}_k, \nu_k) \\
&\quad \times \prod_{k=1}^{K} \mathrm{MN}(\mathbf{A}_k | \mathbf{M}_k, \mathbf{K}_k, \mathbf{V}_k)\, \mathrm{N}(\mathbf{c}_k | \boldsymbol{\theta}_k, \rho_k \mathbf{V}_k)\, \mathrm{W}(\mathbf{V}_k | \boldsymbol{\Phi}_k, \eta_k),
\end{aligned}$$

where $\mathbf{r}_n$ are the expected responsibilities of the mixture. During evaluation, the truncation threshold K is chosen to be very high and is seldom reached.

2.3.4 Variational Expectation Step

In the E-step, the responsibilities $\mathbf{r}_n$ are computed by following the recipe of VBEM. The responsibilities are variational parameters of the posterior categorical

$$\begin{aligned}
\log q(\mathbf{Z}) &= \mathbb{E}_{q(\mathbf{s})}\left[\log p(\mathbf{Z} | \boldsymbol{\pi}(\mathbf{s}))\right] + \mathbb{E}_{q(\boldsymbol{\mu}, \boldsymbol{\Lambda})}\left[\log p(\mathbf{X} | \mathbf{Z})\right] \\
&\quad + \mathbb{E}_{q(\mathbf{A}, \mathbf{c}, \mathbf{V})}\left[\log p(\mathbf{Y} | \mathbf{Z}, \mathbf{X})\right] + \text{const} \\
&= \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{s})}\left[\log \mathrm{Cat}(\mathbf{z}_n | \boldsymbol{\pi})\right] + \sum_{k=1}^{K} \sum_{n=1}^{N} z_{nk}\, \mathbb{E}_{q(\boldsymbol{\mu}, \boldsymbol{\Lambda})}\left[\log \mathrm{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)\right] \\
&\quad + \sum_{k=1}^{K} \sum_{n=1}^{N} z_{nk}\, \mathbb{E}_{q(\mathbf{A}, \mathbf{c}, \mathbf{V})}\left[\log \mathrm{N}(\mathbf{y}_n | \mathbf{A}_k \mathbf{x}_n + \mathbf{c}_k, \mathbf{V}_k)\right] + \text{const} \\
&= \sum_{k=1}^{K} \sum_{n=1}^{N} z_{nk} \log r_{nk}.
\end{aligned}$$

The expectations associated with the Gaussian likelihoods are straightforwardly computed (Bishop, 2006). The expectations associated with infinite-dimensional categorical require more consideration (Blei & Jordan, 2006). We provide the necessary details in Appendix B.

2.3.5 Variational Maximization Step

The M-step updates the variational distributions given the responsibilities as follows

$$\begin{aligned}
\log q(\mathbf{s}) &= \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{Z}|\boldsymbol{\pi}(\mathbf{s}))\right] + \log p(\mathbf{s}) + \text{const} \\
&= \sum_{k=1}^{K}\sum_{n=1}^{N}\mathbb{E}_{q(\mathbf{z})}\left[z_{nk}\right]\log\left[s_k\prod_{l=1}^{k}(1-s_l)\right] + \sum_{k=1}^{K}\log \text{Beta}(s_k|1,\alpha_0) + \text{const}, \\
\log q(\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{X}|\mathbf{Z})\right] + \log p(\boldsymbol{\mu},\boldsymbol{\Lambda}) + \text{const} \\
&= \sum_{k=1}^{K}\sum_{n=1}^{N}\mathbb{E}_{q(\mathbf{z})}\left[z_{nk}\right]\log \text{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k) + \sum_{k=1}^{K}\log \text{W}(\mathbf{V}_k|\boldsymbol{\Psi}_0,\nu_0) + \text{const}, \\
\log q(\mathbf{A},\mathbf{c},\mathbf{V}) &= \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{Y}|\mathbf{Z},\mathbf{X})\right] + \log p(\mathbf{A},\mathbf{c},\mathbf{V}) + \text{const} \\
&= \sum_{k=1}^{K}\sum_{n=1}^{N}\mathbb{E}_{q(\mathbf{z})}\left[z_{nk}\right]\log \text{N}(\mathbf{y}_n|\mathbf{A}_k\mathbf{x}_n+\mathbf{c}_k,\mathbf{V}_k) + \sum_{k=1}^{K}\log \text{W}(\mathbf{V}_k|\boldsymbol{\Phi}_0,\eta_0) \\
&\quad + \sum_{k=1}^{K}\log \text{MN}(\mathbf{A}_k|\mathbf{M}_0,\mathbf{K}_0,\mathbf{V}_k) + \sum_{k=1}^{K}\log \text{N}(\mathbf{c}_k|\boldsymbol{\theta}_0,\rho_0\mathbf{V}_k) + \text{const},
\end{aligned}$$

where $\mathbb{E}_{q(\mathbf{z})}\left[z_{nk}\right] = r_{nk}$. Consequently, each update reflects a conjugate computation of K log-posterior densities given a log-prior and a weighted log-likelihood. We provide general recipes for these updates in Appendix A.

2.3.6 Posterior Predictive Distribution

For predicting the function value $\hat{\mathbf{y}}$ conditioned on a test query $\hat{\mathbf{x}}$, we marginalize the likelihood over the posterior parameters to get the joint posterior predictive density. To make the marginalization tractable, we replace the true posterior with our approximate variational posterior $q(.|\mathcal{D})$ inferred under a training dataset $\mathcal{D}$.

The conditional predictive for a single component $\mathbf{z} = k$ is a Student's t-distribution

$$\begin{aligned}
p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\hat{\mathbf{z}}=k,\mathcal{D}) &= \mathbb{E}_{q(\mathbf{A}_k,\mathbf{c}_k,\mathbf{V}_k)}\left[p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\mathbf{A}_k,\mathbf{c}_k,\mathbf{V}_k)\right] \\
&= \text{T}\left(\mathbf{M}_k\hat{\mathbf{x}}+\boldsymbol{\theta}_k, a_k\boldsymbol{\Phi}_k, \eta_k+1\right),
\end{aligned}$$

where we have defined

$$a_k = 1 - \tilde{\mathbf{x}}^\top\left(\mathbf{L}_k + \tilde{\mathbf{x}}\tilde{\mathbf{x}}^\top\right)^{-1}\tilde{\mathbf{x}},$$

with $\tilde{\mathbf{x}} = [\hat{\mathbf{x}},\ 1]^\top$ and $\mathbf{L}_k = \text{Block}\left(\mathbf{K}_k, \rho_k\right)$.

Additionally, the joint activation of a component k is a Student's t-distribution weighted

by the expected categorical probability under the posterior stick-breaking process

$$
\begin{aligned}
p(\hat{\mathbf{x}}, \hat{\mathbf{z}} = k|\mathcal{D}) &\propto \mathbb{E}_{q(s_k)}\left[p(\hat{\mathbf{z}} = k|s_k)\right]\mathbb{E}_{q(\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k)}\left[p(\hat{\mathbf{x}}|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)\right] \\
&= \frac{\gamma_k}{\gamma_k + \alpha_k}\prod_{l=1}^{k-1}\left(1 - \frac{\gamma_l}{\gamma_l + \alpha_l}\right) \\
&\quad \times \mathrm{T}\left(\boldsymbol{\mu}_k, \frac{\kappa_k}{1+\kappa_k}\boldsymbol{\Psi}_k, \nu_k + 1\right).
\end{aligned}
$$

These K-activations enable two prediction techniques. A *mode-prediction*, where the most likely active component is selected and used to perform prediction with the corresponding linear regression model, or a *mean-prediction*, that averages the predictions of all components weighted by their activation probabilities.

2.3.7 Computational Complexity

We calculate the training-time computational cost to be $\mathcal{O}(NK(d+m)^3)$, which can be straightforwardly reduced to $\mathcal{O}(LK(d+m)^3)$ by applying stochastic updates (Hoffman et al., 2013), where L is the batch size. This result shows linear scalability with the data, which is considerably more efficient than simple variants of GPR. The test-time complexity of a mean prediction is $\mathcal{O}(K(d^3+dm))$, which combines the input membership query and the linear matrix transformation for every model k. This computation is, in contrast to GPR, independent of the training data size, hence the advantage of memoryless locally-parametric representations during real-time critical applications.

2.4 Hierarchical Infinite Mixture of Local Regression

The local regression model presented in Section 2.3 offers a very flexible and well-regularized alternative to previously developed approaches (Schaal & Atkeson, 1998; Vijayakumar et al., 2005; Meier et al., 2014). However, like other representations, it suffers from a subtle drawback that can cause it to generate duplicate regression components to account for similar local function trends across disconnected regions of the input space.

This issue is a consequence of the hierarchical design that directly couples activations and local function approximations via a one-to-one correspondence and enforces a uni-modal activation per regression component. This coupling can be clearly observed in the definition of the likelihood in Section 2.3.1, where the activation and the local regression units share the same assignment variable. It stands to reason that this architecture does not offer enough flexibility and hinders parameter sharing between components. We, therefore, argue for a modified formulation of ILR that explicitly accounts for shift-invariance in the input space and provides the freedom to create regression units with multi-modal, theoretically infinitely-modal, activations, if needed.

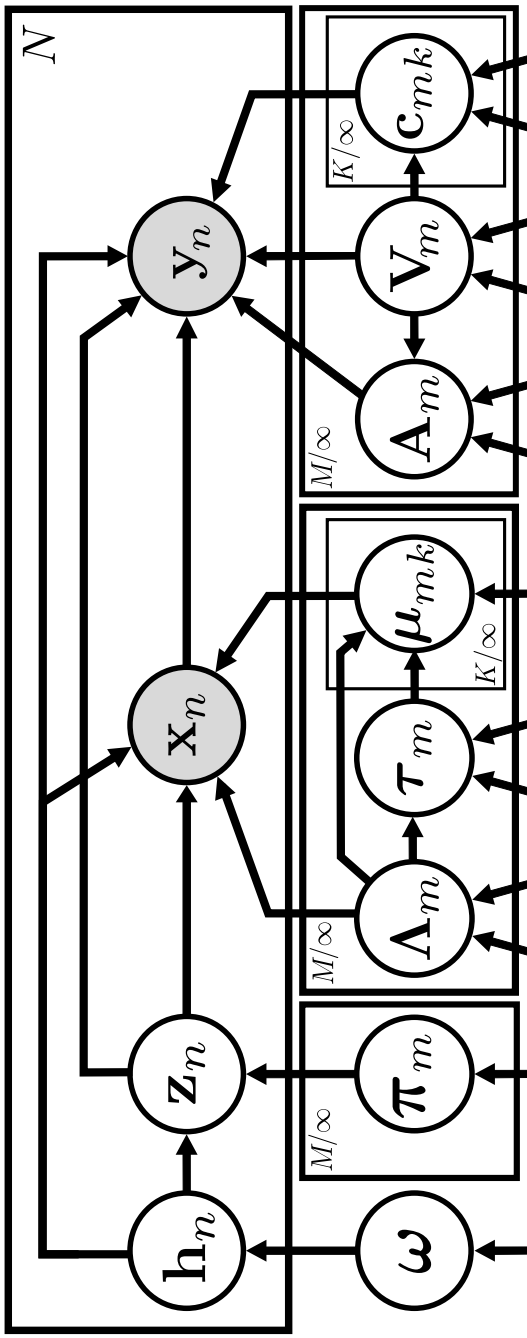


Figure 2.4: A unified plate notation for hierarchical infinite tied mixtures of Bayesian local regression models. This model outlines a two-level architecture that allows sharing of parameters between single components in order to compress the representation. It can be interpreted as a local regression model with multi-modal activation. Each unit of the upper level is itself a mixture of local regression models that share the same slope $\mathbf{A}_m$ and output precision $\mathbf{V}_m$. Each of these m different slopes can be activated at k unique lower-level input regions centered around $\boldsymbol{\mu}_{mk}$ and tied via a shared input precision $\boldsymbol{\Lambda}_m$. The upper- and lower-level mixtures are governed by independent Dirichlet process priors.

Following this motivation, we formulate hierarchical infinite local regression (HILR), an infinite mixture over infinite mixtures, and sketch out a structured variational inference algorithm to approximate its posterior. The resulting model shares some similarities with existing representations developed for hierarchical clustering (Yerebakan et al., 2014; Nguyen et al., 2014; Huynh et al., 2016). We start by describing the generative process of the hierarchical tied mixture as depicted in Figure 2.4. An upper-level Dirichlet process, the meta-process indexed by m, generates the stick-breaking weights $\boldsymbol{\omega}$, the meta-activations $\{\boldsymbol{\tau}_m, \boldsymbol{\Lambda}_m\}_{m=1}^{\infty}$, and the shared slope and output precision matrices $\{\mathbf{A}_m, \mathbf{V}_m\}_{m=1}^{\infty}$

$$\begin{aligned}\boldsymbol{\omega}(\mathbf{t}) &\sim \text{GEM}(\beta),\\ \boldsymbol{\Lambda}_m \sim \text{W}(\boldsymbol{\Psi}, \nu),\ \boldsymbol{\tau}_m &\sim \text{N}(\mathbf{m}, \lambda\boldsymbol{\Lambda}_m),\\ \mathbf{V}_m \sim \text{W}(\boldsymbol{\Phi}, \eta),\ \mathbf{A}_m &\sim \text{MN}(\mathbf{M}, \mathbf{K}, \mathbf{V}_m),\end{aligned}$$

where $\mathbf{t}$ are the stick lengths of the corresponding upper-level DP. The lower-level DPs, indexed by k, generate the weights $\boldsymbol{\pi}_m$, the multi-modal activation centers $\{\boldsymbol{\mu}_{mk}\}_{k=1}^{\infty}$ and the affine shift coefficients $\{\mathbf{c}_{mk}\}_{k=1}^{\infty}$

$$\boldsymbol{\pi}_m(\mathbf{s}_m) \sim \text{GEM}(\alpha), \quad \boldsymbol{\mu}_{mk} \sim \text{N}(\boldsymbol{\tau}_m, \kappa\boldsymbol{\Lambda}_m), \quad \mathbf{c}_{mk} \sim \text{N}(\boldsymbol{\theta}, \rho\mathbf{V}_m),$$

which in turn generate the upper- and lower-level labels $\mathbf{h}_n, \mathbf{z}_n$ and the data pairs $\mathbf{x}_n, \mathbf{y}_n$

$$\begin{aligned}\mathbf{h}_n \sim \text{Cat}(\boldsymbol{\omega}(\mathbf{t})),\ \mathbf{z}_n &\sim \text{Cat}(\boldsymbol{\pi}(\mathbf{s}), \mathbf{h}_n),\\ \mathbf{x}_n \sim \text{N}(\boldsymbol{\mu}_{h_n,z_n}, \boldsymbol{\Lambda}_{h_n}),\ \mathbf{y}_n &\sim \text{N}(\mathbf{A}_{h_n}\mathbf{x}_n + \mathbf{c}_{h_n,z_n}, \mathbf{V}_{h_n}).\end{aligned}$$

Analogous to Section 2.3, we formulate the quantities involved in deriving a structured VBEM algorithm for this model. These quantities are the complete data likelihood, the infinite prior, and the mean-field posterior factorization.

2.4.1 Complete Data Likelihood

The likelihood model is a two-level precision-tied joint density over the observations $\mathbf{X}, \mathbf{Y}$ and the one-hot upper- and lower-labels $\mathbf{H}, \mathbf{Z}$

$$\begin{aligned}p(.) &= p(\mathbf{H})\, p(\mathbf{Z}|\mathbf{H})\, p(\mathbf{X}|\mathbf{H}, \mathbf{Z})\, p(\mathbf{Y}|\mathbf{H}, \mathbf{Z}, \mathbf{X})\\ &= \prod_{n=1}^{N} \text{Cat}(\mathbf{h}_n|\boldsymbol{\omega}(\mathbf{t})) \prod_{n=1}^{N} \text{Cat}(\mathbf{z}_n|\boldsymbol{\pi}(\mathbf{s}), \mathbf{h}_n)\\ &\quad \times \prod_{n=1}^{N}\prod_{m=1}^{\infty}\prod_{k=1}^{\infty} \text{N}(\mathbf{x}_n|\boldsymbol{\mu}_{mk}, \boldsymbol{\Lambda}_m)^{z_{nk}\times h_{nm}}\\ &\quad \times \prod_{n=1}^{N}\prod_{m=1}^{\infty}\prod_{k=1}^{\infty} \text{N}(\mathbf{y}_n|\mathbf{A}_m\mathbf{x}_n + \mathbf{c}_{mk}, \mathbf{V}_m)^{z_{nk}\times h_{nm}},\end{aligned}$$

2.4.2 Infinite Conjugate Prior

We assume a factorized two-level tied conjugate infinite mixture prior

$$p(\mathbf{t}, \mathbf{s}, \boldsymbol{\mu}, \boldsymbol{\tau}, \boldsymbol{\Lambda}, \mathbf{A}, \mathbf{c}, \mathbf{V}) = p(\mathbf{t})\, p(\mathbf{s})\, p(\boldsymbol{\mu}|\boldsymbol{\tau}, \boldsymbol{\Lambda})\, p(\boldsymbol{\tau}|\boldsymbol{\Lambda})\, p(\boldsymbol{\Lambda})\, p(\mathbf{A}|\mathbf{V})\, p(\mathbf{c}|\mathbf{V})\, p(\mathbf{V}).$$

The meta activation prior is a normal-Wishart distribution over the meta centers $\boldsymbol{\tau}_m$ and precision matrices $\boldsymbol{\Lambda}_m$

$$p(\boldsymbol{\tau}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda}) = \prod_{m=1}^{\infty} \mathrm{N}(\boldsymbol{\tau}_m|\mathbf{m}_0, \lambda_0\boldsymbol{\Lambda}_m)\, \mathrm{W}(\boldsymbol{\Lambda}_m|\boldsymbol{\Psi}_0, \nu_0),$$

while the activation centers $\boldsymbol{\mu}_{mk}$ are sampled from a conditional normal distribution

$$p(\boldsymbol{\mu}|\boldsymbol{\tau}, \boldsymbol{\Lambda}) = \prod_{m=1}^{\infty}\prod_{k=1}^{\infty} \mathrm{N}(\boldsymbol{\mu}_{mk}|\boldsymbol{\tau}_m, \kappa_0\boldsymbol{\Lambda}_m).$$

The mappings $\mathbf{A}_m$ and precision matrices $\mathbf{V}_m$ are sampled form a matrix-normal-Wishart

$$p(\mathbf{A}|\mathbf{V})p(\mathbf{V}) = \prod_{m=1}^{\infty} \mathrm{MN}(\mathbf{A}_m|\mathbf{M}_0, \mathbf{K}_0, \mathbf{V}_m)\, \mathrm{W}(\mathbf{V}_m|\boldsymbol{\Phi}_0, \eta_0).$$

while the biases $\mathbf{c}_{mk}$ are drawn from a K-tied conditional normal

$$p(\mathbf{c}|\mathbf{V}) = \prod_{m=1}^{\infty}\prod_{k=1}^{\infty} \mathrm{MN}(\mathbf{c}_{mk}|\boldsymbol{\theta}_0, \rho_0\mathbf{V}_m).$$

Finally, the stick-breaking priors $p(\mathbf{t}, \mathbf{s})$ follow the definitions from Section 2.3

$$p(\mathbf{t}) = \prod_{m=1}^{\infty} \mathrm{Beta}(t_k|1, \beta_0),$$
$$p(\mathbf{s}) = \prod_{m=1}^{\infty}\prod_{k=1}^{\infty} \mathrm{Beta}(s_{mk}|1, \alpha_0).$$

2.4.3 Truncated Mean-Field Factorization

We assume a structured decomposition of the posterior that leads to conjugate computation while maintaining the dependencies between the discrete labels, the input activations, and the regression parameters, respectively. Moreover, we apply the truncation scheme

from (Blei & Jordan, 2006) to establish the following posterior approximation

$$
\begin{aligned}
p(.|\mathcal{D}) &\approx q(\mathbf{H})\, q(\mathbf{Z}|\mathbf{H})\, q(\mathbf{t})\, q(\mathbf{s})\, q(\boldsymbol{\mu}, \boldsymbol{\tau}, \boldsymbol{\Lambda})\, q(\mathbf{A}, \mathbf{c}, \mathbf{V}) \\
&= \prod_{n=1}^{N} \mathrm{Cat}(\mathbf{h}_n|\mathbf{g}_n)\, \mathrm{Cat}(\mathbf{z}_n|\mathbf{h}_n, \mathbf{r}_n) \\
&\times \prod_{m=1}^{M-1} \mathrm{Beta}(t_m|\delta_m, \beta_m) \\
&\times \prod_{m=1}^{M-1} \prod_{k=1}^{K-1} \mathrm{Beta}(s_{mk}|\gamma_{mk}, \alpha_{mk}) \\
&\times \prod_{m=1}^{M} \mathrm{N}(\boldsymbol{\tau}_m|\mathbf{m}_m, \lambda_m \boldsymbol{\Lambda}_m)\, \mathrm{W}(\boldsymbol{\Lambda}_m|\boldsymbol{\Psi}_m, \nu_m) \\
&\times \prod_{m=1}^{M} \mathrm{MN}(\mathbf{A}_m|\mathbf{M}_m, \mathbf{K}_m, \mathbf{V}_m)\, \mathrm{W}(\mathbf{V}_m|\boldsymbol{\Phi}_m, \eta_m) \\
&\times \prod_{m=1}^{M} \prod_{k=1}^{K} \mathrm{N}(\boldsymbol{\mu}_{mk}|\boldsymbol{\tau}_m, \kappa_{mk} \boldsymbol{\Lambda}_m)\, \mathrm{N}(\mathbf{c}_{mk}|\boldsymbol{\theta}_{mk}, \rho_{mk} \mathbf{V}_k),
\end{aligned}
$$

where $\mathbf{g}_n$ and $\mathbf{r}_n$ are the upper- and lower-level posterior responsibilities, respectively.

2.4.4 Variational Expectation Step

The E-step computes the joint posterior categorical over joint labels $\mathbf{H}$ and $\mathbf{Z}$

$$
\begin{aligned}
\log q(\mathbf{Z}|\mathbf{H}) &= \mathbb{E}_{q(\mathbf{s})}\left[\log p(\mathbf{Z}|\mathbf{H})\right] \\
&\quad + \mathbb{E}_{q(\boldsymbol{\mu}, \boldsymbol{\tau}, \boldsymbol{\Lambda})}\left[\log p(\mathbf{X}|\mathbf{H}, \mathbf{Z})\right] \\
&\quad + \mathbb{E}_{q(\mathbf{A}, \mathbf{c}, \mathbf{V})}\left[\log p(\mathbf{Y}|\mathbf{H}, \mathbf{Z}, \mathbf{X})\right] + \text{const} \\
&= \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{n=1}^{N} h_{nm} z_{nmk} \log r_{nmk}, \\
\log q(\mathbf{H}) &= \mathbb{E}_{q(\mathbf{t})}\left[\log p(\mathbf{H})\right] + \log q(\mathbf{Z}|\mathbf{H}) + \text{const} \\
&= \sum_{m=1}^{M} \sum_{n=1}^{N} h_{nm} \log g_{nm},
\end{aligned}
$$

where these expectations can computed in a similar fashion to Section 2.3.4 and Appendix B.

2.4.5 Variational Maximization Step

The M-step updates the variational gating, activation, and regression parameters

$$\log q(\mathbf{t}) = \mathbb{E}_{q(\mathbf{H})}\big[\log p(\mathbf{H})\big] + \log p(\mathbf{t}) + \text{const},$$

$$\log q(\mathbf{s}) = \mathbb{E}_{q(\mathbf{H},\mathbf{Z})}\big[\log p(\mathbf{Z}|\mathbf{H})\big] + \log p(\mathbf{s}) + \text{const},$$

$$\log q(\boldsymbol{\mu}) = \mathbb{E}_{q(\mathbf{H},\mathbf{Z},\boldsymbol{\tau},\boldsymbol{\Lambda})}\big[\log p(\mathbf{X}|\mathbf{H},\mathbf{Z})\big] + \mathbb{E}_{q(\boldsymbol{\tau},\boldsymbol{\Lambda})}\big[\log p(\boldsymbol{\mu}|\boldsymbol{\tau},\boldsymbol{\Lambda})\big] + \text{const},$$

$$\begin{aligned}\log q(\boldsymbol{\tau},\boldsymbol{\Lambda}) = {} & \mathbb{E}_{q(\mathbf{H},\mathbf{Z},\boldsymbol{\mu})}\big[\log p(\mathbf{X}|\mathbf{H},\mathbf{Z})\big] + \log p(\boldsymbol{\tau},\boldsymbol{\Lambda}) \\ & + \mathbb{E}_{q(\boldsymbol{\mu})}\big[\log p(\boldsymbol{\mu}|\boldsymbol{\tau},\boldsymbol{\Lambda})\big] + \text{const},\end{aligned}$$

$$\begin{aligned}\log q(\mathbf{c}) = {} & \mathbb{E}_{q(\mathbf{H},\mathbf{Z},\mathbf{A},\mathbf{V})}\big[\log p(\mathbf{Y}|\mathbf{H},\mathbf{Z},\mathbf{X})\big] \\ & + \mathbb{E}_{q(\mathbf{V})}\big[\log p(\mathbf{c}|\mathbf{V})\big] + \text{const},\end{aligned}$$

$$\begin{aligned}\log q(\mathbf{A},\mathbf{V}) = {} & \mathbb{E}_{q(\mathbf{H},\mathbf{Z},\mathbf{c})}\big[\log p(\mathbf{Y}|\mathbf{H},\mathbf{Z},\mathbf{X})\big] + \log p(\mathbf{A},\mathbf{V}) \\ & + \mathbb{E}_{q(\mathbf{c})}\big[\log p(\mathbf{c}|\mathbf{V})\big] + \text{const}\,.\end{aligned}$$

As previously stated, these updates resemble posterior computations weighted by $\mathbb{E}[h_{nm}] = g_{nm}$ and $\mathbb{E}[z_{nmk}] = r_{nmk}$. Appendices A and B provide further details.

2.4.6 Posterior Predictive Distribution

Prediction with HILR is akin to that with ILR, as described in Section 2.3.6. We briefly state the conditional predictive for a component $\mathbf{h} = m$ and an activation $\mathbf{z} = k$

$$\begin{aligned}p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\hat{\mathbf{h}} = m,\hat{\mathbf{z}} = k,\mathcal{D}) &= \mathbb{E}_{q(\mathbf{A},\mathbf{c},\mathbf{V})}\big[p(\hat{\mathbf{y}}|\hat{\mathbf{x}},\mathbf{A}_m,\mathbf{c}_{mk},\mathbf{V}_m)\big] \\ &= \mathrm{T}\big(\mathbf{M}_m\hat{\mathbf{x}} + \boldsymbol{\theta}_{mk}, a_{mk}\boldsymbol{\Phi}_m, \eta_m + 1\big),\end{aligned}$$

where a_{mk} has the same structure as in ILR. Further, the weight of the k—th activation of the m—th component is computed as follows

$$\begin{aligned}p(\hat{\mathbf{x}},\hat{\mathbf{h}} = m,\hat{\mathbf{z}} = k|\mathcal{D}) \propto {} & \mathbb{E}_{q(t_m)}\big[p(\hat{\mathbf{h}} = m|t_m)\big]\,\mathbb{E}_{q(s_{mk})}\big[p(\hat{\mathbf{z}} = k|s_{mk})\big] \\ & \times \mathbb{E}_{q(\boldsymbol{\mu}_{mk},\tau_m,\boldsymbol{\Lambda}_m)}\big[p(\hat{\mathbf{x}}|\boldsymbol{\mu}_{mk},\boldsymbol{\Lambda}_m)\big] \\ = {} & \frac{\delta_m}{\delta_m + \beta_m}\prod_{l=1}^{m-1}\left(1 - \frac{\delta_l}{\delta_l + \beta_l}\right) \times \frac{\gamma_{mk}}{\gamma_{mk} + \alpha_{mk}}\prod_{l=1}^{k-1}\left(1 - \frac{\gamma_{ml}}{\gamma_{ml} + \alpha_{ml}}\right) \\ & \times \mathrm{T}\left(\boldsymbol{\mu}_{mk}, \frac{\kappa_{mk}}{1 + \kappa_{mk}}\boldsymbol{\Psi}_m, \nu_m + 1\right).\end{aligned}$$

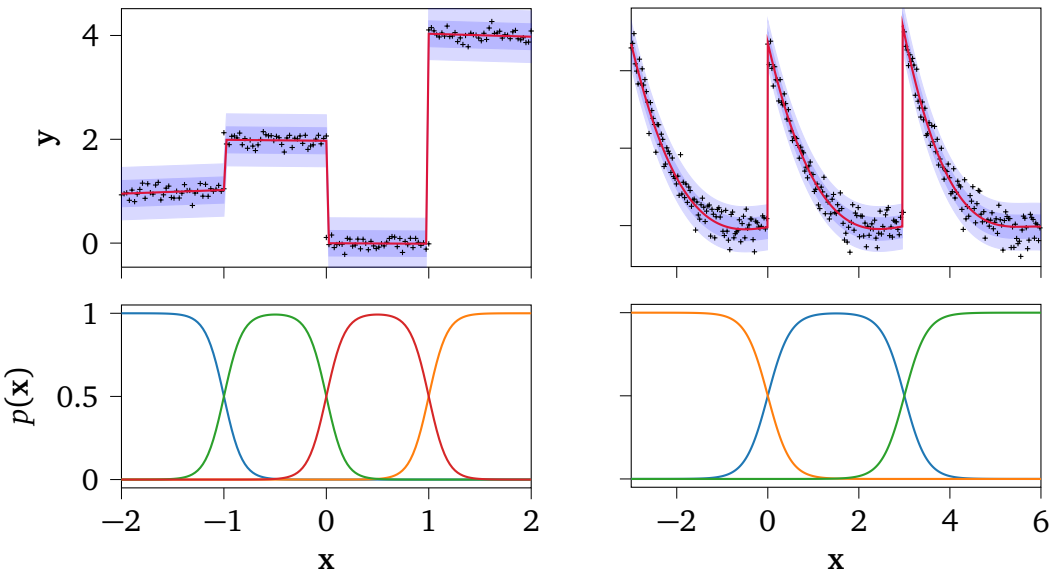


Figure 2.5: Discontinuous functions learned by ILR. The top figures show the mode prediction (red) and two standard deviations confidence (shaded blue). The left example is a simple step function that can be captured with linear features, while the on the right, we use a polynomial transformation of the input for more flexibility. The bottom plots show the activation over the input space.

2.5 Empirical Evaluation

We evaluate different aspects of the presented models on a range of tasks. Our goals are (1) to highlight some of the advantages of ILR and HILR, such as dealing with out-of-distribution predictions, recovering an input-dependent noise function, hierarchical gating, sharing parameters, and the ability to perform Bayesian sequential updates, (2) to benchmark the models on high dimensional datasets from real robots, and (3) to deploy the models in a real-world scenario to further empirically demonstrate its validity. A public open-source library is available at `https://github.com/hanyas/mimo`.

2.5.1 Out-of-distribution Uncertainty

In Figure 2.1, we apply ILR on a synthetic Sine dataset with two large gaps. We observe how the predictive uncertainty strongly reflects the lack of training data in these regions and how the mean prediction falls back to the prior values. This example highlights the reasonable quantification of uncertainty by the model. Uncertainty is low, where the mean prediction is accurate, and very high in regions where the prior dominates. The out-of-distribution behavior of ILR is strongly influenced by the discrete gating and a query's activation probability that jointly define the overall membership weights.

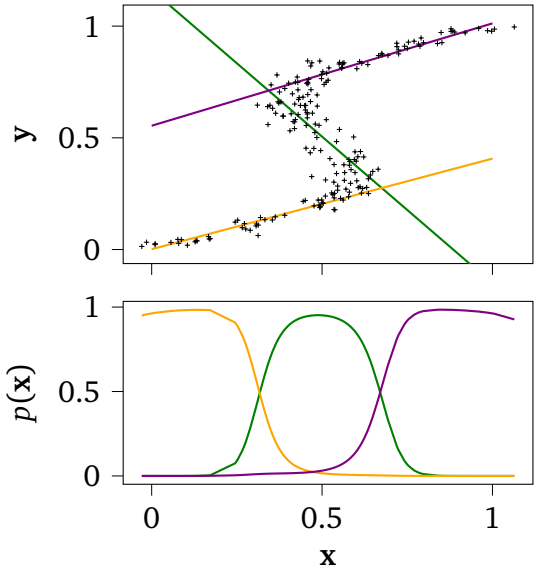


Figure 2.6: Tackling inverse mapping problems with ILR. This example includes scattered data that maps the input $\mathbf{x}$ to multiple output values $\mathbf{y}$. A discriminative modeling approach fails in these scenarios, as it tries to capture the ambiguous mean of the function $f : \mathbf{x} \rightarrow \mathbf{y}$. By approximating the joint density over both input and output, ILR can reconstruct these non-unique relations via local linear approximations.

2.5.2 Heteroscedastic Noise

We test on two different problems with input-dependent noise, the cosmic microwave background (CMB) (Bennett et al., 2003), and a synthetic dataset from a stochastic Sinc function $y(x) = \text{sinc}(x) + \epsilon$, where the noise ϵ is distributed according to zero-mean normal with a standard deviation $\sigma_\epsilon(x) = 0.05{+}0.2(1{+}\sin(2x))/(1{+}e^{-0.2x})$. Figure 2.2 and Figure 2.7 show that ILR can approximate the nonlinear functions well. In particular, the heteroscedastic noise functions are recovered in great detail.

2.5.3 Discontinuous and Local Polynomials

In Figure 2.5 (left), a step function is fitted using the mode of the predictive distribution. More expressive local regressors can be realized by applying a polynomial feature transformation to the input space. Figure 2.5 (right) depicts an example of cubic regressors, which are still linear in the parameters, fitted to data sampled from noisy cubic polynomials.

2.5.4 Inverse Mapping

One important advantage of generative over discriminative modeling is the ability to deal with non-unique inverse mapping problems. Such scenarios arise when the same input can be mapped to multiple output values. Joint modeling of the input-output data allows for flexible conditioning and alleviates the directional graph constraints. In Figure 2.6, we show a simple example of how ILR is able to learn these mappings.

2.5.5 Bayesian Sequential Updates

In Figure 2.8, we construct a sequential learning problem. Data from the Chirp function arrives in batches. ILR uses sequential Bayesian updates to iteratively update the posterior

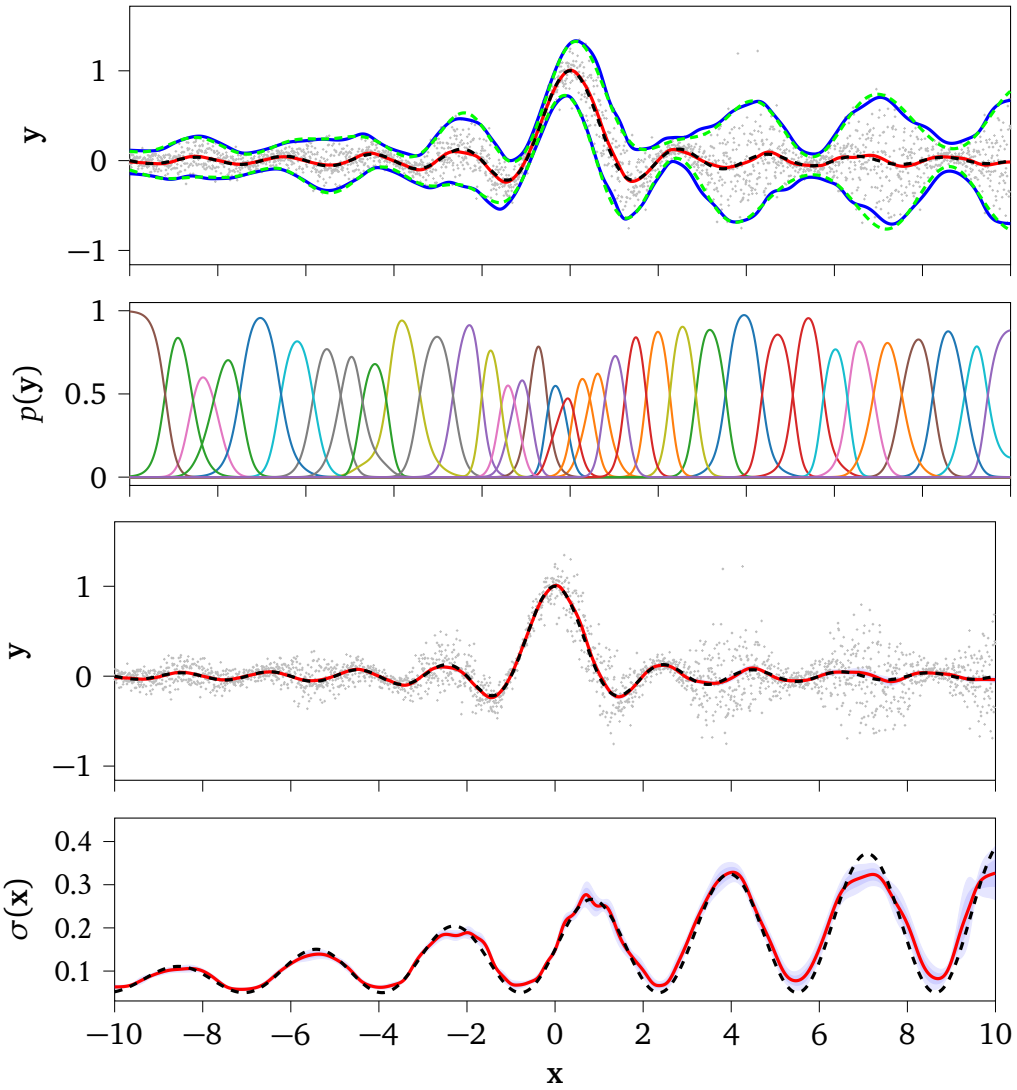


Figure 2.7: A challenging heteroscedastic example of a Sinc function heavily overlayed with input-dependent noise. The first figure shows the mean prediction (red) on the training data (dots) and the true mean function (dashed black) corrupted by noise (dashed green). The blue dashed lines represent the complex noise process recovered by ILR. The second figure shows the activation over the input space. The bottom two figures depict the results of fitting the mean and standard deviation functions averaged over ten different seeds to highlight the robustness of the inference process.

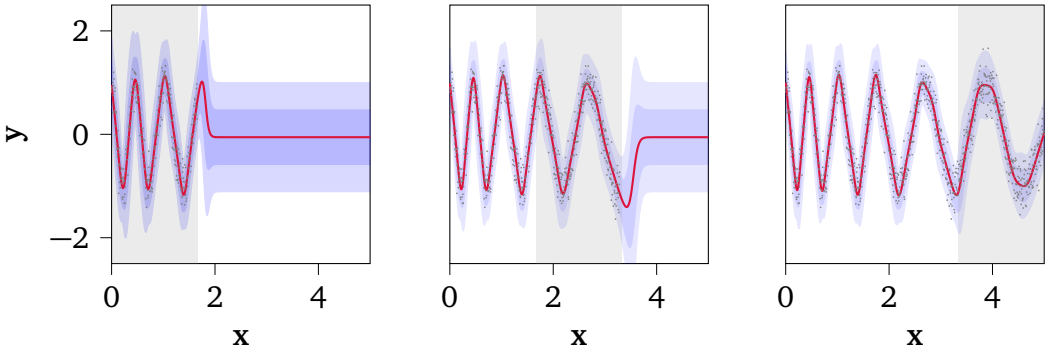


Figure 2.8: Bayesian sequential updates. Mean (red) and a two standard deviations interval (shaded blue) of the predictive distribution fitted to sequentially arriving data (three batches) from the chirp dataset (gray dots). For the second and third plots, the posterior fitted to the previous batches is used as a prior to perform a Bayesian sequential update. There is no catastrophic forgetting and in regions with no data the prediction falls back to the prior.

given a new batch. This approach successfully captures the data trend with no significant catastrophic forgetting. The mean-field posterior approximation errors have little influence because the posterior updates are localized in the input domain.

2.5.6 Hierarchical Parameter Sharing

In Figure 2.9, we test HILR's ability to share slope parameters via multi-modal activations. We consider a dataset stemming from a periodic triangle signal overlayed with additive noise. HILR decides to activate two upper- and two lower-level regions to match the structure of the data, despite having more degrees of freedom at each level.

2.5.7 Robot Inverse Dynamics

Next, we use ILR and HILR to learn the inverse dynamics of anthropomorphic manipulators. These dynamics are governed by the general mechanical equation

$$\mathbf{u} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \epsilon(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}),$$

where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ are joint angles, velocities and accelerations, and $\mathbf{u}$ are torques. $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ are the Coriolis and centripetal forces, and $\mathbf{G}(\mathbf{q})$ is the gravity force. $\epsilon(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ are general unmodelled nonlinearities such as sticktion/friction and hydraulic and tendon/cable dynamics, that motivate a data-driven approach to learn the mapping $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \rightarrow \mathbf{u}$. Later, we use the learned ILR model for online inverse dynamics control.

As evaluation criteria, we use the mean squared error (MSE), normalized mean squared error (NMSE), and the number of experts. These measures cover the prediction accuracy as well as the complexity of the learned model. We compare to popular (probabilistic) meth-

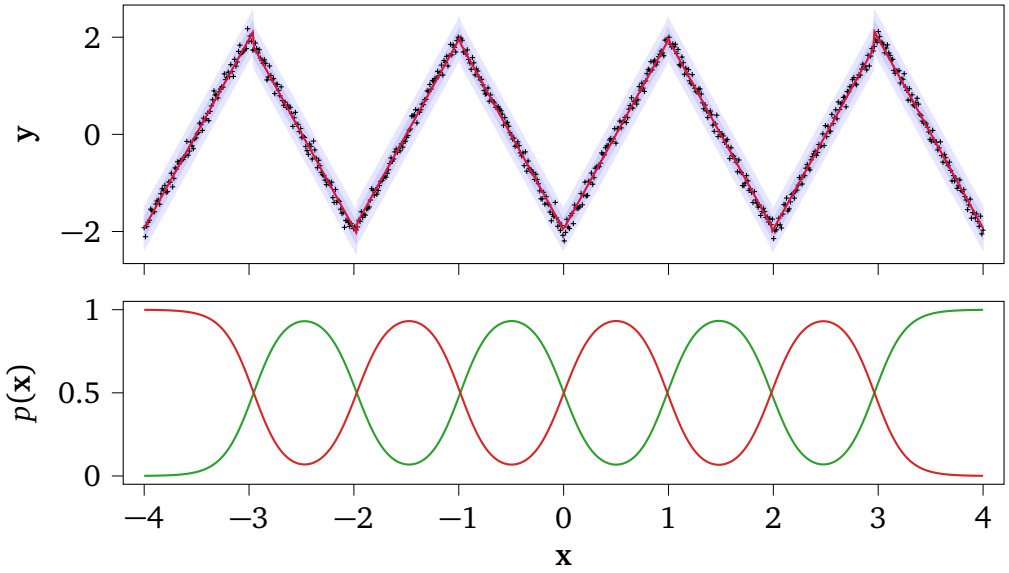


Figure 2.9: Multi-level local regression with HILR. An example of how HILR allows parameter sharing in shift-invariant functions. The top figure shows the mode prediction (red) along with two standard deviations of predictive uncertainty (shaded blue). The bottom plots highlight the multi-modal activation, which allows this representation to share slope information over non-adjacent regions.

ods such as local Gaussian regression (LGR) (Meier et al., 2014), locally weighted projection regression (LWPR) (Vijayakumar et al., 2005), Gaussian process regression (GPR) (Rasmussen & Williams, 2006) and sparse Gaussian process regression (SGPR) (Titsias, 2009), and two scalable Gaussian process product of experts: the robust Bayesian committee machine (rBCM) (Deisenroth & Ng, 2015) and the generalized product of experts (gPoE) (Cao & Fleet, 2014).

We benchmark the prediction accuracy of all regression techniques on a high-dimensional dataset collected from a 7-DoF (degrees of freedom) anthropomorphic SARCOS arm (Vijayakumar et al., 2005). The dataset consists of 44484 training points and 4449 test cases. Overall there are 21 input variables, $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$, mapping to 7 motor torques $\mathbf{u}$. We also benchmark on an inverse dynamics dataset from a 4-DoF Barrett-WAM manipulator, mapping from a 12-D to 4-D space. This dataset contains 25000 training and 5000 test pairs.

Table 2.1 and Table 2.2 list the results for both datasets. We report the average MSE, NMSE, and the number of active models over all joints. The results are obtained by running five seeds and computing the means and standard deviations for every cell in the table, except for LGR*, because of the unreasonable training times achieved while using the authors' code. When evaluating GPR on the SARCOS dataset, we faced GPU-memory constraints (32 GB), and we have discarded this evaluation. For rBCM and gPoE, we assigned

Table 2.1: Accuracy on the SARCOS dataset

	MSE	NMSE	Experts
ILR	$(4.80 \pm 0.30) \times 10^{-1}$	$(3.40 \pm 0.20) \times 10^{-3}$	1700
HILR	$(5.30 \pm 0.40) \times 10^{-1}$	$(3.90 \pm 0.30) \times 10^{-3}$	1450
LGR*	86.00×10^{-1}	50.00×10^{-3}	7000
LWPR	$(26.00 \pm 0.30) \times 10^{-1}$	$(18.00 \pm 0.20) \times 10^{-3}$	32 000
GPR	N/A	N/A	-
SGPR	$(8.50 \pm 0.03) \times 10^{-1}$	$(6.000 \pm 0.008) \times 10^{-3}$	-
rBCM	$(4.52 \pm 0.05) \times 10^{-1}$	$(2.600 \pm 0.030) \times 10^{-3}$	315
gPoE	$(4.60 \pm 0.06) \times 10^{-1}$	$(3.000 \pm 0.075) \times 10^{-3}$	315

Table 2.2: Accuracy on the Barrett-WAM dataset

	MSE	NMSE	Experts
ILR	$(2.90 \pm 0.50) \times 10^{-1}$	$(7.0 \pm 0.5) \times 10^{-3}$	1350
HILR	$(3.10 \pm 0.65) \times 10^{-1}$	$(8.0 \pm 0.6) \times 10^{-3}$	1110
LGR*	7.70×10^{-1}	17.0×10^{-3}	3270
LWPR	$(10.00 \pm 1.50) \times 10^{-1}$	$(37.0 \pm 10.0) \times 10^{-3}$	2900
GPR	$(1.00 \pm 0.01) \times 10^{-1}$	$(2.30 \pm 0.01) \times 10^{-3}$	-
SGPR	$(1.80 \pm 0.05) \times 10^{-1}$	$(6.30 \pm 0.02) \times 10^{-3}$	-
rBCM	$(3.80 \pm 0.35) \times 10^{-1}$	$(19.00 \pm 1.80) \times 10^{-3}$	100
gPoE	$(3.40 \pm 0.13) \times 10^{-1}$	$(16.00 \pm 0.60) \times 10^{-3}$	100

an expert to every 1000 data points, repeated for every output dimension.

The results show that ILR and HILR clearly outperform the related local regression methods LWPR and LGR, both in terms of prediction accuracy and number of used models. However, GPR is still the gold standard when the kernel size is within memory limits. Interestingly, the results also indicate that ILR and HILR are competitive with sparse Gaussian process regression (SGPR) and the two product of experts rBCM and gPoE. Finally, the results reveal that HILR tends to activate roughly 10-15% fewer components than ILR. This observation indicates that HILR may be taking advantage of shift-invariance patterns in the data and avoiding duplicate regression units. This hypothesis is hard to validate due to the data's high dimensionality.

2.5.8 Real Inverse Dynamics Control

Finally, we demonstrate the validity of the learned dynamics captured by ILR by using the learned model in an online trajectory tracking scenario with inverse dynamics control on the Barrett-WAM. In this experiment, we learn two separate models for two different trajectory-tracking tasks.

The first task requires tracking an 8-shaped desired trajectory in the xy-plane of the end-

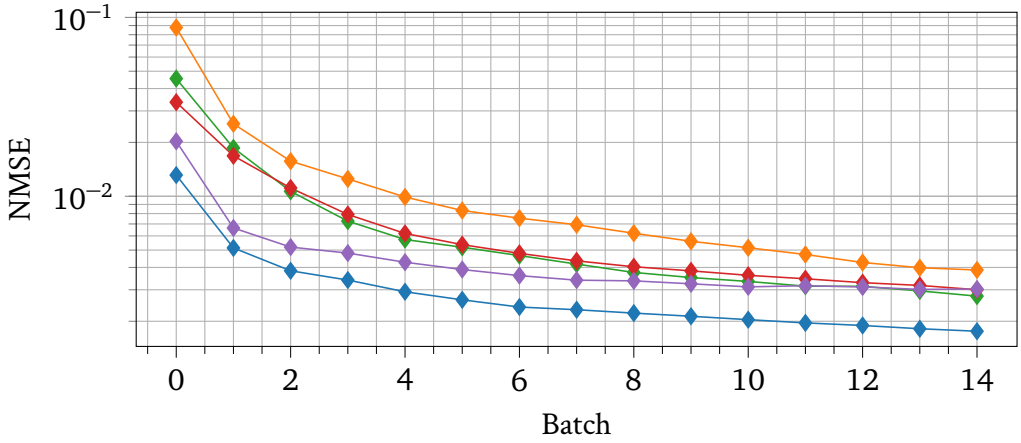


Figure 2.10: 8-Shaped trajectory learning. Bayesian sequential updates on a dataset collected from a Barrett-WAM. For five different seeds, we plot the NMSE on accumulated data over the number of batches. The NMSE consistently improves with new data and no catastrophic forgetting is observed.

Table 2.3: Tracking error and torque contributed by the PD-controller during the Barrett-WAM real robot task.

		PD	Analytic+PD	ILR+PD
T1	MSE	2.33×10^{-2}	2.16×10^{-2}	$\mathbf{1.03 \times 10^{-3}}$
	PD-Torque	8.25×10^{0}	7.12×10^{0}	$\mathbf{1.40 \times 10^{0}}$
T2	MSE	2.60×10^{-2}	2.55×10^{-2}	$\mathbf{9.17 \times 10^{-4}}$
	PD-Torque	8.71×10^{0}	7.41×10^{0}	$\mathbf{1.33 \times 10^{0}}$
T3	MSE	2.94×10^{-2}	3.08×10^{-2}	$\mathbf{8.96 \times 10^{-4}}$
	PD-Torque	9.38×10^{0}	8.06×10^{0}	$\mathbf{1.38 \times 10^{0}}$

effector. We collect 30000 training samples (roughly 1 minute) consisting of multiple trajectories with different velocity profiles. We perform learning with Bayesian sequential updates over 15 batches for multiple seeds. Figure 2.10 depicts the progression of the learning process, where the NMSE consistently improves. We then select the best model and perform online model-based control to track held-out test trajectories with unseen velocity profiles. ILR provides feed-forward torques supported by a low-gain PD-controller. We compare the tracking precision to an analytical dynamics model accompanied by the same low-gain PD-controller and a "model-free" PD-controller. Figure 2.11 shows a comparison of the different controllers on two test trajectories.

We construct a similar scenario for the second task, albeit we learn a model covering a larger region of the state-action space and compute quantitative precision benchmarks.

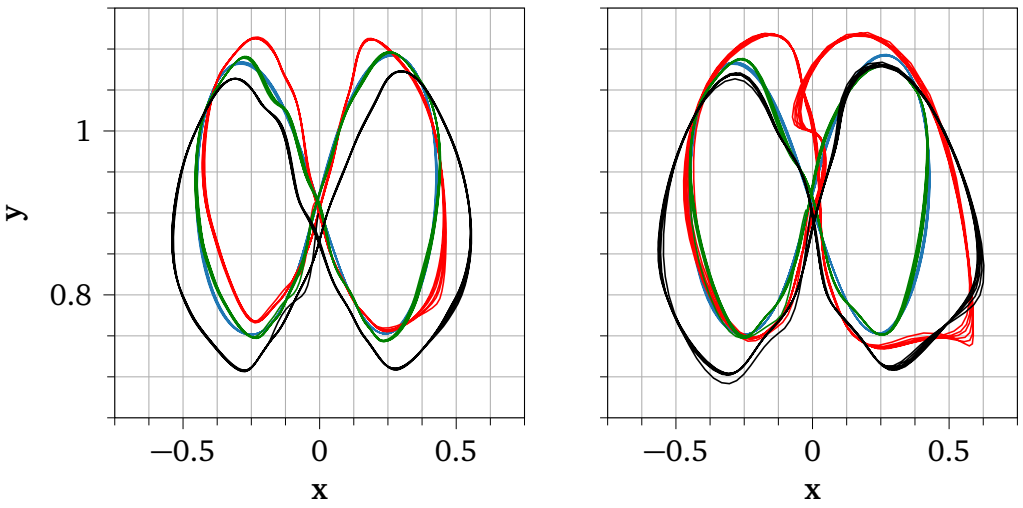


Figure 2.11: 8-shaped trajectory tracking on the Barrett-WAM. We compare three controllers on two test trajectories (blue), a low-gain PD (black), a low-gain PD + feed-forward torques from an analytical model (red), and a low-gain PD + feed-forward torques from ILR (green). The results indicate that ILR delivers the best tracking performance.

We generate a larger real-world Barrett dataset consisting of 150000 training examples (roughly 5 minutes). The movements are sinusoidal joint-space trajectories with slow and fast velocity profiles. We repeat the process of the previous task and run ILR on held-out test trajectories with the same low-gain PD-controller and compare with the analytical model and "model-free" PD. As benchmarking criteria, we evaluate the MSE with respect to the desired trajectory and the mean torque contributed by the low-gain PD-controller to the overall control signal. The rationale is as follows; A good inverse dynamics model will consistently produce a low MSE while not relying on the PD-controller's assistance in the background. Table 2.3 shows the benchmarks for 3 test trajectories. The results indicate that ILR significantly improves the performance and achieves good tracking with little contribution from the PD-controller. During both tasks, we can consistently achieve a prediction frequency of 2000 Hz, although the Barrett-WAM robot requires only 500 Hz.

2.6 Discussion

In this chapter, we presented two probabilistic hierarchical local regression models, ILR and HILR, and derived an efficient variational inference technique for data-driven learning. These representations are based on the principles of infinite mixtures and Bayesian nonparametrics. We situate our contributions as the next iteration in a large family of local linear regression techniques such as RFWR, LWPR, and LGR. We have shown that placing Dirichlet process priors on Bayesian mixtures of local regression units can regularize model complexity with minor loss in performance and without relying on heuristics.

Moreover, we have highlighted the advantages of the generative nature of these models in a set of diverse tasks. Empirical evaluation indicates that the models offer well-calibrated uncertainty quantification, outperform LWPR and LGR, and are competitive with sparse GPR and product of expertss (PoEs). Finally, we have empirically confirmed the practicality of this approach for online inverse dynamics control on a Barrett-WAM robot.

Nonetheless, these presented concepts still suffer from multiple drawbacks. The mean-field assumption is a source of significant errors in posterior inference. Collapsed formulations of Dirichlet process priors promise better approximations (Kurihara et al., 2007). In addition, Bayesian mixture models are generally affected by a large number of hyperparameters, which cannot be directly optimized via empirical Bayes (Maritz & Lwin, 1989), leading to lower predictive performance when compared to optimized GPR. Nonetheless, the evidence lower bound (ELBO) offers a tractable objective based on which the parameters may be optimized. Naive gradient-based techniques have proven to be brittle due to their reliance on Euclidean distance metrics. A natural-gradient approach appears to be a suitable alternative in the future.

Further development of hierarchical local regression may focus on treating ILR and HILR as layers in a multi-layered representation. This extension would allow the models to benefit from intermediate nonlinear projections into high dimensional spaces that have proven powerful in deep neural networks. Another practical consideration is to incorporate physical inductive biases such as inverse dynamics (Nguyen-Tuong & Peters, 2010) to facilitate learning meaningful quantities.

Chapter 3
Reinforcement Learning for Switching Systems

Optimal control of general nonlinear systems is a central challenge in automation. Enabled by powerful function approximators, data-driven approaches to control have recently successfully tackled challenging robotic applications. However, such methods often obscure the structure of dynamics and control behind black-box over-parameterized representations, thus limiting our ability to understand closed-loop behavior.

This chapter adopts a hybrid-system view of nonlinear modeling and control that lends an explicit hierarchical structure to the problem and breaks down complex dynamics into simpler localized units. We consider a sequence modeling paradigm that captures the temporal structure of the data and derive an expectation-maximization (EM) algorithm that automatically decomposes nonlinear dynamics into stochastic piecewise affine dynamical systems with nonlinear boundaries. Furthermore, we show that these time-series models naturally admit a closed-loop extension that we use to extract local polynomial feedback controllers from nonlinear experts via behavioral cloning. Finally, we introduce a novel hybrid relative entropy policy search (Hb-REPS) technique that incorporates the hierarchical nature of hybrid systems and optimizes a set of time-invariant local feedback controllers derived from a local polynomial approximation of a global value function.

3.1 Introduction

The class of nonlinear dynamical systems governs a vast range of real-world applications and underpins the most challenging problems in classical control, and reinforcement learning (RL) (Fantoni & Lozano, 2002; Kober et al., 2013). Recent developments in learning-for-control have pushed towards deploying more complex and highly sophisticated representations, e.g., (deep) neural networks and Gaussian processes, to capture the structure of both dynamics and controllers. This trend led to unprecedented success in the domain of RL (Mnih et al., 2015) and can be observed in both approximate optimal control (Deisenroth & Rasmussen, 2011; Levine et al., 2016; Hafner et al., 2019) and approximate value and policy iteration (Schulman et al., 2015; Lillicrap et al., 2015; Haarnoja et al., 2018).

However, before the latest successful revival of neural networks in control and robotics applications, research focused on different paradigms for solving difficult control tasks. One interesting concept relied on decomposing nonlinear structures of dynamics and control into simpler local (linear) components, each responsible for an area of the state-action

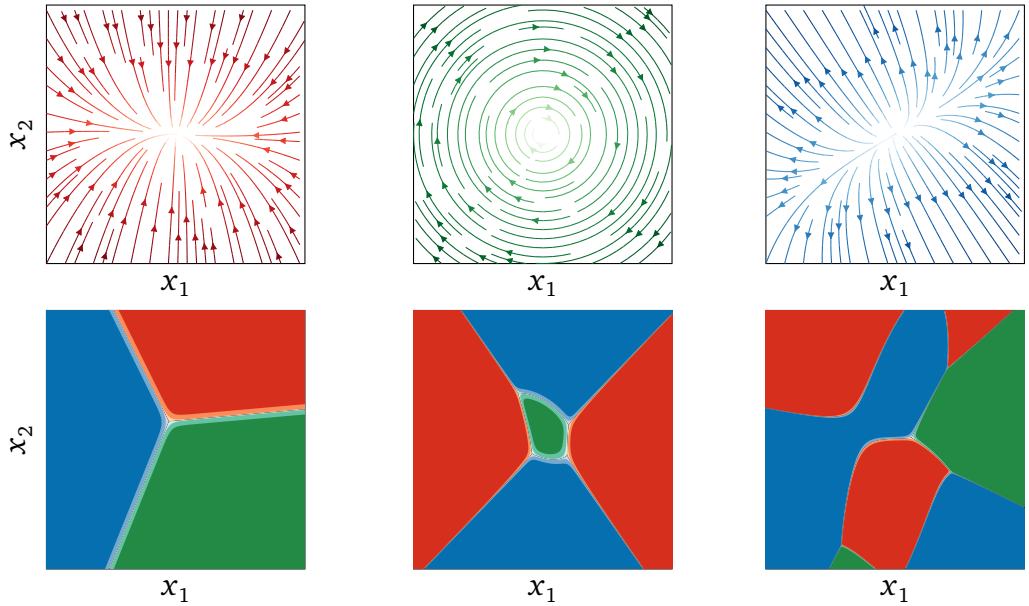


Figure 3.1: A hybrid system with $K = 3$ local linear regimes. The top row depicts the mean unforced continuous transition dynamics in the phase space. The lower row shows the probability of switching, with corresponding color, as a function of the state. We show different decision boundary models: linear (left), quadratic (middle), and third-order polynomial (right).

space. This decomposition is done to preserve interpretability and favorable mathematical properties studied over decades in classical control theory, such as local linear-quadratic assumptions (Liberzon, 2011). Instances of this abstraction can be found in the control literature under the labels of hybrid systems or switched models (Liberzon, 2003; Haddad et al., 2006; Goebel et al., 2012; Borrelli et al., 2017), while in the machine and reinforcement learning communities, the terminology of switching dynamical systems (SDS) and switching state-space models (SSM) is more widely adopted (Ghahramani & Hinton, 2000; Beal, 2003; Fox, 2009; Linderman et al., 2017).

Building on this vision, we present in this work a view of data-driven automatic system identification and learning of composite control from the perspective of hybrid systems and switching linear dynamics. We are motivated by recent in-depth analysis of piecewise linear (PWL) activation functions such as rectified linear units (ReLU) (Montufar et al., 2014; Arora et al., 2016; Pan & Srikumar, 2016; Serra et al., 2017; Petersen & Voigtlaender, 2018), which shows that such representations effectively divide the input space into linear sub-regions. This insight highlights the hierarchical structure hidden between a neural network's input and output layers and supports viewing them as approximators that rely on local experts. We take this interpretation as an impulse to follow up on ideas from optimal control (Forestier & Varaiya, 1978; Lin, 1997), and reinforcement learning (Hauskrecht

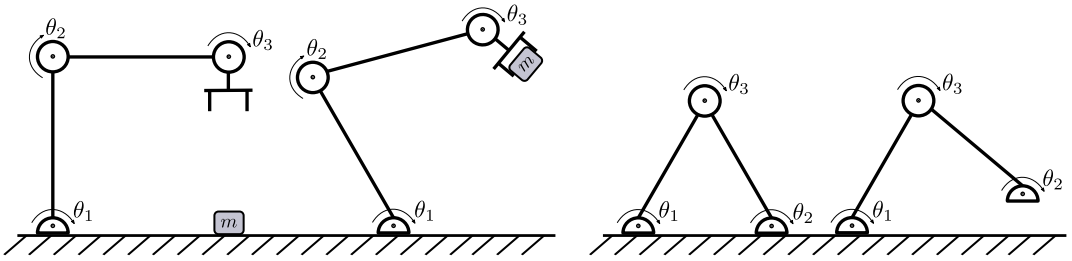


Figure 3.2: Examples of hybrid dynamical systems from the domain of robotics. Left, a manipulator executing a pick-and-place task can be modeled by 2-regime hybrid dynamics that switch between manipulator dynamics with and without the object in the end-effector. Right, the dynamics of a simplified legged robot can also be modeled by 2-regime hybrid dynamics based on the state of foot contact, which determines the possibility of actuation.

et al., 1998; Dietterich, 2000) that deviate from fully differentiable paradigms and investigate whether simpler, hybrid discrete-continuous representations may be sufficient for solving certain tasks.

Furthermore, the interest in hybrid systems as graphical models is motivated by favorable properties inherent in such representations. On the one hand, hybrid systems allow the modeling of discrete events, hard nonlinearities, and region-dependent noise. On the other hand, sequence models carry over the advantages of system identification via Bayesian inference and naturally include built-in time recurrent dynamics, which capture correlations over extended time horizons.

This chapter consolidates prior work on a hierarchical decomposition of nonlinear dynamics (Abdulsamad & Peters, 2020) and introduces a novel reinforcement learning algorithm for optimizing local polynomial policies and value functions. In the upcoming sections, we review the literature and highlight the intersection points between prominent paradigms in control and machine learning with respect to hybrid systems. Then, we introduce the notation of stochastic switching models and the infinite horizon hybrid control problem. Next, we derive a maximum a posteriori expectation-maximization (EM) algorithm for inferring the probabilistic hybrid dynamics.

We use this inference procedure in three different scenarios. First, to perform automatic decomposition of nonlinear open-loop dynamics into switching linear regimes with arbitrary boundaries. Second, to deconstruct state-of-the-art nonlinear expert controllers into simpler local polynomial policies. Finally, we embed the EM procedure into a hybrid policy search algorithm with an explicit discrete-continuous structure. We use this approach to learn hierarchical piecewise polynomial approximations of global value functions and feedback controllers. We empirically evaluate the learned models and policies on a set of numerical examples of stochastic hybrid and nonlinear systems.

3.2 Related Work

This section reviews work related to the modeling and control of hybrid systems and highlights connections and parallels between approaches stemming from the control and machine and reinforcement learning literature.

Hybrid systems have been extensively studied in the control community and are widely used in real-world/real-time applications (Borrelli et al., 2006; Menchinelli & Bemporad, 2008). For research on the topic of hybrid system identification, we refer to survey work in (Paoletti et al., 2007), and (Garulli et al., 2012). There, the authors focus on piecewise affine (PWA) systems and introduce taxonomies of different representations and procedures commonly used for identifying sub-regimes of dynamics, ranging from algebraic approaches (Vidal et al., 2003) to mixed-integer optimization (Bemporad et al., 2001), and Bayesian methods (Juloski et al., 2005). Furthermore, hybrid system identification techniques for piecewise nonlinear systems have been developed based on sparse optimization (Bako et al., 2010) and kernel methods (Lauer et al., 2010). Finally, it is worth noting that the majority of literature considers deterministic mode-switching events with exceptions in (Bemporad & Di Cairano, 2005; Cassandras & Lygeros, 2006).

Research in the area of optimal control for hybrid systems stretches back to the seminal work of (Sontag, 1981), which highlights the possibility of general nonlinear control by considering piecewise linear systems. In (Zhu & Antsaklis, 2015), an overview of control approaches for piecewise affine switching dynamics is presented. The authors categorize the literature by distinguishing between driven and un-driven systems with externally or internally forced switching mechanisms. Given the global nonlinear behavior of switched systems, the bulk of optimal control approaches in this area focus on nonlinear model predictive control (MPC) (Camacho et al., 2010). Here we highlight the influential work in (Bemporad & Morari, 1999), which formulates the optimal control problem as a mixed-integer quadratic program (MIQP). This approach was later extended in (Bemporad et al., 2000), and (Borrelli et al., 2003) to solve multi-parametric MIQP and arrive at time-variant local linear state-feedback controllers and local quadratic value functions with affine boundaries. Recently, more efficient formulations of trajectory-centric hybrid control have been proposed (Marcucci & Tedrake, 2019), which leverage modern techniques from mixed-integer and disjunctive programming and tackle large-scale problems.

Hybrid representations also play a central role in data-driven, general-purpose process modeling and state estimation (Ackerson & Fu, 1970; Hamilton, 1990), where different classes of stochastic hybrid systems serve as powerful generative models for complex dynamical behaviors (Pavlovic et al., 2001; Oh et al., 2005; Mesot & Barber, 2007). The dominant paradigm in this domain has been that of probabilistic graphical models (PGM), more specifically, hybrid dynamic Bayesian networks (HDBN) for temporal modeling (Koller et al., 2009; Lerner, 2002). However, one crucial contribution of recent Bayesian interpretations of switching systems is rooted in the Bayesian nonparametrics (BNP) view (Escobar

& West, 1995; Rasmussen, 1999; Beal et al., 2002; Teh et al., 2005). This perspective theoretically allows for an infinite number of components, thus dramatically increasing the expressiveness of such models. Given the limited scope of this review section, we highlight only recent contributions with high impacts, such as (Fox et al., 2009) and (Linderman et al., 2017), which successfully develop Markov chain Monte Carlo (MCMC) and stochastic variational inference (SVI) techniques for system identification. More recently, the rise of variational auto-encoders (Kingma & Welling, 2013) has enabled a new and powerful view on inference techniques (Becker-Ehmck et al., 2019) of hybrid systems. A distinct property of such approaches is their reliance on end-to-end differentiability and the need to relax discrete variables in order to perform inference.

In the domain of learning-for-control, the notion of switching systems is directly related to the paradigm of model-free hierarchical reinforcement learning (HRL) (Barto & Mahadevan, 2003; Parr, 1998), which combines simple representations to build complex policies. Here it is useful to differentiate between two concepts of hierarchical learning, namely *temporal* (Precup, 2000), and *state* abstractions (Andre & Russell, 2002). In their seminal work (Sutton et al., 1999, 1998), the authors build on the framework of semi-Markov decision processes (SMDP) (Bradtke & Duff, 1995) to learn activation/termination conditions of temporally extended actions (options) for solving discrete environments. Additionally, pioneering work in optimizing hierarchical control structures with temporally extended actions for robotic applications is developed in (Huber & Grupen, 1997; Huber, 2000). Further recent work has focused on different formulations of the option framework that facilitate simultaneous discovery and learning of options (Konidaris & Barto, 2009; Mankowitz et al., 2016; Daniel et al., 2016; Bacon et al., 2017; Smith et al., 2018).

However, the concept of state abstraction - the aggregation of state-action spaces into subregions, each governed by local dynamics and control - carries the most apparent parallels to the classical view of hybrid systems. In (Dietterich, 2000), a proof of convergence for RL in tabular environments with state abstraction is presented, while (Li et al., 2006) does a comprehensive study of different abstraction schemes and gives a formal definition of the problem. Furthermore, recent work has shown promising results in solving complex tasks by combining local linear policies, albeit while still leveraging a complex neural network architecture as an upper-level policy (Akrour et al., 2018).

Switching systems also serve as a powerful tool in behavioral cloning. For example, (Calinon et al., 2010) combines hidden Markov models (HMMs) with Gaussian mixture regression to represent trajectory distributions. In contrast, (Daniel et al., 2016) uses a semi-hidden Markov model (HSMM) to learn hierarchical policies, and (Burke et al., 2020) introduces switching density networks for system identification and behavioral cloning. Finally, a fully Bayesian framework for the hierarchical decomposition of policies is presented in (Sosic et al., 2017), albeit while considering known transition dynamics.

In light of the reviewed literature, we highlight the main differences that distinguish this chapter from the approaches mentioned above. First, this work leverages probabilistic hybrid dynamic networks as hierarchical representations of nonlinear open- and closed-loop behaviors. Contrary to standard piecewise autoregressive exogenous systems (PWARX), HDBN can easily integrate stochasticity and nonlinear switching boundaries, leading to more refined and less redundant segmentation of the state-action space. Furthermore, by pursuing an abstraction over the states instead of time, we circumvent the need to infer so-called termination policies, a characteristic of the option framework. Finally, the proposed hybrid policy search approach formulates a non-convex infinite horizon objective that optimizes a hierarchical local polynomial approximation of the value function. This approximation is used to derive stationary switching feedback controllers. In contrast, trajectory optimization and model predictive control techniques are often cast as sequential convex programs and optimize a fixed horizon objective that yields time-variant value functions and controllers.

3.3 Problem Statement

Consider the discrete-time optimal control problem of a stochastic nonlinear dynamical system to be defined as an infinite horizon Markov decision processes (MDP). An MDP is an abstraction of an environment defined over a state space $\boldsymbol{\mathcal{X}} \subseteq \mathbb{R}^d$ and an action space $\boldsymbol{\mathcal{U}} \subseteq \mathbb{R}^m$. The probability of a state transition from state $\mathbf{x}$ to state $\mathbf{x}'$ by applying action $\mathbf{u}$ is governed by the Markovian time-independent density function $p(\mathbf{x}'|\mathbf{x},\mathbf{u})$. The reward $r(\mathbf{x},\mathbf{u})$ is a function of the state $\mathbf{x}$ and action $\mathbf{u}$ and is discounted over time by a factor $\vartheta \in [0,1)$. The state-dependent policy $\pi(\mathbf{u}|\mathbf{x})$, from which the actions are drawn, is a density determining the probability of an action $\mathbf{u}$ given a state $\mathbf{x}$. The general objective in an infinite horizon optimal control problem is to maximize the expected cumulative sum of discounted rewards $V^\pi(\mathbf{x}) = \mathbb{E}\left[\sum_t^\infty \vartheta^t r\right]$, where V^π denotes as the state-value function under the policy π, starting from an initial state distribution $\mu_1(\mathbf{x})$.

Given the context of this work and our choice to model the system with switching linear models, we introduce to the MDP formulation a new hidden discrete variable $\mathbf{z}$, an indicator of the currently active local regime. The resulting transition dynamics can then be expressed by a factorized density function $p(\mathbf{x}',\mathbf{z}'|\mathbf{x},\mathbf{u},\mathbf{z}) = p(\mathbf{z}'|\mathbf{z},\mathbf{x},\mathbf{u})p(\mathbf{x}'|\mathbf{x},\mathbf{u},\mathbf{z}')$, which we depict as a graphical model in Figure 3.3 and discuss in further detail in the upcoming section. In the same spirit of simplification through hierarchical modeling, we employ a mixture of switching polynomial controllers $\pi(\mathbf{u}|\mathbf{x},\mathbf{z})$, associated with local polynomial value functions $V^\pi(\mathbf{x},\mathbf{z})$. The resulting framework becomes a combination of filtering to infer the active local dynamics denoted by $\mathbf{z}$ and optimal control to apply appropriate actions $\mathbf{u}$ given $\mathbf{x}$ and $\mathbf{z}$. A general closed-loop schematic is available in Figure 3.4.

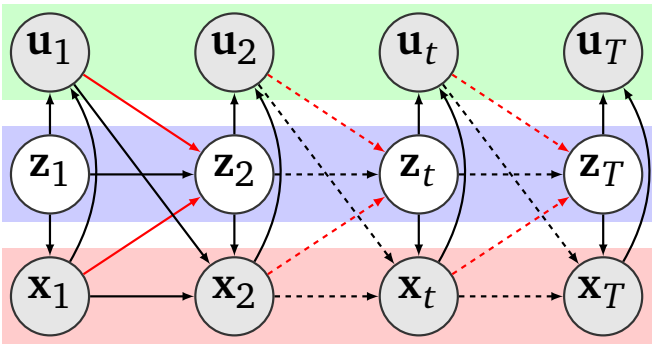


Figure 3.3: A probabilistic graphical model of recurrent autoregressive hidden Markov models (rARHMMs) extended to support hybrid controls. rARHMMs are hybrid dynamic Bayesian networks that explicitly allow the discrete state $\mathbf{z}$ to depend on the continuous variables $\mathbf{x}$ and $\mathbf{u}$, as highlighted in red.

3.4 Hybrid Dynamic Bayesian Networks

We focus on recurrent autoregressive hidden Markov models (rARHMMs) as a representation of closed-loop stochastic hybrid systems. The rARHMM is a special case of recurrent switching linear dynamical systems (rSLDS) (Linderman et al., 2017), also known as augmented SLDS (Barber, 2006). In contrast to rSLDS, an rARHMM lacks an observation model and directly describes the internal state with an additive noise process. We extend rARHMMs to support exogenous and endogenous inputs in order to simulate the open- and closed-loop behaviors of driven dynamics. Figure 3.3 depicts the corresponding graphical model, which closely resembles the linear boundary PWARX.

An rARHMM with K regions models the trajectory of a hybrid system as follows. The initial continuous state $\mathbf{x}_1 \in \mathbb{R}^d$ and continuous action $\mathbf{u}_1 \in \mathbb{R}^m$ are drawn from a pair of Gaussian and conditional Gaussian distributions, respectively. The initial discrete indicator $\mathbf{z}_1$ is a one-hot random vector modeled by a categorical density parameterized by $\boldsymbol{\varphi}$

$$\mathbf{z}_1 \sim \mathrm{Cat}(\boldsymbol{\varphi}),\ \mathbf{x}_1 \sim \mathrm{N}(\boldsymbol{\mu}_{z_1}, \boldsymbol{\Omega}_{z_1}),\ \mathbf{u}_1 \sim \mathrm{N}(\mathbf{K}_{z_1}\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\Delta}_{z_1}).$$

The transition to a state $\mathbf{x}_{t+1}$ and the actions $\mathbf{u}_t$ are modeled by linear-Gaussian dynamics

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{A}_{z_{t+1}}\mathbf{x}_t + \mathbf{B}_{z_{t+1}}\mathbf{u}_t + \mathbf{c}_{z_{t+1}} + \boldsymbol{\lambda}_t, & \boldsymbol{\lambda}_t &\sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Lambda}_{z_{t+1}}), \\ \mathbf{u}_t &= \mathbf{K}_{z_t}\boldsymbol{\phi}(\mathbf{x}_t) + \boldsymbol{\delta}_t, & \boldsymbol{\delta}_t &\sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Delta}_{z_t}), \end{aligned}$$

where $(\mathbf{A}, \mathbf{B}, \mathbf{c}, \mathbf{K}, \boldsymbol{\Omega}, \boldsymbol{\Lambda}, \boldsymbol{\Delta})$ are matrices and vectors of appropriate dimensions with respect to $\mathbf{x}$ and $\mathbf{u}$. Note that we parameterize all Gaussian distribution with precision instead of covariance matrices. $\boldsymbol{\phi}(\mathbf{x})$ are polynomial state features of arbitrary degree. The discrete transition probability $p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{x}_t, \mathbf{u}_t)$ is governed by K categorical distributions param-

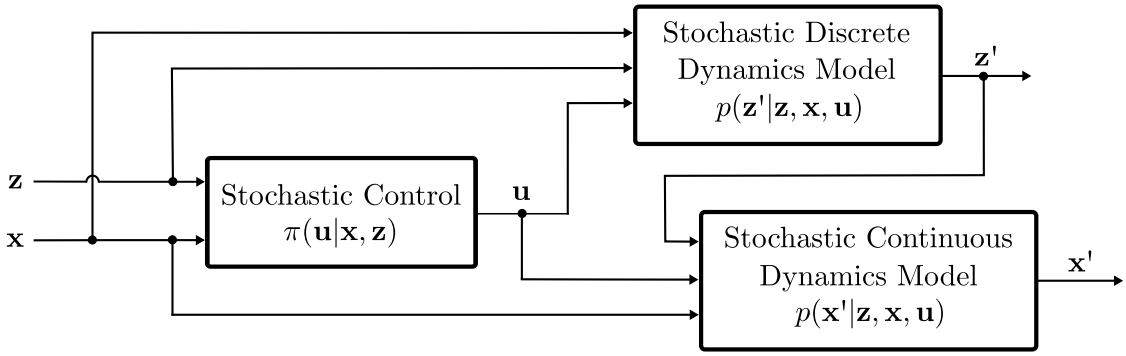


Figure 3.4: A schematic of hybrid dynamics and control. Given the state $\mathbf{x}$ and region indicator $\mathbf{z}$, a corresponding controller $\pi(\mathbf{u}|\mathbf{x},\mathbf{z})$ is selected and the action $\mathbf{u}$ is computed. The transition to a regime $\mathbf{z}'$ is determined based on the discrete dynamics model $p(\mathbf{z}'|\mathbf{z},\mathbf{x},\mathbf{u})$, and in consequence influencing the progression of the state $\mathbf{x}'$ via the appropriate continuous dynamics model $p(\mathbf{x}'|\mathbf{x},\mathbf{u},\mathbf{z}')$.

eterized by state-action dependent multi-class logistic functions

$$\chi_{ij} = p(\mathbf{z}_{t+1} = j|\mathbf{z}_t = i, \mathbf{x}_t, \mathbf{u}_t) = \frac{\exp\left(f(\mathbf{x}_t, \mathbf{u}_t; \boldsymbol{\omega}_{ij})\right)}{\sum_k \exp\left(f(\mathbf{x}_t, \mathbf{u}_t; \boldsymbol{\omega}_{ik})\right)}.$$

We abuse notation slightly by sometimes using $\mathbf{z}$ to refer to the discrete state index instead of treating it as a one-hot vector. The function f may have any type of $\mathbf{x}$ and $\mathbf{u}$ features, e.g. polynomial or neural. A set of vectors $\boldsymbol{\omega}_{ij}$ parameterize all transition combinations $i \to j \ \forall i, j \in [1, K]$. Figure 3.1 depicts realizations of different transition functions that lead to a variety of state space decompositions.

For the sake of completeness, we point out that the Markov property of an rARHMM is evident in Figure 3.3. By applying the principle of D-separation (Bishop, 2006), we conclude that the hidden discrete states $\mathbf{z}_{t-1}$ and $\mathbf{z}_{t+1}$ are conditionally independent given $\mathbf{z}_t$, more formally $\mathbf{z}_{t+1} \perp\!\!\!\perp \mathbf{z}_{t-1}|\mathbf{z}_t$.

This representation of switching dynamics has a significant advantage over other non-recurrent hybrid models (Davis, 1993; Fox et al., 2009), since it couples discrete and continuous dynamics of an HMM in both directions. This aspect has significant implications for the model's expressiveness and ability to capture the underlying dynamics of interesting physical applications, as it limits redundancies in the hierarchical decomposition of the state-action space. For example, one may consider a case in which one component can explain the local dynamics in the neighborhood of multiple non-connected discrete states. To achieve a sharp decision boundary in such scenarios, a non-recurrent model has to duplicate the continuous dynamics with a different transition probability for each set of neighbor-

ing regions. This duplication leads to redundant discrete states with the same continuous dynamics while differing in their switching behavior. One way to circumvent this explosion is to consider a hierarchical abstraction over meta regions similar to factorial hidden Markov model (FHMM) (Ghahramani & Jordan, 1997). Nonetheless, such representations may still require multiple hierarchy levels to match the expensiveness of a recurrent transition, which compactly parameterizes a continuum of transition probabilities.

At the same time, the discrete-continuous coupling introduces inter-dependencies between $\mathbf{z}$, $\mathbf{x}$ and $\mathbf{u}$, which in the case of rSLDS, make exact filtered and smoothed inference intractable (Lerner, 2002; Koller et al., 2009). This issue arises because the hidden state of an rSLDS becomes a mixture over $\mathbf{x}$ and $\mathbf{z}$, whose number of components explodes exponentially when propagated in time (Barber, 2006). Moreover, (rS)LDS are not uniquely identifiable due to rotational invariance (Barber, 2012). These two limitations have informed our decision to focus on rARHMM as a first step since they admit tractable filtering and smoothing over the hidden state $\mathbf{z}$, as the upcoming section will reveal.

The remainder of this chapter focuses on using these hybrid models in three ways. First, an open-loop setting that treats the control $\mathbf{u}$ as an exogenous input is used for automatically identifying nonlinear systems via decomposition into continuous and discrete switching dynamics. Second, a closed-loop setting that assumes the control $\mathbf{u}$ to originate from a nonlinear controller. We show that this setting can simultaneously decompose dynamics and control in an behavioral cloning scenario. Finally, we leverage the same framework in a model-based hybrid reinforcement learning algorithm to learn switching controllers of general nonlinear systems.

3.5 Inference of Switching Dynamics and Control

In this section, we sketch the outline of an expectation-maximization/Baum-Welch algorithm (Baum et al., 1970; Dempster et al., 1977; Rabiner, 1989) for inferring the parameters $\boldsymbol{\theta}$ of an rARHMM given time-series observations $(\mathbf{X}, \mathbf{U})$. The resulting algorithm can be used two-fold. First, it can be applied to perform automatic hybrid system identification to learn the open-loop dynamics of nonlinear systems given state-action observations. Second, it can clone the closed-loop behavior of a nonlinear controller and decompose it into a set of local experts.

Our developed approach is related in some aspects to the Baum-Welch algorithms proposed in (Bengio & Frasconi, 1995) and (Daniel et al., 2016). However, we introduce suitable priors over all parameters and derive a maximum a posteriori (MAP) technique with a stochastic maximization step and hyperparameter optimization. In our experience, the priors and noisy gradient estimate significantly reduce the sensitivity of EM with respect to initialization and appear to be less prone to get stuck in bad local minima - an effect well-studied in neural networks (Bottou, 1998).

Although our procedure is not fully Bayesian like methods that rely on Gibbs sampling (Fox et al., 2009; Linderman et al., 2017), it has computational and predictive advantages. On the one hand, Gibbs sampling can suffer from slow convergence in high dimensional spaces leading to an overall high computational cost (Gelman et al., 2013). On the other hand, standard Gibbs sampling-based approaches are not flexible enough to incorporate neural transition predictor functions due to their reliance on conditionally conjugate computation. Finally, a good prior specification is crucial in small data regimes since a vague prior may dominate the predictive posterior and effectively cause under-fitting. We implement a hyperparameter optimization scheme that elevates this concern by optimizing the prior parameters via empirical Bayes (Maritz & Lwin, 1989), thus attenuating the prior influence and improving the predictive performance significantly.

3.5.1 Maximum A Posteriori Optimization

Consider again the rARHMM in Figure 3.3 where the continuous state $\mathbf{x}$ and action $\mathbf{u}$ are observed variables, while the K-region indicators $\mathbf{z}$ are hidden. To infer the model parameters, we assume a dataset consisting of N state-action trajectories $\mathcal{D} = \{(\mathbf{X}^n, \mathbf{U}^n)\}_{n=1}^N$, each of length T, where $(\mathbf{X}^n, \mathbf{U}^n, \mathbf{Z}^n)$ represent an entire trajectory. The inference objective is to maximize the log-posterior probability of the observations $(\mathbf{X}^n, \mathbf{U}^n)$ conditioned on the free parameters $\boldsymbol{\theta} = \{\boldsymbol{\varphi}, \boldsymbol{\mu}_k, \boldsymbol{\Omega}_k, \mathbf{A}_k, \mathbf{B}_k, \mathbf{c}_k, \boldsymbol{\Lambda}_k, \mathbf{K}_k, \boldsymbol{\Delta}_k, \boldsymbol{\omega}_{ik}\}\ \forall i, k \in [1, K]$

$$\boldsymbol{\theta}_{\text{MAP}} := \arg\max_{\boldsymbol{\theta}} \quad \log \prod_{n=1}^{N} \sum_{\mathbf{z}^n} p(\mathbf{X}^n, \mathbf{U}^n, \mathbf{Z}^n | \boldsymbol{\theta}) p(\boldsymbol{\theta}), \tag{3.1}$$

where $p(\mathbf{X}^n, \mathbf{U}^n, \mathbf{Z}^n | \boldsymbol{\theta})$ is the likelihood of a trajectory and factorizes according to

$$\begin{aligned} p(\mathbf{X}^n, \mathbf{U}^n, \mathbf{Z}^n | \boldsymbol{\theta}) = p(\mathbf{z}_1^n) p(\mathbf{x}_1^n | \mathbf{z}_1^n) p(\mathbf{u}_1^n | \mathbf{x}_1^n, \mathbf{z}_1^n) \\ \prod_{t=2}^{T} p(\mathbf{x}_t^n | \mathbf{x}_{t-1}^n, \mathbf{u}_{t-1}^n, \mathbf{z}_t^n) p(\mathbf{u}_t^n | \mathbf{x}_t^n, \mathbf{z}_t^n) p(\mathbf{z}_t^n | \mathbf{z}_{t-1}^n, \mathbf{x}_{t-1}^n, \mathbf{u}_{t-1}^n), \end{aligned} \tag{3.2}$$

and $p(\boldsymbol{\theta} | \mathbf{h})$ is the factorized parameter prior

$$\begin{aligned} p(\boldsymbol{\theta} | \mathbf{h}) = p(\boldsymbol{\varphi}) \prod_{i=1}^{K} \prod_{k=1}^{K} p(\boldsymbol{\omega}_{ik}) \prod_{k=1}^{K} p(\boldsymbol{\mu}_k | \boldsymbol{\Omega}_k) p(\boldsymbol{\Omega}_k) \\ \times \prod_{k=1}^{K} p(\mathbf{A}_k | \boldsymbol{\Lambda}_k) p(\mathbf{B}_k | \boldsymbol{\Lambda}_k) p(\mathbf{c}_k | \boldsymbol{\Lambda}_k) p(\boldsymbol{\Lambda}_k) \\ \times \prod_{k=1}^{K} p(\mathbf{K}_k | \boldsymbol{\Delta}_k) p(\boldsymbol{\Delta}_k). \end{aligned}$$

We choose all priors to be conjugate or semi-conjugate with respect to their likelihoods, if possible. Therefore, we place a normal-Wishart (NW) prior on the initial state distribution $(\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k) \sim \mathrm{NW}(\mathbf{0}, \kappa_0, \boldsymbol{\Psi}_0, \nu_0)$, and a matrix-normal-Wishart (MNW) on the linear transition dynamics $(\mathbf{A}_k, \mathbf{B}_k, \mathbf{c}_k, \boldsymbol{\Lambda}_k) \sim \mathrm{MNW}(\mathbf{0}, \mathbf{R}_0, \boldsymbol{\Phi}_0, \rho_0)$. The initial discrete state takes a Dirichlet prior $\boldsymbol{\varphi} \sim \mathrm{Dir}(\boldsymbol{\tau}_0)$, while the logistic transition parameters are governed by a non-conjugate zero-mean Gaussian prior with diagonal precision $\boldsymbol{\omega}_{ik} \sim \mathrm{N}(\mathbf{0}, \alpha\mathbf{I})$. Finally, we place a separate matrix-normal-Wishart prior on the action likelihood $(\mathbf{K}_k, \boldsymbol{\Delta}_k) \sim \mathrm{MNW}(\mathbf{0}, \mathbf{S}_0, \boldsymbol{\Gamma}_0, \varepsilon_0)$. The quantities $(\kappa_0, \boldsymbol{\Psi}_0, \nu_0, \mathbf{R}_0, \boldsymbol{\Phi}_0, \rho_0, \boldsymbol{\tau}_0, \alpha, \mathbf{S}_0, \boldsymbol{\Gamma}_0, \varepsilon_0)$ are hyperparameters that we aggregate in the hyperparameter set $\mathbf{h}$.

The choice of priors is not restricted to these distributions. Depending on modeling assumptions, one can assume dynamics with diagonal noise matrices and pair them with gamma distribution priors. Moreover, if the system is known to have a state-independent noise process, the K Wishart and gamma priors can be *tied* across components, leading to a more structured representation.

3.5.2 Baum-Welch Expectation-Maximization

Expectation-maximization algorithms introduce a variational posterior distribution over the hidden variables $q(\mathbf{Z}^n)$ and derive a lower bound on the complete log-probability

$$\log \prod_{n=1}^{N} \sum_{\mathbf{z}^n} p(\mathcal{D}^n, \mathbf{Z}^n, \boldsymbol{\theta}) \geq \sum_{n=1}^{N} \sum_{\mathbf{z}^n} q(\mathbf{Z}^n) \log \frac{p(\mathcal{D}^n, \mathbf{Z}^n, \boldsymbol{\theta})}{q(\mathbf{Z}^n)}. \tag{3.3}$$

We can find a point estimate of the parameters $\boldsymbol{\theta}_{\mathrm{MAP}}$ by following a modified scheme of EM, alternating between an expectation step (E-step), in which the lower bound in Equation (3.3) is maximized with respect to the variational distributions $q(\mathbf{Z}^n)$ given a parameter estimate $\hat{\boldsymbol{\theta}}$, a maximization step (M-step), that updates $\boldsymbol{\theta}$ given $(\hat{q}(\mathbf{Z}^n), \hat{\mathbf{h}})$, and finally, an empirical Bayes step (EB-step) that updates $\mathbf{h}$ given $(\hat{q}(\mathbf{Z}^n), \hat{\boldsymbol{\theta}})$. A sketch of the overall iterative procedure is presented in Algorithm 3.1.

Exact Expectation Step. Maximizing the lower bound with respect to $q(\mathbf{Z}^n)$ can be determined by reformulating Equation (3.3)

$$\begin{aligned}
L &= \sum_{n=1}^{N} \sum_{\mathbf{z}^n} q(\mathbf{Z}^n) \log \frac{p(\mathbf{X}^n, \mathbf{U}^n, \mathbf{Z}^n, \boldsymbol{\theta} | \mathbf{h})}{q(\mathbf{Z}^n)} \\
&= \sum_{n=1}^{N} \sum_{\mathbf{z}^n} q(\mathbf{Z}^n) \log p(\mathbf{X}^n, \mathbf{U}^n, \boldsymbol{\theta} | \mathbf{h}) + \sum_{n=1}^{N} \sum_{\mathbf{z}^n} q(\mathbf{Z}^n) \log \frac{p(\mathbf{Z}^n | \mathbf{X}^n, \mathbf{U}^n, \boldsymbol{\theta})}{q(\mathbf{Z}^n)} \\
&= \sum_{n=1}^{N} \log p(\mathbf{X}^n, \mathbf{U}^n, \boldsymbol{\theta} | \mathbf{h}) - \mathrm{KL}(q(\mathbf{Z}^n) \,||\, p(\mathbf{Z}^n | \mathbf{X}^n, \mathbf{U}^n, \boldsymbol{\theta})).
\end{aligned}$$

Algorithm 3.1: Expectation-Maximization for rARHMM

input: $K, \mathbf{X}, \mathbf{U}, \mathbf{h}$

initialize: $\hat{\boldsymbol{\theta}} \sim p(\boldsymbol{\theta}|\mathbf{h}), \hat{\mathbf{h}} \leftarrow \mathbf{h}$

1 **while** $\log p(\mathbf{X}, \mathbf{U}, \boldsymbol{\theta}|\mathbf{h})$ not converged **do**

`// Expectation step`

2 **for** $n \leftarrow 1$ **to** N **do**

3 $\boldsymbol{\alpha}^n, \boldsymbol{\beta}^n \leftarrow$ **ForwardBackward**$(\mathbf{X}^n, \mathbf{U}^n, \hat{\boldsymbol{\theta}})$

4 $\boldsymbol{\gamma}^n, \boldsymbol{\xi}^n \leftarrow$ **SmoothedPosteriors**$(\boldsymbol{\alpha}^n, \boldsymbol{\beta}^n, \hat{\boldsymbol{\theta}})$

`// Maximization step`

5 $\hat{\boldsymbol{\theta}} \leftarrow$ **Maximize** $Q(\hat{\boldsymbol{\theta}}, \boldsymbol{\gamma}, \boldsymbol{\xi}, \hat{\mathbf{h}})$

`// Empirical Bayes`

6 $\hat{\mathbf{h}} \leftarrow \hat{\mathbf{h}} + \varrho \, \nabla_{\mathbf{h}} Q \,|_{\mathbf{h}=\hat{\mathbf{h}}}$

output: $\hat{\boldsymbol{\theta}}$

This form of the lower bound implies that the optimal variational distribution $\hat{q}(\mathbf{Z}^n)$ minimizes the Kullback-Leibler divergence (KL), meaning

$$\hat{q}(\mathbf{Z}^n) = p(\mathbf{Z}^n|\mathbf{X}^n, \mathbf{U}^n, \boldsymbol{\theta}) = p(\mathbf{Z}^n|\mathbf{x}^n_{1:T}, \mathbf{u}^n_{1:T}, \boldsymbol{\theta}). \tag{3.4}$$

This update tightens the bound if the posterior model $\hat{q}(\mathbf{Z}^n)$ belongs to the same family of the true posterior (Beal, 2003). Notice that the E-step is independent of the prior $p(\boldsymbol{\theta})$. Moreover, Equation (3.4) indicates that the E-step reduces to the computation of smoothed marginals $p(\mathbf{z}^n_t|\mathbf{x}^n_{1:T}, \mathbf{u}^n_{1:T}, \hat{\boldsymbol{\theta}})$ under the current parameter estimate $\hat{\boldsymbol{\theta}}$. Following (Baum et al., 1970) and (Murphy, 2012), we derive a forward-backward algorithm, which enables closed-form exact inference of these quantities

$$\boldsymbol{\gamma}^n_t(k) = p(\mathbf{z}^n_t = k|\mathbf{x}^n_{1:T}, \mathbf{u}^n_{1:T}) \propto p(\mathbf{z}^n_t = k|\mathbf{x}^n_{1:t}, \mathbf{u}^n_{1:t}) p(\mathbf{x}^n_{t+1:T}, \mathbf{u}^n_{t+1:T}|\mathbf{z}^n_t = k, \mathbf{x}^n_t, \mathbf{u}^n_t),$$

where $\alpha^n_t(k) = p(\mathbf{z}^n_t = k|\mathbf{x}^n_{1:t}, \mathbf{u}^n_{1:t})$ is the forward message that computes the filtered marginals via a forward recursion

$$\begin{aligned}\alpha^n_t(k) \propto\, & p(\mathbf{x}^n_t|\mathbf{x}^n_{t-1}, \mathbf{u}^n_{t-1}, \mathbf{z}^n_t = k) p(\mathbf{u}^n_t|\mathbf{x}^n_t, \mathbf{z}^n_t = k) \\ & \times \sum_{j=1}^{K} p(\mathbf{z}^n_t = k|\mathbf{z}^n_{t-1} = j, \mathbf{x}^n_{t-1}, \mathbf{u}^n_{t-1}) \alpha^n_{t-1}(j),\end{aligned}$$

and $\beta_t^n(k) = p(\mathbf{x}_{t+1:T}^n | \mathbf{z}_t^n = k, \mathbf{x}_t^n, \mathbf{u}_t^n)$ is the backward message that performs smoothing by computing the conditional likelihood of future evidence

$$\begin{aligned}\beta_t^n(k) = \sum_{j=1}^{K} &\beta_{t+1}^n(j) p(\mathbf{z}_{t+1}^n = j | \mathbf{z}_t^n = k, \mathbf{x}_t^n, \mathbf{u}_t^n) \\ &\times p(\mathbf{x}_{t+1}^n | \mathbf{x}_t^n, \mathbf{u}_t^n, \mathbf{z}_{t+1}^n = j) p(\mathbf{u}_{t+1}^n | \mathbf{x}_{t+1}^n, \mathbf{z}_{t+1}^n = j).\end{aligned}$$

Additionally, by combining both forward and backward messages, we can compute the two-slice smoothed marginals $p(\mathbf{z}_t^n, \mathbf{z}_{t+1}^n | \mathbf{x}_{1:T}^n, \mathbf{u}_{1:T}^n, \hat{\boldsymbol{\theta}})$ which will be useful during the maximization and empirical Bayes steps

$$\begin{aligned}\xi_{t,t+1}^n(i,j) &= p(\mathbf{z}_t^n = i, \mathbf{z}_{t+1}^n = j | \mathbf{x}_{1:T}^n, \mathbf{u}_{1:T}^n) \\ &\propto p(\mathbf{x}_{t+1}^n | \mathbf{x}_t^n, \mathbf{u}_t^n, \mathbf{z}_{t+1}^n = j) p(\mathbf{u}_{t+1}^n | \mathbf{x}_{t+1}^n, \mathbf{z}_{t+1}^n = j) \\ &\quad \times \alpha_t^n(i) p(\mathbf{z}_{t+1}^n = j | \mathbf{z}_t^n = i, \mathbf{x}_t^n, \mathbf{u}_t^n) \beta_{t+1}^n(j).\end{aligned}$$

This concludes all needed computations for the forward-backward messages of the E-step.

Stochastic Maximization Step. After performing the E-step and computing the smoothed posteriors, we are able to evaluate the lower bound and maximize it with respect to $\boldsymbol{\theta}$ given $(\hat{q}(\mathbf{Z}^n), \hat{\mathbf{h}})$. By plugging Equation (3.4) into (3.3) and leveraging conditional independence, we arrive at the complete log-probability function

$$\begin{aligned}Q &= \sum_{n=1}^{N} \sum_{\mathbf{z}^n} \hat{q}(\mathbf{Z}^n) \log p(\mathbf{X}^n, \mathbf{U}^n, \mathbf{Z}^n, \boldsymbol{\theta} | \hat{\mathbf{h}}) \\ &= \log p(\boldsymbol{\theta} | \hat{\mathbf{h}}) + \sum_{k=1}^{K} \sum_{n=1}^{N} \gamma_1^n \Big[\log \varphi_k + \log \mathrm{N}(\mathbf{x}_1^n | \boldsymbol{\mu}_k, \boldsymbol{\Omega}_k) \Big] \\ &\quad + \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{t=2}^{T} \gamma_t^n \log \mathrm{N}(\mathbf{x}_t^n | \mathbf{A}_k \mathbf{x}_{t-1}^n + \mathbf{B}_k \mathbf{u}_{t-1}^n + \mathbf{c}_k, \boldsymbol{\Lambda}_k) \\ &\quad + \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{t=1}^{T} \gamma_t^n \log \mathrm{N}(\mathbf{u}_t^n | \mathbf{K}_k \phi(\mathbf{x}_{t-1}^n), \boldsymbol{\Delta}_k) \\ &\quad + \sum_{i=1}^{K} \sum_{j=1}^{K} \sum_{n=1}^{N} \sum_{t=2}^{T} \xi_{t-1,t}^n \log \chi_{ij}(\mathbf{x}_{t-1}^n, \mathbf{u}_{t-1}^n, \boldsymbol{\omega}_{ij}).\end{aligned}$$

The optimization of Q is commonly done via coordinate ascent. Simpler models, e.g., Gaussian- and Binomial-HMMs, lead to an exact, convex M-step with closed-form optimality conditions. This is not the case in rARHMM, given the possibility of choosing nonlinear transition predictor functions. Such a choice leads to an approximate, non-convex

M-step that requires gradient-based updates. In this case, stochastic optimization is recommended (Robbins & Monro, 1951) as batched noisy gradient estimates allow algorithms to escape shallow local minima and reduce the computational cost that comes with evaluating the gradients for all data instances. When implementing the M-step, we apply stochastic optimization on the transition parameters $\boldsymbol{\omega}$. We use a stochastic gradient ascent direction with an adaptive learning rate ε and batch size M (Robbins & Monro, 1951)

$$\boldsymbol{\omega}^{(l+1)} = \boldsymbol{\omega}^{l} + \frac{\varepsilon}{M} \sum_{m=1}^{M} \nabla_{\boldsymbol{\omega}} Q_m \mid_{\boldsymbol{\omega}=\boldsymbol{\omega}^{l}},$$

$$\nabla_{\boldsymbol{\omega}} Q_m = \nabla_{\boldsymbol{\omega}} \left[\log p(\boldsymbol{\omega}|\alpha) + \sum_{i=1}^{K} \sum_{j=1}^{K} \xi_m \log \chi_{ij}(\mathbf{x}_m, \mathbf{u}_m, \boldsymbol{\omega}_{ij}) \right].$$

For the parameters with conjugate priors, we derive closed-form optimality conditions. Effectively, this part of the optimization constitutes formulating the posterior distribution and taking the mode of each posterior density for a point estimate update. By considering only relevant terms, we write the optimization of the initial gating parameter $\boldsymbol{\varphi}$ as

$$\max_{\boldsymbol{\varphi}} \quad \log \mathrm{Dir}(\boldsymbol{\varphi}|\hat{\boldsymbol{\tau}}_0) + \sum_{k=1}^{K} \sum_{n=1}^{N} \gamma_1^n \log \varphi_k,$$

while the objective of the initial parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k)$ can be decoupled for each k

$$\max_{\boldsymbol{\mu},\boldsymbol{\Omega}} \quad \log \mathrm{NW}(\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k|(\mathbf{0}, \hat{\kappa}_0, \hat{\boldsymbol{\Psi}}_0, \hat{\nu}_0)_k) + \sum_{n=1}^{N} \gamma_1^n \log \mathrm{N}(\mathbf{x}_1^n|\boldsymbol{\mu}_k, \boldsymbol{\Omega}_k).$$

Analogously, the objective terms related to the dynamics parameter $(\mathbf{A}_k, \mathbf{B}_k, \mathbf{c}_k, \boldsymbol{\Lambda}_k)$ are also decoupled to k parts

$$\begin{aligned} \max_{\mathbf{A},\mathbf{B},\mathbf{c},\boldsymbol{\Lambda}} \quad & \log \mathrm{MNW}(\mathbf{A}_k, \mathbf{B}_k, \mathbf{c}_k, \boldsymbol{\Lambda}_k|(\mathbf{0}, \hat{\mathbf{R}}_0, \hat{\boldsymbol{\Phi}}_0, \hat{\nu}_0)_k) \\ & + \sum_{n=1}^{N} \sum_{t=2}^{T} \gamma_t^n \log \mathrm{N}(\mathbf{x}_t^n|\mathbf{A}_k \mathbf{x}_{t-1}^n + \mathbf{B}_k \mathbf{u}_{t-1}^n + \mathbf{c}_k, \boldsymbol{\Lambda}_k), \end{aligned}$$

and, finally, to learn closed-loop behavior, we infer the controller parameters $(\mathbf{K}_k, \boldsymbol{\Delta}_k)$

$$\begin{aligned} \max_{\mathbf{K},\boldsymbol{\Delta}} \quad & \log \mathrm{MNW}(\mathbf{K}_k, \boldsymbol{\Delta}_k|(\mathbf{0}, \hat{\mathbf{S}}_0, \hat{\boldsymbol{\Gamma}}_0, \hat{\varepsilon}_0)_k) \\ & + \sum_{n=1}^{N} \sum_{t=1}^{T} \gamma_t^n \log \mathrm{N}(\mathbf{u}_t^n|\mathbf{K}_k \phi(\mathbf{x}_t^n), \boldsymbol{\Delta}_k). \end{aligned}$$

We refrain from stating the explicit solution for these former problems. Instead, we provide a general outline of how to compute these posteriors and their modes in Appendix A.

Approximate Empirical Bayes. Inference techniques that leverage data-independent assumptions run the risk of prior miss-specification. In our MAP approach, the priors are weakly informative and carry little information. Their main purpose is to regularize greedy updates that might lead to premature convergence. However, when there is little data, the priors, especially those on the precision matrices, may dominate the posterior probability, leading to over-regularization and under-fitting of the objective. See (Gelman, 2006) for a discussion on the choice of weakly-informative and non-informative precision priors.

Empirical Bayes approaches remedy this issue by integrating out the parameters $\boldsymbol{\theta}$ and optimizing the marginal likelihood with respect to the hyperparameters $\mathbf{h}$ (Maritz & Lwin, 1989). In our setting, marginalizing all hidden quantities does not admit a closed-form formula. Instead, we interleave the E- and M-steps with hyperparameter updates that optimize the lower bound given an estimate of parameters $\hat{\boldsymbol{\theta}}$ and an adaptive step size ϱ

$$\mathbf{h}^{(l+1)} = \mathbf{h}^{(l)} + \varrho \, \nabla_{\mathbf{h}} Q \,|_{\mathbf{h}=\mathbf{h}^{l}}, \quad \text{where} \quad \nabla_{\mathbf{h}} Q = \nabla_{\mathbf{h}} \log p(\hat{\boldsymbol{\theta}} | \mathbf{h}).$$

3.6 Reinforcement Learning for Hybrid Systems

In our review of related work in Section 3.2, we highlight that many successful classical hybrid control algorithms are often limited to computationally expensive trajectory-centric control policies, e.g., model predictive control. This type of controller is disadvantageous in applications that require fast reactive feedback control with broad coverage over the state-action space. Another common drawback of classical hybrid control approaches is their reliance on linear separation boundaries of the local dynamics. On the other hand, we mentioned several RL-based approaches that learn global time-invariant policies. Nonetheless, these algorithms are exclusively model-free and mostly rely on time abstraction and temporally extended actions, which require learning task-specific termination policies, in addition to the local controllers.

In this section, we present a stochastic infinite horizon sample-based optimization technique that leverages the structure and properties of hybrid systems under the paradigm of state abstraction. Our algorithm extends the step-based formulation of relative entropy policy search (REPS) (Peters et al., 2010; Van Hoof et al., 2015; Belousov & Peters, 2017) by introducing a discrete state variable $\mathbf{z}$ and taking into account the structure of hybrid dynamics. Our approach, hybrid REPS (Hb-REPS), leverages the state-action-dependent nonlinear switches $p(\mathbf{z}'|\mathbf{z}, \mathbf{x}, \mathbf{u})$ as a task-independent upper-level coordinator to a mixture of K lower-level stationary polynomial policies $\pi(\mathbf{u}|\mathbf{x}, \mathbf{z})$. While the proposed approach shares many features with (Daniel et al., 2016), our formulation relies on a state-abstraction representation of hybrid systems and embeds the hierarchical model structure into the op-

timization problem in order to learn a hierarchy over the global value function. In contrast, (Daniel et al., 2016) operates in the framework of semi-Markov decision processes and optimizes a mixture over termination and feedback policies without considering the existence of a hierarchical structure in the space of dynamics and value functions.

3.6.1 Infinite-Horizon Stochastic Hybrid Control

In the REPS framework an optimal control problem is presented as an iterative trust-region optimization for a discounted average-reward objective under a stationary state-action distribution $\pi(\mathbf{u}|\mathbf{x},\mathbf{z})\mu(\mathbf{x},\mathbf{z})$, Equation (3.5a). The trust-region is formulated as a KL (Kullback & Leibler, 1951), Equation (3.5c). Its purpose is to bound the information loss between iterations. The REPS formulation explicitly incorporates a dynamics consistency constraint, Equation (3.5b), that describes how the stochastic state of the system evolves. The following describes the optimization solved during a single iteration of what we refer to as hybrid REPS

$$\underset{\pi,\mu}{\text{maximize}} \quad J = \sum_{\mathbf{z}} \iint r(\mathbf{x},\mathbf{u})\pi(\mathbf{u}|\mathbf{x},\mathbf{z})\mu(\mathbf{x},\mathbf{z})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}, \tag{3.5a}$$

$$\text{subject to} \quad \mu(\mathbf{x}',\mathbf{z}') = (1-\vartheta)\mu_1(\mathbf{x}',\mathbf{z}') \tag{3.5b}$$

$$+\vartheta \sum_{\mathbf{z}} \iint \pi(\mathbf{u}|\mathbf{x},\mathbf{z})\mu(\mathbf{x},\mathbf{z})p(\mathbf{x}',\mathbf{z}'|\mathbf{x},\mathbf{u},\mathbf{z})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x},$$

$$\epsilon \geq \mathrm{KL}(\pi(\mathbf{u}|\mathbf{x},\mathbf{z})\mu(\mathbf{x},\mathbf{z}) \,||\, q(\mathbf{x},\mathbf{u},\mathbf{z})), \tag{3.5c}$$

$$1 = \sum_{\mathbf{z}} \iint \pi(\mathbf{u}|\mathbf{x},\mathbf{z})\mu(\mathbf{x},\mathbf{z})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}, \tag{3.5d}$$

where $\mu(\mathbf{x},\mathbf{z})$ is the stationary mixture distribution, $q(\mathbf{x},\mathbf{u},\mathbf{z})$ is the trust-region reference distribution, and the constraint in Equation (3.5d) guarantees the normalization of the state-action distribution. The factor $1-\vartheta$ is the probability of an infinite process to reset to an initial distribution $\mu_1(\mathbf{x},\mathbf{z})$. The notion of resetting is necessary to ensure ergodicity and allows the interpretation of ϑ as a discount factor, and regularization of the MDP (Puterman, 2014; Belousov & Peters, 2017).

3.6.2 Optimality Conditions and Dual Optimization

Let $p(\mathbf{x},\mathbf{u},\mathbf{z}) = \pi(\mathbf{u}|\mathbf{x},\mathbf{z})\mu(\mathbf{x},\mathbf{z})$. Using the method of Lagrangian multipliers (Boyd & Vandenberghe, 2004), we can solve the constrained primal problem with respect to state-action distribution $p(\mathbf{x},\mathbf{u},\mathbf{z})$

$$p^*(\mathbf{x},\mathbf{u},\mathbf{z}) \propto q(\mathbf{x},\mathbf{u},\mathbf{z}) \exp\left[\frac{A(\mathbf{x},\mathbf{u},\mathbf{z})}{\eta}\right] \tag{3.6}$$

where η is the Lagrangian variable associated with Equation (3.5c), and $A(\mathbf{x}, \mathbf{u}, \mathbf{z})$ the advantage function given as

$$\begin{aligned} A(\mathbf{x}, \mathbf{u}, \mathbf{z}) &= r(\mathbf{x}, \mathbf{u}) + (1 - \vartheta) \sum_{\mathbf{z}'} \int V(\mathbf{x}', \mathbf{z}') \mu_1(\mathbf{x}', \mathbf{z}') \, \mathrm{d}\mathbf{x}' \\ &\quad + \vartheta \sum_{\mathbf{z}'} \int V(\mathbf{x}', \mathbf{z}') p(\mathbf{x}', \mathbf{z}' | \mathbf{x}, \mathbf{u}, \mathbf{z}) \, \mathrm{d}\mathbf{x}' - V(\mathbf{x}, \mathbf{z}) \\ &= Q(\mathbf{x}, \mathbf{u}, \mathbf{z}) - V(\mathbf{x}, \mathbf{z}). \end{aligned} \tag{3.7}$$

The functions $V(\mathbf{x}, \mathbf{z})$ and $Q(\mathbf{x}, \mathbf{u}, \mathbf{z})$ are the state- and state-action value functions, respectively. The function $V(\mathbf{x}, \mathbf{z})$ appears naturally in REPS as the Lagrangian function associated with Equation (3.5b). The full Lagnrangian of the primal can be found in Appendix C. By substituting the optimal parameter p^* back into the Lagrangian and factorizing $q(\mathbf{x}, \mathbf{u}, \mathbf{z})$, we arrive at the dual function G as a function of the remaining Lagrangian variables η and V

$$G = \eta \epsilon + \eta \log \iint q(\mathbf{x}, \mathbf{u}) \sum_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}, \mathbf{u}) \exp \left[\frac{A(\mathbf{x}, \mathbf{u}, \mathbf{z}, V)}{\eta} \right] \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x},$$

where $q(\mathbf{z}|\mathbf{x}, \mathbf{u})$ is the posterior over $\mathbf{z}$ given the observations $\mathbf{x}$ and $\mathbf{u}$. In Section 3.5, we derive a forward-backward algorithm for inferring these probabilities, allowing us to compute the expectation over $\mathbf{z}$. The expectations over $\mathbf{X}$ and $\mathbf{U}$ are analytically intractable, therefore, we approximate them given samples from the reference distribution $q(\mathbf{x}, \mathbf{u})$. The multipliers η and V are obtained by numerically minimizing the dual $G(\eta, V)$

$$\underset{\eta, V}{\text{minimize}} \quad G(\eta, V), \qquad \text{subject to} \quad \eta \geq 0,$$

that acts as the upper bound on the primal objective.

3.6.3 Stationarity of State Distribution Mixtures

The dynamics Equation (3.5b), which ensures the stationarity of $\mu(\mathbf{x}, \mathbf{z})$, uncovers an interesting aspect of our optimization problem. A careful inspection of that integral equation reveals that a mixture distribution $\mu(\mathbf{x}, \mathbf{z})$ propagated through the mixture dynamics $p(\mathbf{x}', \mathbf{z}' | \mathbf{x}, \mathbf{u}, \mathbf{z})$ results in a mixture $\mu(\mathbf{x}', \mathbf{z}')$ that keeps growing with every pass, leading to an explosion in the number of components of the joint state distribution. This problem highlights a crucial computational issue of trajectory-based optimal control approaches of stochastic hybrid systems, as it becomes expensive to maintain the full mixture state distribution $\mu(\mathbf{x}, \mathbf{z})$ after several time steps. Common solutions to this issue usually involve falling back to crude Gaussian mixture reduction techniques (Crouse et al., 2011) that inadvertently sacrifice information and blur the distribution as it progresses in time.

This consideration has motivated us to formulate the optimal control problem as one that focuses on finding a stationary solution for $\mu(\mathbf{x}, \mathbf{z})$. We hypothesize that including Equation (3.5b) as a constraint leads to an optimal mixture policy $\pi(\mathbf{u}|\mathbf{x}, \mathbf{z})$ that acts as a *filter* and *dampens* certain modes of $\mu(\mathbf{x}', \mathbf{z}')$ with little contribution to the average-reward objective. However, we recognize that a more in-depth analysis of the nature of this integral equation is necessary.

3.6.4 Modeling Dynamics and State-Value Function

Up to this point, our derivation has been generic. We have made no assumptions on initial distributions $\mu_1(\mathbf{x}, \mathbf{z})$, the dynamics $p(\mathbf{x}', \mathbf{z}'|\mathbf{x}, \mathbf{u}, \mathbf{z})$, or the value function $V(\mathbf{x}, \mathbf{z})$. Now we introduce the local Gaussian linear dynamics and logistic switching described in Section 3.4 and assume these representations to be available in parametric form as a result of a separate learning process. Furthermore, we model the state-value function with local n-th degree polynomial functions $V(\mathbf{x}, \mathbf{z}) = \boldsymbol{\omega}_{\mathbf{z}}^{\top} \phi_{\mathbf{z}}(\mathbf{x})$, where $\phi_{\mathbf{z}}(\mathbf{x})$ is the state-feature vector which contains all polynomial features of the state $\mathbf{x}$, and $\boldsymbol{\omega}_{\mathbf{z}}$ is the parameter vector assigned to the different regions.

Under these assumptions, we can leverage the available joint density $\mu_1(\mathbf{x}, \mathbf{z})$ and $p(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \mathbf{z})$ to compute the necessary expectations in Equation (3.7)

$$\mathbb{E}_{\mathbf{x}',\mathbf{z}'}\left[V(\mathbf{x}', \mathbf{z}')\right] = \sum_{\mathbf{z}'} \int V(\mathbf{x}', \mathbf{z}') p(\mathbf{x}', \mathbf{z}'|\mathbf{x}, \mathbf{u}, \mathbf{z}) \, \mathrm{d}\mathbf{x}',$$

$$\mathbb{E}_{\mathbf{x}_1,\mathbf{z}_1}\left[V(\mathbf{x}', \mathbf{z}')\right] = \sum_{\mathbf{z}'} \int V(\mathbf{x}', \mathbf{z}') \mu_1(\mathbf{x}', \mathbf{z}') \, \mathrm{d}\mathbf{x}'.$$

This computation allows our approach to capture the stochasticity of the dynamics and delivers an estimate of the advantage function $A(\mathbf{x}, \mathbf{u}, \mathbf{z})$ instead of the temporal difference (TD) error in the general REPS framework (Peters et al., 2010). Ultimately, this leads to better estimates of the expected discounted future returns.

Practically, these integrals can be either naively approximated by applying Monte Carlo integration (Robert et al., 1999), or, more efficiently, by leveraging the structure of the integrand $V(\mathbf{x}', \mathbf{z}')$, and using Gauss-Hermite cubature rules for exact integration over polynomial functions (Särkkä, 2013).

3.6.5 Maximum-A-Posteriori Policy Improvement

Another important advantage of this model-based approach becomes evident when considering the policy improvement step in the REPS framework. The policy update is incorporated into the optimality condition of the stationary state-action distribution $p(\mathbf{x}, \mathbf{u}, \mathbf{z}) = \pi(\mathbf{u}|\mathbf{x}, \mathbf{z})\mu(\mathbf{x}, \mathbf{z})$ in Equation (3.6). As a consequence, updating the mixture policies $\pi(\mathbf{u}|\mathbf{x}, \mathbf{z})$ requires the computation of state probabilities $\mu(\mathbf{x}, \mathbf{z})$, which in turn require knowledge of

Algorithm 3.2: Model-Based Relative Entropy Policy Search for Hybrid Systems

input: $p^0(\mathbf{x}', \mathbf{z}' | \mathbf{x}, \mathbf{u}, \mathbf{z})$
initialize: $q(\mathbf{u}|\mathbf{x}, \mathbf{z})$, $\boldsymbol{\omega}_\mathbf{z}$, η
while J not converged **do**
    `// Sample interactions`
    $(\mathbf{X}, \mathbf{U}) \leftarrow$ **Environment**(q, p^0)
    `// Policy evaluation`
    $\eta^*, \boldsymbol{\omega}_\mathbf{z}^*, \mathbf{w}^* \leftarrow$ **Minimize** $G(\mathbf{X}, \mathbf{U}, p^0, \eta, \boldsymbol{\omega}_\mathbf{z}, \epsilon)$
    `// Policy improvement`
    $\pi^*(\mathbf{u}|\mathbf{x}, \mathbf{z}) \leftarrow$ **BaumWelch**$(\mathbf{X}, \mathbf{U}, p^0, \mathbf{w}^*)$
    `// Update parameters`
    $q, \boldsymbol{\omega}_\mathbf{z}, \eta \leftarrow \pi^*, \boldsymbol{\omega}_\mathbf{z}^*, \eta^*$
output: $\pi^*(\mathbf{u}|\mathbf{x}, \mathbf{z})$

the dynamics model. This issue is circumvented in other model-free realizations of REPS by introducing a crude approximation to enable a model-free policy update nonetheless. In (Deisenroth et al., 2013), the authors postulate that the distribution $\mu(\mathbf{x}, \mathbf{z})$ is usually "close enough" to $q(\mathbf{x}, \mathbf{z})$, thus allowing the ratio $q(\mathbf{x}, \mathbf{z})/\mu(\mathbf{x}, \mathbf{z})$ to be ignored when a weighted maximum-likelihood fit of the actions $\mathbf{u}$ is performed to update π.

While the assumption of "closeness" may be practical and empowers many successful flavors of REPS, it is crucial to be aware of its technical ramifications, as it undermines the primary motivation of a relative entropy bound on the state-action distribution in Equation (3.5c). This aspect is unique in the REPS framework when compared to other state-of-the-art approximate policy iteration algorithms (Schulman et al., 2015, 2017; Haarnoja et al., 2018), that optimize a similar objective, albeit with a relaxed bound that only limits the change of the action distribution π.

In contrast, our proposed algorithm leverages the modeled continuous-discrete dynamics and updates the policy $\pi(\mathbf{u}|\mathbf{x}, \mathbf{z})$ with the correct weighting. The optimality condition in Equation (3.6) is satisfied by deriving a weighted maximum a posteriori estimate based on the state-action distribution $p(\mathbf{x}, \mathbf{z}, \mathbf{u})$, and implicitly updating $\pi(\mathbf{u}|\mathbf{x}, \mathbf{z})$. This procedure is equivalent to a modified Baum-Welch expectation-maximization algorithm that fits a closed-loop rARHMM, as derived in Section 3.5. The difference is that the EM objective

in Equation (3.1) has to be augmented with the importance weights from Equation (3.6)

$$\begin{aligned} &\arg\max_{\boldsymbol{\theta}} \log \prod_{n=1}^{N} \sum_{\mathbf{z}^n} \mathbf{w}^n p(\mathbf{X}^n, \mathbf{U}^n, \mathbf{Z}^n | \boldsymbol{\theta}) p(\boldsymbol{\theta}), \\ &\text{where} \quad \mathbf{w}^n = \exp\left[A(\mathbf{X}^n, \mathbf{U}^n, \mathbf{Z}^n)/\eta\right]. \end{aligned}$$

This augmentation leads to weighted M- and EB-steps while the E-step is not altered.

Note that during the policy improvement step, we can either assume an a priori estimate of the open-loop dynamics $p^0(\mathbf{x}', \mathbf{z}' | \mathbf{x}, \mathbf{u}, \mathbf{z})$ and only update the control parameters corresponding to the conditional $\pi(\mathbf{u}|\mathbf{x}, \mathbf{z})$, or we can continuously update $p(\mathbf{x}', \mathbf{z}' | \mathbf{x}, \mathbf{u}, \mathbf{z})$ as more data iteratively becomes available. A compact sketch of the overall optimization process is available in Algorithm 3.2.

3.7 Empirical Evaluation

In this section, we benchmark different aspects related to the inference of rARHMMs and the learning of hybrid controllers via Hb-REPS. First, we assess the predictive performance of rARHMMs in an open-loop setting to validate the choice of this representation of hybrid dynamics. Second, we test the inference on closed-loop rARHMMs and their ability to capture and decompose an expert nonlinear controller in a behavioral cloning scenario. Finally, we use rARHMMs in the proposed RL algorithm Hb-REPS to solve the infinite horizon stochastic control objective and optimize piecewise local polynomial controllers and value functions. Here we make no claim to the absolute efficiency of our approach when compared to other state-of-the-art algorithms. Instead, we aim to provide an empirical proof-of-concept that supports further research into sample-based hybrid system optimization as a framework for structured nonlinear control.

3.7.1 Hybrid System Identification Examples

We empirically benchmark rARHMMs on nonlinear systems. We aim to quantify the quality of learned open-loop models and their ability to capture the underlying dynamics. For this purpose, we set up a direct comparison to popular representations for dynamics in a *long-horizon* and *limited-data* setting. A public code base is available on `https://github.com/hanyas/sds`.

In this evaluation, we focus on rARHMMs with exogenous inputs. We learn the dynamics of three simulated deterministic systems; a bouncing ball, an actuation-constrained pendulum, and a cart-pole system. We compare the predictive accuracy of rARHMMs to classical non-recurrent autoregressive hidden Markov models (ARHMMs) (Fox, 2009), feed-forward neural nets (FNNs), Gaussian processs (GPs), long-short-term memory networks (LSTMs) (Hochreiter & Schmidhuber, 1997), and recurrent neural networks (RNNs).

During the evaluation, we collected segregated training and test datasets. The training dataset is randomly split into 24 groups, each containing a subset of trajectories, and used to train different instances of all representations. These instances are then tested on the test dataset. All neural models have two hidden layers, which we test for a variety of different layer sizes, $S = \{16, 32, 64, 128, 256, 512\}$ for FNNs, $S = \{16, 32, 64, 128, 256\}$ for RNNs, and $S = \{16, 32, 64, 128\}$ for LSTMs. In the case of (r)ARHMMs, we test for different numbers of component K, dependent on the task. As a metric, we evaluate the normalized mean squared error (NMSE), averaged over the 24 data splits for a range of horizons. During evaluation, we comb through the test trajectories step by step and predict the given horizon. Moreover, in Table 3.1, we qualitatively compare the complexity of all representations in terms of their total number of parameters.

Bouncing Ball This example is a canonical instance of a dual-regime hybrid system due to the hard velocity switch at the moment of impact. We simulate the dynamics with a frequency of 20 Hz and collect 25 training trajectories with different initial heights and velocities, each 30 s long. This dataset is split 24 folds with ten trajectories, 10×150 data points, in each subset. The test dataset consists of 5 trajectories, each 30 s long. We evaluate the NMSE for horizons $h = \{1, 20, 40, 60, 80\}$ time steps. We did not evaluate a GP model in this setting due to the long prediction horizons that led to a very high computational burden. The (r)ARHMMs are tested for $K = 2$. The logistic link function of an rARHMM is parameterized by a neural net with one hidden layer containing 16 neurons. The results in Figure 3.5 show that the rARHMM approximates the dynamics well and outperforms both ARHMMs and the neural models.

Pendulum and Cart-Pole These systems are classical benchmarks from the control literature. Here we consider two different observation models, one in the wrapped joint space, where the angle space $\theta \in [-\pi, \pi]$ includes a sharp discontinuity, and a second model with smooth observations parameterized with the Cartesian trigonometric features $\{\cos(\theta), \sin(\theta)\}$. Both dynamics are simulated with a frequency of 100 Hz. We collect 25 training trajectories starting from different initial conditions and applying random uniform explorative actions. Each trajectory is 2.5 s long. The 24 splits consist of 10 trajectories each, 10×250 data points. The test dataset consists of 5 trajectories, each 2.5 s long. Forecasting accuracy is evaluated for horizons $h = \{1, 5, 10, 15, 20, 25\}$. The (r)ARHMMs are tested for $K = \{3, 5, 7, 9\}$ on both tasks. The logistic link function of the rARHMM is parameterized by a neural net with one hidden layer containing 24 neurons. As shown in Figure 3.5, the forecast evaluation provides empirical evidence for the representation power of rARHMMs in both smooth and discontinuous state spaces. FNNs and GPs perform equally well in the smooth Cartesian observation space and struggle in the discontinuous space, similar to RNNs and LSTMs.

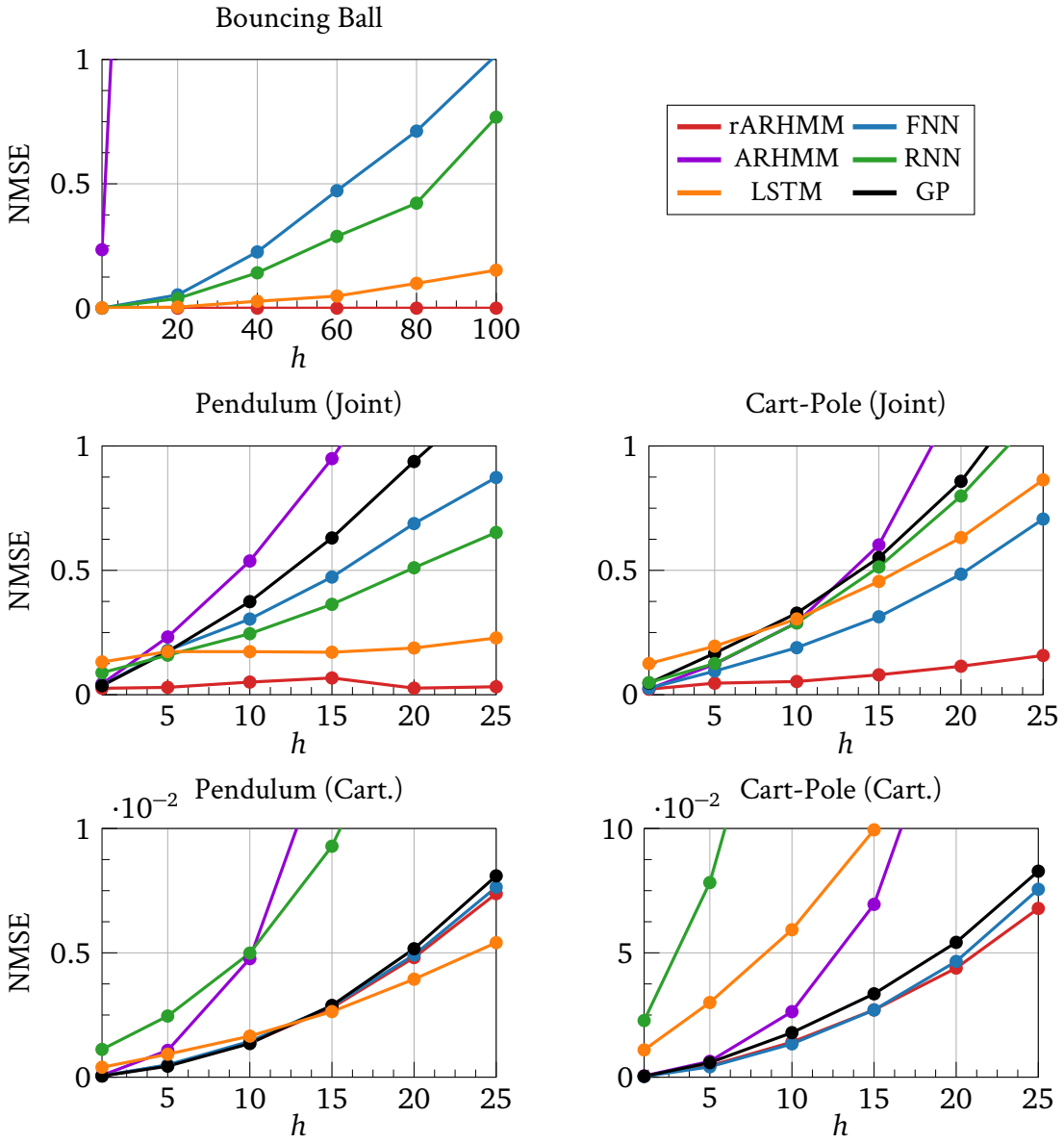


Figure 3.5: System identification: comparing the h-step NMSE of rARHMMs to other dynamics approximation models. Every evaluation point is averaged over 24 data splits. Benchmarking on three dynamical systems, a bouncing ball, a pendulum, and a cart-pole. In limited-data scenario, rARHMMs exhibit the most consistent approximation capabilities.

	Bouncing Ball	Pendulum (Joint)	Pendulum (Cartesian)	Cart-Pole (Joint)	Cart-Pole (Cartesian)
ARHMM	22 (2)	180 (9)	130 (5)	287 (7)	275 (5)
rARHMM	86 (2)	468 (9)	582 (9)	575 (7)	711 (7)
FNN	1250 (32)	546 (64)	1315 (32)	1380 (32)	1445 (32)
RNN	12866 (64)	50306 (128)	3427 (32)	50820 (128)	51077 (128)
LSTM	200450 (128)	51074 (64)	51395 (64)	201732 (128)	202373 (128)

Table 3.1: System identification: a qualitative comparison of model complexity for the best performing representations in Figure 3.5. The values reflect the total number of parameters of each model. The values in parentheses represent the hidden layer sizes S of the neural models and the number of discrete components K for the (r)ARHMM, respectively.

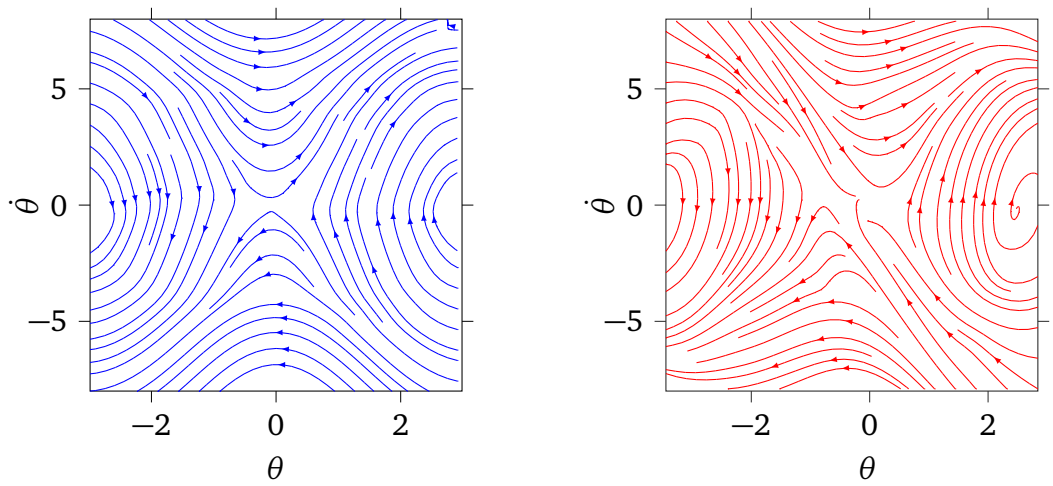


Figure 3.6: Behavioral cloning: phase space of the pendulum. The identified unforced dynamics is on the left (blue). The learned model qualitatively captures the phase portrait. On the right (red) are the closed-loop dynamics. The learned stationary hybrid policy with five regions successfully imitates a global nonlinear SAC controller to stabilize the system around the origin.

3.7.2 Hierarchical Closed-Loop Behavioral Cloning

Before applying the RL method proposed in Section 3.6.1, we first want to analyze the closed-loop rARHMM with endogenous inputs as a behavioral cloning framework. The task is to reproduce the closed-loop behavior of expert policies on challenging nonlinear systems. For this purpose, we train two different feedback experts on the pendulum and cart-pole. The two environments are simulated at 50 Hz and are influenced by static Gaussian noise with a standard deviation $\sigma = 1 \times 10^{-2}$. The experts are two-layer neural nets with 4545 parameters (pendulum) and 17537 parameters (cart-pole), optimized with the soft actor-critic (SAC) algorithm (Haarnoja et al., 2018).

For cloning, we construct two 5-regime rARHMMs with local polynomial policies of the third order. The hybrid controllers have a total number of parameters of 100 (pendulum) and 280 (cart-pole). Learning is realized on a dataset of 25 trajectories, each 5 s long, for each environment and using the EM technique from Section 3.5. The decomposed controllers complete the task of swinging up and stabilizing both systems with over 95% success rate. Figure 3.6 shows the phase portraits of the unforced dynamics and closed-loop control identified during cloning. Figure 3.7 depicts sampled trajectories of the hybrid policies highlighting the switching behavior.

3.7.3 Reinforcement Learning for Hybrid Systems

Finally, we evaluate the qualitative performance of the proposed hybrid policy search algorithm Hb-REPS on two nonlinear stochastic dynamical systems: an underpowered pendu-

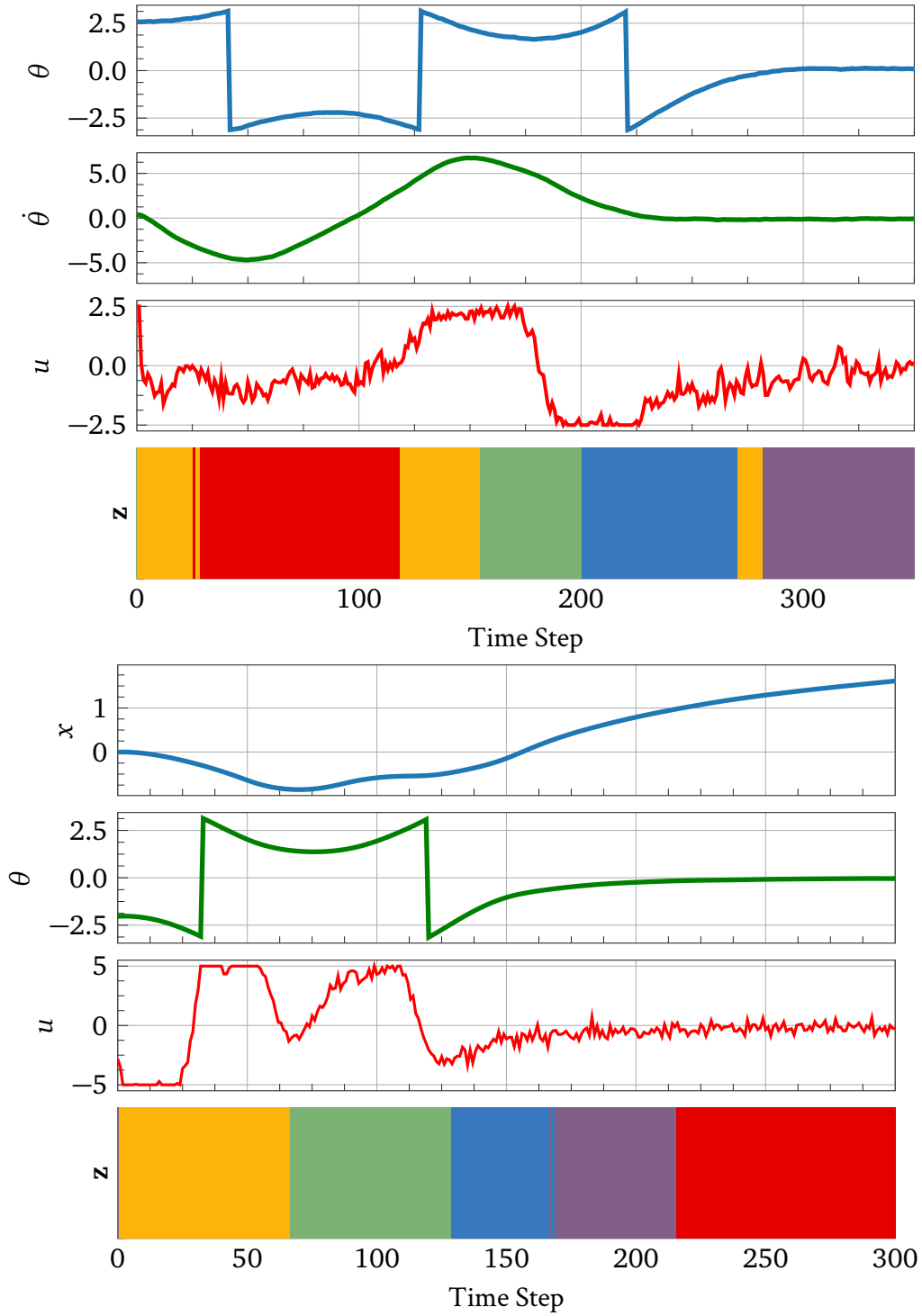


Figure 3.7: Behavioral cloning: sample trajectories from the learned hybrid policies on the pendulum (top) and cart-pole (bottom) environments. Both hybrid controllers are able to consistently solve both tasks while relying on simple local representations of the feedback controllers. The colors indicate the active dynamics and control regimes over time.

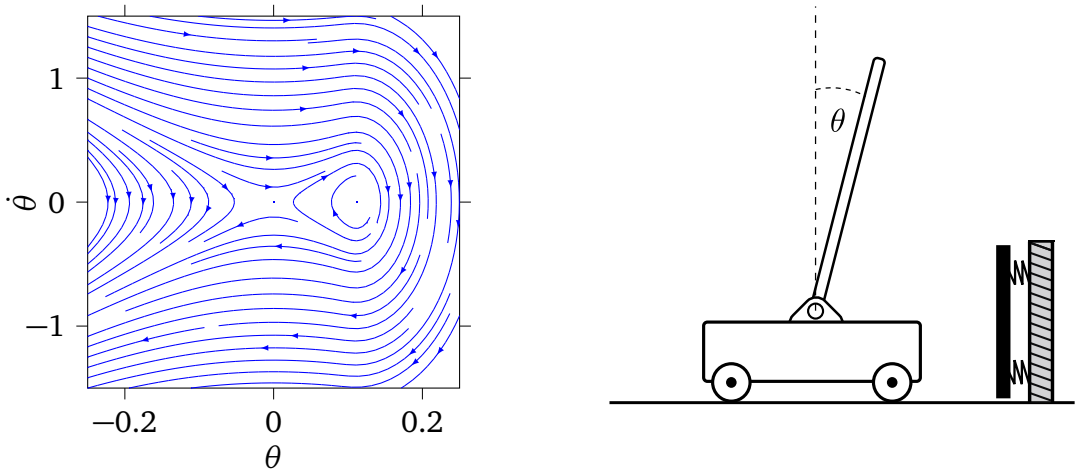


Figure 3.8: Cart-pole with an elastic wall: a hybrid system with two linear regimes. The cart-pole dynamics is linearized around the upright pole position, and the wall is elastic and modeled by spring dynamics. The switching boundary is linear. The unforced dynamics is depicted on the left (blue), and the aim is to stabilize the pole around the origin.

lum swing-up and a cart-pole stabilization task that explicitly simulates an abrupt switch in dynamics when the cart hits an elastic wall.

We compare the performance of Hb-REPS to two baselines. The first is a *vanilla* version of REPS that does not maintain any hierarchical structure and uses nonlinear function approximators with random Fourier features (RFFs) (Rahimi & Recht, 2008) to represent both policy and value function. The second baseline assumes a hierarchical policy structure and a nonlinear value function with Fourier features. This baseline is akin to what is implemented in (Daniel et al., 2016), albeit with a hierarchy based on state abstraction rather than time. We will refer to this algorithm version as hierarchical REPS (Hy-REPS). We assume an offline learning phase in which the hybrid models are learned from pre-collected data.

Pendulum Swing-up. In this experiment, the power-limited pendulum is simulated at 50 Hz and perturbed by Gaussian noise with a standard deviation $\sigma = 1 \times 10^{-2}$. The REPS agent relies on a policy and value function with 50 and 75 Fourier features, respectively. Hy-REPS assumes a similar form of the value function but with five third-order polynomial policies. Hb-REPS represents both policy and value function with five third-order polynomials. Empirical results in Figure 3.9 feature comparable learning performance of all algorithms over ten random seeds. Every iteration involves 5000 interactions with the environment. We provide a phase portrait of the closed-loop behavior for a qualitative assessment of the final stationary hybrid policy.

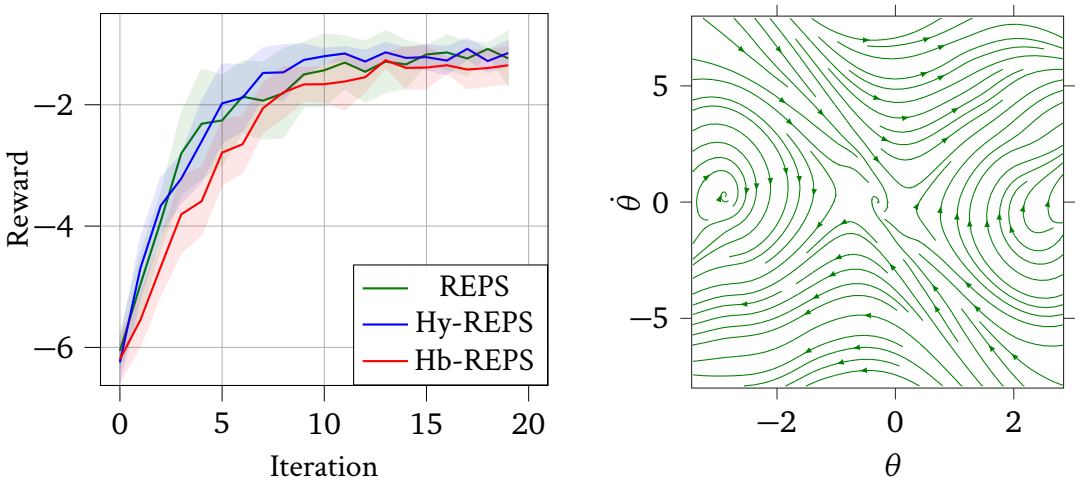


Figure 3.9: Reinforcement learning: REPS, Hy-REPS and Hb-REPS evaluated on the pendulum swing-up task. The learning curves, mean reward with two standard deviations, show that all algorithms perform equally well in terms of transient and final performance. However, Hb-REPS relies on simpler polynomial models of the policy and value function, while Hy-REPS and REPS rely on nonlinear representations. The phase portraits depict the closed-loop behavior achieved by Hb-REPS. Hb-REPS solves the task and stabilizes the pendulum.

Cart-pole Stabilization. This evaluation features a cart-pole constrained by an elastic wall modeled as a spring, Figure 3.8. The dynamics is linearized around the upright position, naturally resulting in a two-regime hybrid system. The environment is simulated at 100 Hz and perturbed by Gaussian noise with a standard deviation $\sigma = 1 \times 10^{-4}$. The REPS policy and value function both use 25 random Fourier features. Hy-REPS adopts the same value function structure with two affine linear policies. Hb-REPS also assumes two affine linear policies combined with two second-order local value functions. The results in Figures 3.10 depict matching learning performance of the three approaches over 10 random seeds. Every iteration involves 2500 interactions with the environment.

3.8 Discussion

We presented a data-driven view of hybrid system identification and control that serves as an alternative paradigm to common popular techniques. Our approach is not restricted to the class of explicit hybrid dynamics and can be seen as a general approach to structured identification and nonlinear control. We argue that this structure often exists under the hood of complex neural representation. Therefore, making it explicit may offer an avenue to apply Occam's razor and regularize over-parameterized representations. Initial empirical results support this motivation. The proposed hybrid reinforcement learning technique can do without neural networks and instead rely on a hierarchy of local polynomial repre-

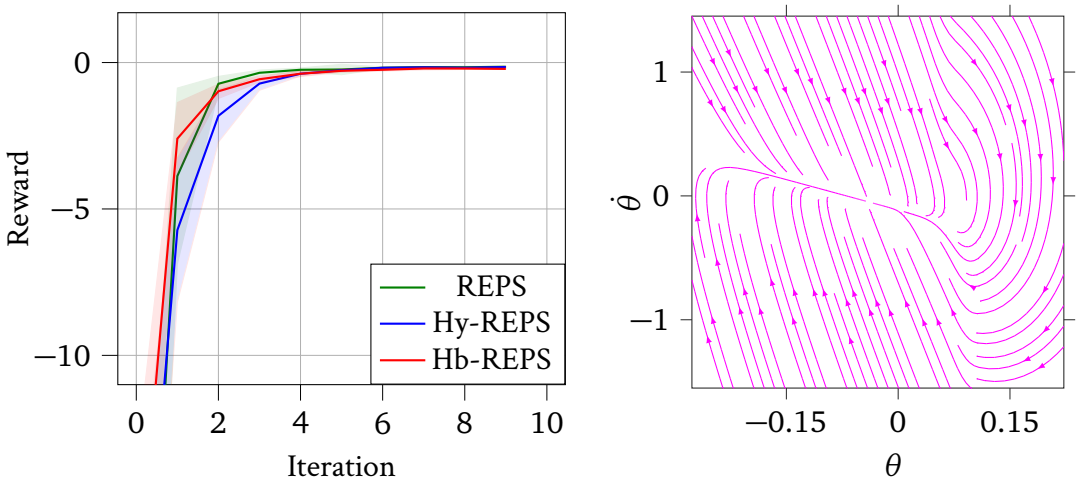


Figure 3.10: Reinforcement learning: REPS, Hy-REPS and Hb-REPS evaluated on the cart-pole stabilization task. By inspecting the learning curves, mean reward with two standard deviations, we conclude that all algorithms perform equally well. However, Hb-REPS relies on simpler polynomial models of the policy and value function, while Hy-REPS and REPS rely on nonlinear representations. The phase portraits depict the closed-loop behavior achieved by Hb-REPS.

sentations dictated by a hierarchical structure.

Nonetheless, the application of this work is limited to simple low-dimensional systems. Although a viable alternative to expensive mixed-integer optimization, the inference techniques used in this chapter still present a bottleneck in the face of scalability to more complex systems and higher dimensions. While our MAP approach significantly improves the quality of expectation-maximization solutions, it nevertheless struggles in more challenging environments.

A possible course of action is to investigate Bayesian nonparametric extensions of hybrid dynamic Bayesian networks based on non-conjugate variational inference. Fully Bayesian methods tend to improve learning in large structured models significantly. Another potential avenue of research is to improve the hybrid reinforcement learning framework by considering the control-as-inference paradigm. Such approaches may offer ways of integrating the Bayesian structure of the models into the control optimization and, in the process, achieve an uncertainty-aware approach that is better equipped to deal with the exploration-exploitation dilemma.

Chapter 4

Distributionally Robust Control and Filtering

Trajectory optimization and model predictive control (MPC) are essential techniques underpinning advanced robotic applications, ranging from autonomous driving to full-body humanoid control. State-of-the-art algorithms have focused on data-driven approaches that infer the Bayesian system dynamics online and incorporate that posterior uncertainty during planning and control. However, despite their success, such approaches are still susceptible to catastrophic errors due to statistical learning biases, unmodeled disturbances, or even directed adversarial attacks.

In this chapter, we tackle the problem of dynamics mismatch and propose a distributionally robust optimal control formulation that alternates between two relative entropy trust region optimization problems. Our method finds the worst-case maximum entropy Gaussian posterior over the dynamics parameters and the corresponding robust policy. Furthermore, our approach admits a closed-form backward pass for a certain class of systems. Finally, we demonstrate the resulting robustness on linear and nonlinear numerical examples. In addition, we propose using the same structure of multi-stage optimization to deal with the exponentially growing mixture size in stochastic switching systems. We present multiple variations that incorporate the cost into Gaussian reduction techniques and lead to optimistic or pessimistic approximation of the stochastic state.

4.1 Introduction

Trajectory optimization (Mayne, 1966; Tassa et al., 2012; Watson et al., 2020a) is a well-established tool for solving control problems that rely on a model of the system dynamics to optimize a control signal that induces a desired system behavior. However, as systems of interest are getting more complex, involving nonlinear effects and high-dimensional state-action spaces, accurate analytical modeling has become challenging, if not impossible, in some cases. Consequently, data-driven control approaches that employ black- and gray-box statistical models are becoming popular rapidly.

However, trajectory optimization may exploit model imperfections that can arise as a result of statistical learning biases, resulting in brittle controllers that may fail at deployment time due to modeling discrepancies. The recent trend of over-reliance on learning from simulated data carries with it similar pitfalls with respect to optimization bias.

Advanced trajectory optimization and MPC techniques successfully use learned probabilistic models to incorporate the uncertainty of data-driven learning (Kamthe & Deisenroth, 2018; Hewing et al., 2018). However, those approaches are not robust against adversarial disturbances and general model mismatch. An alternative to such probabilistic modeling is the robust control paradigm. Unfortunately, robust methods tend to produce sub-optimal controllers on average because the disturbance model gives too much power to the adversary, forcing the controller to be too conservative.

Lately, a new class of methods at the intersection of robust and stochastic optimal control is gaining momentum, based on distributionally robust optimization (DRO) (Scarf, 1958; Delage & Ye, 2010; Van Parys et al., 2015; Coulson et al., 2019; Yang, 2020; Zhu et al., 2020; Coppens et al., 2020), as it promises to combine the strengths of both approaches. In the DRO framework, one seeks to find a controller that performs optimally under a worst-case stochastic model chosen from a so-called ambiguity set. Here, the adversary's strength is easier to calibrate compared to the classical robust control.

To make distributionally robust optimization practical, one needs to be able to infer the optimal adversary, i.e., find the worst-case stochastic system in closed form or numerically. So far, closed-form solutions have been obtained only for special choices of the ambiguity set (Rahimian & Mehrotra, 2019). For example, when the ambiguity set is given as a relative entropy ball around a nominal distribution (Hu & Hong, 2013; Charalambous & Rezaei, 2007). In this chapter, we build upon this insight to develop an algorithm for distributionally robust trajectory optimization.

We consider the problem of controlling a discrete-time stochastic dynamical system with transition density $f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \boldsymbol{\theta})$ for which system uncertainty is encoded in the parameter distribution $p(\boldsymbol{\theta})$ and the ambiguity set is given by the KL ball $\mathcal{B}_\delta(\hat{p}) = \{p \mid \mathrm{KL}(p \,||\, \hat{p}) \leq \delta\}$ centered around the nominal parameter distribution $\hat{p}(\boldsymbol{\theta})$, that we assume is available after a data-driven model learning phase. We seek a time-varying stochastic policy $\pi_t(\mathbf{u}|\mathbf{x})$ that minimizes the worst-case expected cost $\max_{p \in \mathcal{B}_\delta(\hat{p})} J(\pi, p)$. In this setting, we develop an iterative trust region algorithm that alternates between optimizing the worst-case distribution p and the corresponding distributionally robust policy π. We derive optimality conditions for p and π, and an efficient forward-backward procedure in the style of differential dynamic programming is provided for each optimization step.

The resulting method applies to nonlinear systems via iterative local linearization. Empirical validation on uncertain linear and nonlinear dynamical systems demonstrates the robustness of the optimized policies against adversarial disturbances. Furthermore, we present an outlook to applying similar DRO principles to stochastic switching systems with the motivation of managing the size of exponentially growing mixtures by deriving an optimistic and a pessimistic state-filtering approach.

Our approach brings together several strands of research. First, we rely on distributionally robust optimization to find the worst-case parameter distribution. Second, our problem formulation is based on an iterative scheme of relative entropy policy search, a trust region algorithm for policy optimization (Peters et al., 2010). Finally, we use iterative linearization and numerical integration to enable the transfer to nonlinear systems. Below, we highlight related work from these areas and point out key differences.

4.2 Related Work

Distributionally robust optimization finds numerous applications in control. However, methods differ in ambiguity set representations, uncertainty, system modeling assumptions, and optimization algorithms. For example, in (Van Parys et al., 2015), the problem of controlling a linear system under distributionally robust chance constraints was tackled using a moment-based ambiguity set. The moment-based representation was also employed in (Coppens et al., 2020) to derive high-probability guarantees for the stability of a linear system with multiplicative noise. For Wasserstein-based ambiguity sets, a general formulation was given in (Yang, 2020), which, however, requires solving a semi-infinite problem numerically while we obtain a closed-form solution instead. Furthermore, a Wasserstein-based ambiguity set on the noise distribution was used to solve a data-enabled control problem in (Coulson et al., 2019). For the linear-quadratic case, a relaxed version was solved in (Kim & Yang, 2020) via a modified Riccati equation. The model, however, only included ambiguity over the additive noise while the rest of the dynamics were assumed known and time-invariant. In this chapter, in contrast, we employ a time-varying linearized probabilistic dynamics model and allow ambiguity in the distribution over all model parameters.

Ambiguity sets based on relative entropy were also studied in several prior works. Distributionally robust optimization of stochastic nonlinear partially observable systems with relative entropy constraints was studied in a general abstract setting in (Charalambous & Rezaei, 2007), where the ambiguity set is formulated on the space of path measures. In contrast, we consider ambiguity sets on the space of parameter distributions of the underlying dynamical system. In (Petersen et al., 2000), a formulation for nonlinear systems was presented where the uncertainty in the additive disturbance was ambiguous. Through Lagrangian duality, a connection to risk-sensitive control was furthermore established. This connection was recently used to derive a model predictive control algorithm (Nishimura et al., 2021) that builds upon the iterative linear-exponential-quadratic Gaussian (iLEQG) (Farshidian & Buchli, 2015). That approach uses a cross-entropy method (CEM) to optimize the risk-sensitivity parameter and obtain the worst-case distribution, as opposed to our algorithm, which uses a principled approach based on our trust region optimization to solve the resulting DRO problem.

Finally, our DRO formulation captures the uncertainty in the whole trajectory rather than considering only the noise distribution. This feature gives the adversary additional free-

dom in allocating the disturbance budget along the trajectory at critical time steps.

Our approach builds upon relative entropy policy search (REPS) (Peters et al., 2010) by extending it with an adversarial optimization with respect to the ambiguous distribution over the dynamics parameters. A risk-sensitive formulation based on REPS involving the entropic risk measure in a model-free setting was considered in (Nass et al., 2019). In this chapter, we instead consider the model-based setting (Levine & Koltun, 2013) and optimize a controller under the worst-case distribution instead of the nominal one. Furthermore, we introduce an additional approximate state propagation step to accommodate for uncertainty in the dynamics parameters.

Stochastic optimal control with linearized dynamics (Mayne, 1966; Todorov & Li, 2005; Watson et al., 2020a) is a powerful technique for controlling nonlinear systems. A locally optimal controller can be found via dynamic programming techniques by using first-order approximations of the dynamics along a given trajectory. The updated controller is used to generate a new reference trajectory, and the process is iterated until convergence. Since the linearized dynamics is only a good approximation around the linearization point, it is important to limit the optimism in the controller updates. One successful approach has been to enforce a relative entropy bound between the trajectory distributions of successive iterations. This technique is used for trajectory optimization in the context of guided policy search (GPS) (Levine & Koltun, 2013) and is dubbed maximum-entropy iterative linear quadratic Gaussian. Our approach relies on a similar formulation to optimize a time-variant policy under uncertain dynamics.

In contrast to the aforementioned stochastic control frameworks, a distributionally robust approach not only considers the stochasticity captured by the probabilistic model but also accounts for *ambiguity*, meaning uncertainty about the probabilistic model itself.

4.3 Problem Statement

In this chapter, we concentrate on finite-horizon Markov decision processess (MDPs) with a state space $\mathbf{X} \subseteq \mathbb{R}^d$, an action space $\mathbf{U} \subseteq \mathbb{R}^m$, and a time horizon T. We assume a probabilistic state transition density $p(\mathbf{x}', \boldsymbol{\theta} | \mathbf{x}, \mathbf{u}) = f(\mathbf{x}' | \mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) p(\boldsymbol{\theta})$, where $p(\boldsymbol{\theta})$ is a distribution over the dynamics parameters. The policy $\pi_t(\mathbf{u}|\mathbf{x})$, a time-variant conditional density, induces the state distribution $\mu_t(\mathbf{x})$ according to transition dynamics.

In this setting, the stochastic optimal control objective can be written as

$$J(\pi_t, p) = \sum_{t=1}^{T-1} \iint c_t(\mathbf{x}, \mathbf{u}) \mu_t(\mathbf{x}) \pi_t(\mathbf{u}|\mathbf{x}) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{u} + \int c_T(\mathbf{x}) \mu_T(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \tag{4.1}$$

where $c(\mathbf{x}, \mathbf{u})$ is the cost function. With slight abuse of notation, we refer to $\pi_{1:T-1}$ with π_t. When necessary, we extend this notation to μ_t and p_t. This objective is constrained

by the following integral equation describing the evolution of $\mu_t(\mathbf{x})$ over time

$$\mu_{t+1}(\mathbf{x}') = \iiint \mu_t(\mathbf{x})\pi_t(\mathbf{u}|\mathbf{x})f(\mathbf{x}'|\mathbf{x},\mathbf{u},\boldsymbol{\theta})p(\boldsymbol{\theta})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta}. \tag{4.2}$$

The distributionally robust trajectory optimization can then be written as a minimax problem over the distributions $\pi_t(\mathbf{u}|\mathbf{x})$ and $p(\boldsymbol{\theta})$

$$\underset{\pi_t}{\text{minimize}} \quad J(\pi_t, p^*), \tag{4.3a}$$

$$\text{subject to} \quad \int \pi_t(\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u} = 1, \quad \forall \mathbf{x}, \forall t < T, \tag{4.3b}$$

where $p^*(\boldsymbol{\theta})$ is the worst-case distribution given by

$$p^* := \underset{p}{\arg\max} \quad J(\pi_t, p), \tag{4.4a}$$

$$\text{subject to} \quad \mathrm{KL}(p(\boldsymbol{\theta})\,||\,\hat{p}(\boldsymbol{\theta})) \leq \delta, \tag{4.4b}$$

$$\int p(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} = 1, \tag{4.4c}$$

where $\hat{p}(\boldsymbol{\theta})$ is the nominal parameter distribution, and δ controls the size of the corresponding KL-based distributional ambiguity set.

The robust optimization problem is typically hard to solve for general nonlinear dynamical systems $p(\mathbf{x}'|\mathbf{x},\mathbf{u},\boldsymbol{\theta})$ and arbitrary forms of $p(\boldsymbol{\theta})$. Our approach is to solve the nested optimization via a regularized iterative sequential programming technique by using an additional trust region imposed on the outer policy optimization problem in Equation (4.3).

4.4 Trust Region Distributionally Robust Control

Introducing a trust region over π_t has multiple advantages. On the one hand, it regularizes the policy optimization step, which is crucial for the convergence of the overall minimax problem. On the other hand, it leads to a tractable maximum-entropy stochastic optimal control framework for dealing with nonlinear dynamics through successive linearization around a local trajectory distribution (Levine & Koltun, 2013; Arenz et al., 2016).

The resulting overall approach alternates between updating the parameter and policy distribution. For every iteration k, we compute the updated worst-case distribution p^{k+1} given the ambiguity set $\mathcal{B}_\delta$ around the nominal $\hat{p}$, and policy π_t^k

$$p^{k+1} = \underset{p\in\mathcal{B}_\delta(\hat{p})}{\arg\max} \quad J(\pi_t^k, p), \tag{4.5}$$

Algorithm 4.1: Distributionally Robust Trajectory Optimization

input: $\hat{\mu}_1, c_t, f, \hat{p}, \delta, \varepsilon, K$

initialize: π_t^1

1 **for** $k \leftarrow 1$ **to** K **do**

2 $\quad p^{k+1} \leftarrow$ **WorstCaseParameters**$(\hat{p}, \pi_t^k, \hat{\mu}_1, c_t, f, \delta)$

3 $\quad \pi_t^{k+1} \leftarrow$ **RobustPolicyUpdate**$(\pi_t^k, p^{k+1}, \hat{\mu}_1, c_t, f, \varepsilon)$

4 $p^* \leftarrow p^{K+1}, \quad \pi_t^* \leftarrow \pi_t^{K+1}$

output: π_t^*, p^*

then we compute the updated robust policy π_t^{k+1} under p^{k+1} in a trust region $\mathcal{B}_\varepsilon$ around the old policy π_t^k

$$\pi_t^{k+1} = \underset{\pi_t \in \mathcal{B}_\varepsilon(\pi_t^k)}{\arg\min} \quad J(\pi_k, p^{k+1}). \tag{4.6}$$

These steps can also be seen as trust region versions of the proximal updates performed by the mirror descent algorithm (Beck & Teboulle, 2003). Algorithm 4.1 offers a schematic view of the optimization. The following sections provide further details.

4.4.1 Worst-Case Parameter Distribution

The parameter distribution optimization (4.5) for a single iteration k is given by

$$\underset{p_t}{\text{maximize}} \quad \sum_{t=1}^{T-1} \iint c_t(\mathbf{x}, \mathbf{u})\, \mu_t(\mathbf{x})\, \pi_t^k(\mathbf{u}|\mathbf{x})\, \mathrm{d}\mathbf{x}\, \mathrm{d}\mathbf{u} + \int c_T(\mathbf{x})\, \mu_T(\mathbf{x})\, \mathrm{d}\mathbf{x}, \tag{4.7a}$$

$$\text{subject to} \quad \iiint \mu_t(\mathbf{x})\, \pi_t^k(\mathbf{u}|\mathbf{x})\, f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \boldsymbol{\theta})\, p_t^{k+1}(\boldsymbol{\theta})\, \mathrm{d}\mathbf{u}\, \mathrm{d}\mathbf{x}\, \mathrm{d}\boldsymbol{\theta} = \mu_{t+1}(\mathbf{x}'), \tag{4.7b}$$

$$\sum_{t=1}^{T-1} \mathrm{KL}(p_t^{k+1}(\boldsymbol{\theta}) \,||\, \hat{p}(\boldsymbol{\theta})) \leq \delta, \tag{4.7c}$$

$$\int p_t^{k+1}(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} = 1, \quad \mu_1(\mathbf{x}) = \hat{\mu}_1(\mathbf{x}). \tag{4.7d}$$

Notice that we have moved to a time-variant worst-case parameter distribution $p_t(\boldsymbol{\theta})$. Although this formulation is more general, it is crucial to make this assumption in order to disentangle the adversary's influence over time and restrict it to future time steps. This modification makes sense when considering that the robust policy $\pi_t(\mathbf{u}|\mathbf{x})$ is likewise time-variant and only influences the current and future time steps.

By solving the former primal problem using the method of Lagrangian multipliers (Boyd & Vandenberghe, 2004), we arrive at the optimal worst-case parameter distribution p_t^{k+1}

$$p_t^{k+1}(\boldsymbol{\theta}) = \frac{\hat{p}(\boldsymbol{\theta}) \exp\left[-\frac{1}{\beta} Q_t(\boldsymbol{\theta})\right]}{\int \hat{p}(\boldsymbol{\theta}) \exp\left[-\frac{1}{\beta} Q_t(\boldsymbol{\theta})\right] \mathrm{d}\boldsymbol{\theta}} \tag{4.8}$$

a softmax distribution with a temperature $\beta \leq 0$ that corresponds to the trust region constraint in Equation (4.7c) and a parameter value function $W_t(\boldsymbol{\theta})$

$$W_t(\boldsymbol{\theta}) = \iiint V_{t+1}^{\boldsymbol{\theta}}(\mathbf{x}') \, \mu_t(\mathbf{x}) \, \pi_t^k(\mathbf{u}|\mathbf{x}) \, f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{x}', \tag{4.9}$$

where $V_{t+1}^{\boldsymbol{\theta}}(\mathbf{x}')$ is the Lagrangian function associated with Equation (4.7b) and acts as an adversarial state-value function under the last policy $\pi_t^k(\mathbf{u}|\mathbf{x})$.

By plugging the solution in Equation (4.8) back into the primal, we retrieve the dual F as a function of μ, $V^{\boldsymbol{\theta}}$ and β

$$\begin{aligned} F = &\sum_{t=1}^{T-1} \iint c_t(\mathbf{x}, \mathbf{u}) \mu_t(\mathbf{x}) \pi_t^k(\mathbf{u}|\mathbf{x}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x} + \int c_T(\mathbf{x}) \mu_T(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ &+ \int V_1^{\boldsymbol{\theta}}(\mathbf{x}) \hat{\mu}_1(\mathbf{x}) \, \mathrm{d}\mathbf{x} - \sum_{t=1}^{T-1} \int V_t^{\boldsymbol{\theta}}(\mathbf{x}') \mu_t(\mathbf{x}') \, \mathrm{d}\mathbf{x}' - \int V_T^{\boldsymbol{\theta}}(\mathbf{x}') \mu_T(\mathbf{x}') \, \mathrm{d}\mathbf{x}' \\ &- \beta\delta - \beta \sum_{t=1}^{T-1} \log \int \hat{p}(\boldsymbol{\theta}) \exp\left[-\frac{1}{\beta} W_{t+1}(\boldsymbol{\theta})\right] \mathrm{d}\boldsymbol{\theta}. \end{aligned} \tag{4.10}$$

We set the partial derivative of the dual with respect to $\mu_t(\mathbf{x})$ to zero and get a backward recursion for computing $V_t^{\boldsymbol{\theta}}(\mathbf{x})$

$$\begin{aligned} V_t^{\boldsymbol{\theta}}(\mathbf{x}) = &\int c_t(\mathbf{x}, \mathbf{u}) \, \pi_t^k(\mathbf{u}|\mathbf{x}) \, \mathrm{d}\mathbf{u} \\ &+ \iiint V_{t+1}^{\boldsymbol{\theta}}(\mathbf{x}') \, \pi_t^k(\mathbf{u}|\mathbf{x}) \, f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \, p^{k+1}(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x}', \end{aligned} \tag{4.11}$$

where $V_T(\mathbf{x}) = c_T(\mathbf{x})$. Similarly, setting the partial derivative of the dual with respect to $V^{\boldsymbol{\theta}}$ to zero delivers a forward recursion for $\mu_t(\mathbf{x})$

$$\mu_{t+1}(\mathbf{x}') = \iiint \mu_t(\mathbf{x}) \, \pi_t^k(\mathbf{u}|\mathbf{x}) \, f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \, p_t^{k+1}(\boldsymbol{\theta}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x} \, \mathrm{d}\boldsymbol{\theta}, \tag{4.12}$$

Algorithm 4.2: Worst-Case Parameter Optimization

input: $\hat{p}, \pi_t^k, \hat{\mu}_1, c_t, f, \delta$

initialize: β

$q_t \leftarrow$ **ParameterForwardPass**$(\hat{p}, \pi_t^k, f, \hat{\mu}_1)$

while F is not at minimum **do**

  repeat

    $p_t^{k+1}, V_t^{\theta} \leftarrow$ **ParameterBackwardPass**$(q_t, \pi_t^k, c_t, f, \beta)$

    $\mu_t \leftarrow$ **ParameterForwardPass**$(p_t^{k+1}, \pi_t^k, f, \hat{\mu}_1)$

    $q_t \leftarrow \lambda\mu_t + (1-\lambda)q_t$

  until $\mathrm{KL}(q_t \,||\, \mu_t) \approxeq 0$

  $\dfrac{\partial F}{\partial \beta} \leftarrow$ **ComputeBetaGradient**$(p_t^{k+1}, \hat{p}, \delta)$

  $\beta \leftarrow \beta - \eta \dfrac{\partial F}{\partial \beta}$

output: p_t^{k+1}, μ_t

Algorithm 4.3: Distributionally Robust Policy Optimization

input: $\pi_t^k, p_t^{k+1}, \hat{\mu}_1, c_t, f, \varepsilon$

initialize: α

while G is not at maximum **do**

  $\pi_t^{k+1}, V_t^{\pi} \leftarrow$ **PolicyBackwardPass**$(p^{k+1}, c_t, f, \alpha)$

  $\mu_t \leftarrow$ **PolicyForwardPass**$(p_t^{k+1}, \pi_t^{k+1}, f, \hat{\mu}_1)$

  $\dfrac{\partial G}{\partial \alpha} \leftarrow$ **ComputeAlphaGradient**$(\pi_t^{k+1}, \pi_t^k, \varepsilon, \mu_t)$

  $\alpha \leftarrow \alpha + \rho \dfrac{\partial G}{\partial \alpha}$

output: π_t^{k+1}, μ_t

where the initial state distribution $\mu_1(\mathbf{x}) = \hat{\mu}_1(\mathbf{x})$ which is assumed given.

Finally, the optimal temperature β that satisfies the trust region in Equation (4.7c) is optimized numerically via gradient descent on the dual where

$$\beta^{i+1} = \beta^i - \eta_i \sum_{t=1}^{T-1} \mathrm{KL}(p_t^{k+1}(\boldsymbol{\theta}) \,||\, \hat{p}(\boldsymbol{\theta})) + \eta_i \delta,$$

and η_i is some step size. This process iterates over μ, $V^{\boldsymbol{\theta}}$ and β until convergence, see Algorithm 4.2. Given the circular dependency between $V^{\boldsymbol{\theta}}$, μ and p, we update μ through a barycentric interpolation scheme, analogous to (Abdulsamad et al., 2017). A more detailed derivation of this optimization problem is available in Appendix D.

4.4.2 Worst-Case Robust Policy

Imposing a trust region constraint on the robust stochastic optimal control formulation in (4.3) results in the following optimization problem

$$\begin{aligned}
\underset{\pi_t}{\text{minimize}} \quad & \sum_{t=1}^{T-1} \iint c_t(\mathbf{x}, \mathbf{u})\, \mu_t(\mathbf{x}) \pi_t^{k+1}(\mathbf{u}|\mathbf{x})\, \mathrm{d}\mathbf{x}\, \mathrm{d}\mathbf{u} + \int c_T(\mathbf{x}) \mu_T(\mathbf{x})\, \mathrm{d}\mathbf{x}, && \text{(4.13a)} \\
\text{subject to} \quad & \iiint \mu_t(\mathbf{x})\, \pi_t^{k+1}(\mathbf{u}|\mathbf{x})\, f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \boldsymbol{\theta})\, p_t^{k+1}(\boldsymbol{\theta})\, \mathrm{d}\mathbf{u}\, \mathrm{d}\mathbf{x}\, \mathrm{d}\boldsymbol{\theta} = \mu_{t+1}(\mathbf{x}'), && \text{(4.13b)} \\
& \sum_{t=1}^{T-1} \int \mu_t(\mathbf{x})\, \mathrm{KL}(\pi_t^{k+1}(\mathbf{u}|\mathbf{x}) \,||\, \pi_t^k(\mathbf{u}|\mathbf{x}))\, \mathrm{d}\mathbf{x} \leq \varepsilon, && \text{(4.13c)} \\
& \int \pi_t^{k+1}(\mathbf{u}|\mathbf{x})\, \mathrm{d}\mathbf{u} = 1, \quad \mu_1(\mathbf{x}) = \hat{\mu}_1(\mathbf{x}). && \text{(4.13d)}
\end{aligned}$$

By formulating the Lagrangian and solving for the robust policy π_t^{k+1}, we find

$$\pi_t^{k+1}(\mathbf{u}|\mathbf{x}) = \frac{\pi_t^k(\mathbf{u}|\mathbf{x}) \exp\left[-\frac{1}{\alpha} Q_t^{\pi}(\mathbf{x}, \mathbf{u})\right]}{\int \pi_t^k(\mathbf{u}|\mathbf{x}) \exp\left[-\frac{1}{\alpha} Q_t^{\pi}(\mathbf{x}, \mathbf{u})\right] \mathrm{d}\mathbf{u}}, \tag{4.14}$$

where $Q_t(\mathbf{x}, \mathbf{u})$ is the state-action value function

$$Q_t(\mathbf{x}, \mathbf{u}) = c_t(\mathbf{x}, \mathbf{u}) + \iint V_{t+1}^{\pi}(\mathbf{x}')\, f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \boldsymbol{\theta})\, p_t^{k+1}(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}\, \mathrm{d}\mathbf{x}'.$$

The temperature parameter $\alpha \geq 0$ and function $V_t^{\pi}(\mathbf{x})$ are the Lagrangian variables associated with Equation (4.13c) and Equation (4.13b).

Substituting Equation (4.14) back into the primal delivers the policy dual function G

$$\begin{aligned} G = &\int c_T(\mathbf{x})\mu_T(\mathbf{x})\,\mathrm{d}\mathbf{x} + \int V_1^\pi(\mathbf{x})\hat{\mu}_1(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &- \sum_{t=1}^{T-1}\int V_t^\pi(\mathbf{x}')\mu_t(\mathbf{x}')\,\mathrm{d}\mathbf{x}' - \int V_T^\pi(\mathbf{x}')\mu_T(\mathbf{x}')\,\mathrm{d}\mathbf{x}' - \alpha\varepsilon \\ &- \alpha\sum_{t=1}^{T-1}\pi_t^k(\mathbf{u}|\mathbf{x})\exp\left[-\frac{1}{\alpha}Q_{t+1}^\pi(\mathbf{x},\mathbf{u})\right]\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}. \end{aligned} \tag{4.15}$$

By setting the derivatives of G with respect to $\mu_t(\mathbf{x})$ to zero, we arrive at an optimality condition in the form of a backward recursion for calculating $V_t^\pi(\mathbf{x})$

$$V_t^\pi(\mathbf{x}) = \alpha\log\int \pi_t^k(\mathbf{u}|\mathbf{x})\exp\left[-\frac{1}{\alpha}Q_t(\mathbf{x},\mathbf{u})\right]\mathrm{d}\mathbf{u}, \tag{4.16}$$

where $V_T^\pi = c_T(\mathbf{x})$. On the other hand, the derivatives of G with respect to $V^\pi(\mathbf{x})$ lead to a forward recursion for $\mu_t(\mathbf{x})$ that fulfills the propagation constraint

$$\mu_{t+1}(\mathbf{x}') = \iiint \mu_t(\mathbf{x})\,\pi_t^{k+1}(\mathbf{u}|\mathbf{x})\,f(\mathbf{x}'|\mathbf{x},\mathbf{u},\boldsymbol{\theta})\,p_t^{k+1}(\boldsymbol{\theta})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta}. \tag{4.17}$$

Similar to the optimization in Section 4.4.1, the temperature α is optimized via gradient ascent on the policy dual with

$$\alpha^{i+1} = \alpha^i + \rho_i\sum_{t=1}^{T-1}\int \mu_t(\mathbf{x})\,\mathrm{KL}(\pi_t^{k+1}(\mathbf{u}|\mathbf{x})\,||\,\pi_t^k(\mathbf{u}|\mathbf{x}))\,\mathrm{d}\mathbf{x} - \rho_i\varepsilon,$$

where ρ_i is an adaptive step size. We refer to (Nocedal & Wright, 2006) for the convergence properties of trust region optimization and specific rules for choosing and adapting the size ε. Algorithm 4.3 gives an outline of the overall optimization procedure. A more detailed derivation is available in Appendix D.

4.5 Practical Realization Conditions

The recursive optimality conditions in Section 4.4.1 and Section 4.4.2 offer a general solution to the optimization problems without any guarantees for computational tractability. In this section, we discuss the assumptions necessary so that the proposed forward and backward passes are feasible.

4.5.1 Linearized Quadratic Systems

Firstly, we assume linear dynamics with a Gaussian additive noise

$$f(\mathbf{x}'|\mathbf{x},\mathbf{u},\boldsymbol{\theta}) = \mathrm{N}(\mathbf{x}'|\boldsymbol{\Theta}\boldsymbol{\tau}, \boldsymbol{\Sigma}_{\mathbf{x}'}),$$

where $\boldsymbol{\Theta} = \begin{bmatrix}\mathbf{A}, & \mathbf{B}, & \mathbf{c}\end{bmatrix}$ is the aggregate linear parameter matrix and $\boldsymbol{\tau} = \begin{bmatrix}\mathbf{x}, & \mathbf{u}, & \mathbf{1}\end{bmatrix}^\top$ is the combined state-action vector. Moreover, the cost function $c(\mathbf{x},\mathbf{u})$ is presumed quadratic in state and action. Finally, the nominal parameter distribution is a Gaussian density $\hat{p}(\boldsymbol{\theta}) = \mathrm{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)$. Under these assumptions, the following holds

1. The state- and parameter-value functions V^π and V^θ start at time T as quadratic functions and remain as such during the backward recursion due to the functional compatibility with the Gaussian probabilistic dynamics.
2. The resulting policy is a time-variant linear Gaussian $\pi_t(\mathbf{u}|\mathbf{x}) = \mathrm{N}(\mathbf{u}|\mathbf{K}_t\mathbf{x}+\mathbf{k}_t, \boldsymbol{\Sigma}_{\mathbf{u},t})$, where $(\mathbf{K}_t, \mathbf{k}_t)$ are the linear feedback matrix and affine offset.
3. The optimal time-variant worst-case distribution p_t is a Gaussian density.
4. Propagation of the state through probabilistic dynamics results in a non-Gaussian distribution due to the expectation over $p_t(\boldsymbol{\theta})$. We circumvent this issue by approximating the forward recursion via spherical cubature.

This setting can be extended to support nonlinear dynamical systems and non-convex costs via local approximations, which mirrors the iterative schemes used in differential dynamic programming (DDP) (Mayne, 1966) and iterative linear-quadratic regulator (iLQR) (Tassa et al., 2012). However, in our formulation, a more principled regularization is achieved through the trust region constraint on the policy (Levine & Koltun, 2013). Note that this extension requires a new reference nominal distribution $\hat{p}^k(\boldsymbol{\theta})$ for every linearization iteration k, which we assume is given by an external statistical learning process.

4.5.2 Cubature-Based State Propagation

We briefly discuss the details of the approximate cubature forward recursion. The state propagation adheres to the probabilistic dynamics constraint

$$\mu(\mathbf{x}') = \iiint \mu(\mathbf{x})\pi(\mathbf{u}|\mathbf{x})f(\mathbf{x}'|\mathbf{x},\mathbf{u},\boldsymbol{\theta})p(\boldsymbol{\theta})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta}, \tag{4.18}$$

in which we omit the superscripts and subscripts for brevity. Under the assumptions introduced in the previous section, the expected dynamics can be written as

$$p(\mathbf{x}'|\mathbf{x},\mathbf{u}) = \int f(\mathbf{x}'|\mathbf{x},\mathbf{u},\boldsymbol{\theta})p(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} = \mathrm{N}(\mathbf{x}'|\mathbf{M}_\theta\boldsymbol{\tau}, \boldsymbol{\Sigma}_{\mathbf{x}'} + (\boldsymbol{\tau}^\top \otimes \mathbf{I}_\theta)^\top \boldsymbol{\Sigma}_\theta(\boldsymbol{\tau}\otimes\mathbf{I}_\theta)),$$

where $\mathbf{M}_\theta$ is defined according to $\boldsymbol{\mu}_\theta = \text{vec}(\mathbf{M}_\theta)$ with vec denoting the vectorization operator, the operator $\otimes$ stands for the Kronecker product, and $\mathbf{I}_\theta$ is the identity matrix with size equal to the dimension of $\boldsymbol{\theta}$. We write the covariance as

$$\boldsymbol{\Sigma}(\boldsymbol{\tau}) = \boldsymbol{\Sigma}_{\mathbf{x}'} + (\boldsymbol{\tau}^\top \otimes \mathbf{I}_\theta)^\top \boldsymbol{\Sigma}_\theta (\boldsymbol{\tau} \otimes \mathbf{I}_\theta)$$

where the second term depends on both state and action through $\boldsymbol{\tau}$. This leads to the integral in Equation (4.18) being non-Gaussian. We use the cubature transform as described in (Solin, 2010), which constitutes a variant of the unscented transform (Wan et al., 2001), to approximate the propagated state distribution. Therefore, we rewrite the dynamics equivalently as

$$\mathbf{x}' = \mathbf{M}_\theta \boldsymbol{\tau} + \sqrt{\boldsymbol{\Sigma}(\boldsymbol{\tau})}\,\boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathrm{N}(\mathbf{0}, \mathbf{I}),$$

where the matrix square root is the triangular Cholesky factor. If we include the noise $\boldsymbol{\xi}$ in the augmented state $\hat{\boldsymbol{\tau}} := \begin{bmatrix}\mathbf{x}, & \mathbf{u}, & \boldsymbol{\xi}\end{bmatrix}^\top$, the cubature computation resembles propagating an augmented distribution $p(\hat{\boldsymbol{\tau}})$ through nonlinear deterministic dynamics

$$\begin{aligned} p(\hat{\boldsymbol{\tau}}) &= \mu(\mathbf{x}|\mathbf{m}, \boldsymbol{\Sigma}_\mathbf{x})\, \pi(\mathbf{u}|\mathbf{K}\mathbf{x} + \mathbf{k}, \boldsymbol{\Sigma}_\mathbf{u})\, p(\boldsymbol{\xi}|\mathbf{0}, \mathbf{I}_\mathbf{x}) \\ &= \mathrm{N}\left(\begin{bmatrix}\mathbf{x} \\ \mathbf{u} \\ \boldsymbol{\xi}\end{bmatrix} \middle| \begin{bmatrix}\mathbf{m} \\ \mathbf{K}\mathbf{m} + \mathbf{k} \\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_\mathbf{x} & \boldsymbol{\Sigma}_\mathbf{x}\mathbf{K}^\top & \mathbf{0} \\ \mathbf{K}\boldsymbol{\Sigma}_\mathbf{x} & \boldsymbol{\Sigma}_\mathbf{u} + \mathbf{K}\boldsymbol{\Sigma}_\mathbf{x}\mathbf{K}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_\mathbf{x}\end{bmatrix} \right). \end{aligned}$$

This reformulation allows us to apply standard cubature rules to obtain the next approximation of the state distribution $\mu(\mathbf{x}')$.

4.5.3 Existence of The Worst-Case Distribution

Given the assumptions in Section 4.5, it is possible to compute the worst-case parameter distribution $p_t^*(\boldsymbol{\theta}) = \mathrm{N}(\boldsymbol{\theta}|\boldsymbol{\Omega}_t^{-1}\boldsymbol{\omega}_t, \boldsymbol{\Omega}_t^{-1})$ in closed-form

$$\begin{aligned} \boldsymbol{\omega}_t &= \hat{\boldsymbol{\Lambda}}_\theta \hat{\boldsymbol{\mu}}_\theta - \frac{1}{\beta}(\mathbf{s}_{\mathbf{xu},t}^\top \otimes \mathbf{I}_\mathbf{x})^\top \nu_{t+1}^\theta, \\ \boldsymbol{\Omega}_t &= \hat{\boldsymbol{\Lambda}}_\theta + (\boldsymbol{\Sigma}_{\mathbf{xu},t} \otimes \mathbf{V}_{t+1}^\theta) + \frac{2}{\beta}(\mathbf{s}_{\mathbf{xu},t}^\top \otimes \mathbf{I}_\mathbf{x})^\top \mathbf{V}_{t+1}^\theta (\mathbf{s}_{\mathbf{xu},t}^\top \otimes \mathbf{I}_\mathbf{x}), \end{aligned}$$

where $\hat{\boldsymbol{\mu}}_\theta$ and $\hat{\boldsymbol{\Lambda}}_\theta$ are the mean and precision of the nominal distribution $\hat{p}(\boldsymbol{\theta})$, $\mathbf{V}^\theta$ and ν^θ are the quadratic and linear terms of the adversarial state-value function V^θ and $\mathbf{s}_\mathbf{xu}$ is the state-action distribution mean. Considering that $\beta \leq 0$ and $V^\theta \geq 0$, depending on $\hat{\boldsymbol{\Lambda}}_\theta$, there exists a value of β, for which $\boldsymbol{\Omega}$ becomes a negative-definite matrix and the distribution $p_t^*(\boldsymbol{\theta})$ does not exist anymore in a Gaussian form. To overcome such issues, we propose a variant of our algorithm that mimics the trust region sequential quadratic programming method (Nocedal & Wright, 2006). Instead of the p-update in (4.5), we

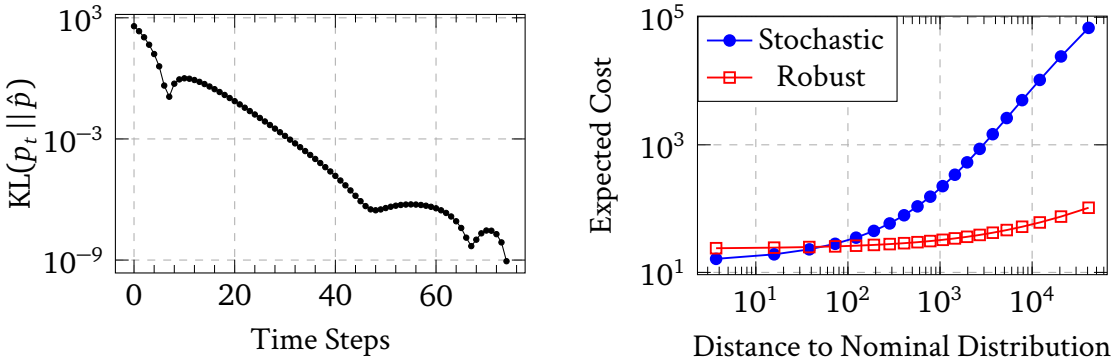


Figure 4.1: Uncertain linear system experiment. Right, the worst-case KL budget allocation over the whole trajectory. Notice that most of the deviation happens in the first part of the trajectory. Left, the expected cost of the uncertainty-aware (blue) and robust (red) controllers evaluated on a range of distributions inter- and extrapolated between and beyond the nominal and worst-case distribution. The robust controller shows much lower sensitivity to changes in the disturbance. Note the double logarithmic scale.

iteratively update the worst-case distribution over smaller trust regions

$$p^{k+1} = \max_{p \in \mathcal{B}_\delta(\hat{p}) \cap \mathcal{B}_{\delta_k}(p^k)} J(\pi^k, p),$$

where $\mathcal{B}_{\delta_k}$ is the KL-divergence trust region $\mathcal{B}_{\delta_k}(p^k) = \{p \mid \mathrm{KL}(p \| p^k) \leq \delta_k\}$. In practice, this iterative update is performed until the constraint in (4.5) becomes active.

4.6 Empirical Evaluation

We empirically evaluate the proposed distributionally robust control on a set of linear and nonlinear dynamical systems with uncertain dynamics. Without loss of generality, we limit the scope and assume the existence of a probabilistic dynamics model that has been won from data at an earlier stage. We linearize this model along a trajectory to deliver the probabilistic time-variant dynamics, i.e., the nominal distribution. Moreover, we limit the evaluation to a classic finite-horizon trajectory optimization scenario and do not consider a receding horizon control scheme.

The evaluation highlights the performance of the distributionally robust controller, iteratively optimized under its worst-case distribution, compared to an uncertainty-aware optimal controller, optimized under the nominal distributional dynamics using only the policy optimization stage of our approach. We perform this comparison by using the worst-case parameter optimization to compute an optimal disturbance on the uncertainty-aware controller and subsequently evaluate the performance of both controllers under this distur-

bance. This comparison allows the assessment of both controllers under previously unseen distributional disturbances since the worst-case attack on the uncertainty-aware controller may vary from the worst-case attack on the iteratively optimized robust controller. As evaluation criteria, we consider (1) the overall expected cost on a set of intermediate distributions between the nominal and worst-case, which we find using barycentric interpolation, (2) the induced trajectory distributions, and (3) the allocation strategy of the disturbance budget over the complete trajectory distribution. The source code of an efficient implementation can be found under https://github.com/hanyas/trajopt.

4.6.1 Uncertain Linear Dynamical System.

We consider a simple actuated mass-spring-damper linear system with a mass $m = 1\,\mathrm{kg}$, a spring constant $k = 0.01\,\mathrm{N/m}$, and a damping factor $d = 0.1\,\mathrm{N\,s/m}$. The linear differential equation has the form

$$\begin{bmatrix} \dot{x}_t \\ \ddot{x}_t \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.01 & -0.1 \end{bmatrix} \begin{bmatrix} x_t \\ \dot{x}_t \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_t,$$

which is integrated in time for a horizon $T = 75$ with a step size $\Delta t = 0.01\,\mathrm{s}$. Moreover, we assume an initial distribution $\mu_1(\mathbf{x})$ centered at $\mathbf{x}_0 = \mathbf{0}$ with a diagonal standard deviation of $\sigma_{\mathbf{x}_0} = 1 \times 10^{-1}$ and a discrete-time zero-mean process noise with a diagonal standard deviation $\sigma_{\mathbf{x}} = 1 \times 10^{-2}$. The aim is to drive the system towards a goal state $\mathbf{x}_g = [1, 0]^\top$, under a quadratic state-action cost with the matrices $\mathbf{C}_{\mathbf{x}} = \mathrm{diag}([100, 0])$ and $\mathbf{C}_{\mathbf{u}} = \mathrm{diag}([0.001])$. As stated previously, we assume the existence of a nominal distribution $\hat{p}(\boldsymbol{\theta})$, which in this case is centered at the true linear dynamics with a diagonal standard deviation of $\sigma_{\boldsymbol{\theta}} = 1 \times 10^{-4}$ to represent uncertainty over the parameters. We initialize a zero-mean controller with a diagonal standard deviation $\sigma_\pi = 10$ and set the trust region sizes $\varepsilon = 0.25$ and $\delta = 750$.

The comparison between the controllers on the linear system is depicted in Figure 4.2. The plots on the left show the trajectory distribution induced by the uncertainty-aware (blue) and robust (red) policies under the nominal parameter distribution $\hat{p}(\boldsymbol{\theta})$. The figures show an aggressive uncertainty-aware controller that takes advantage of the nominal dynamics to reach the goal as fast as possible, while the robust controller shows sub-optimal behavior. However, when evaluated on the worst-case dynamics, on the right, the uncertainty-aware controller (green) overshoots beyond the target incurring a massive cost, while the robust policy (magenta) maintains a consistent behavior associated with much lower overall cost.

Furthermore, the right plot in Figure 4.1 illustrates the worst-case KL allocation over the trajectory. A large portion of the overall deviation takes place in the first 20 time steps, leading to the sub-optimal performance of the uncertainty-aware controller in the same time window. Finally, the left plot highlights the superior performance of the robust policy (red) on a continuum of distributions interpolated between the nominal and worst-case distribu-

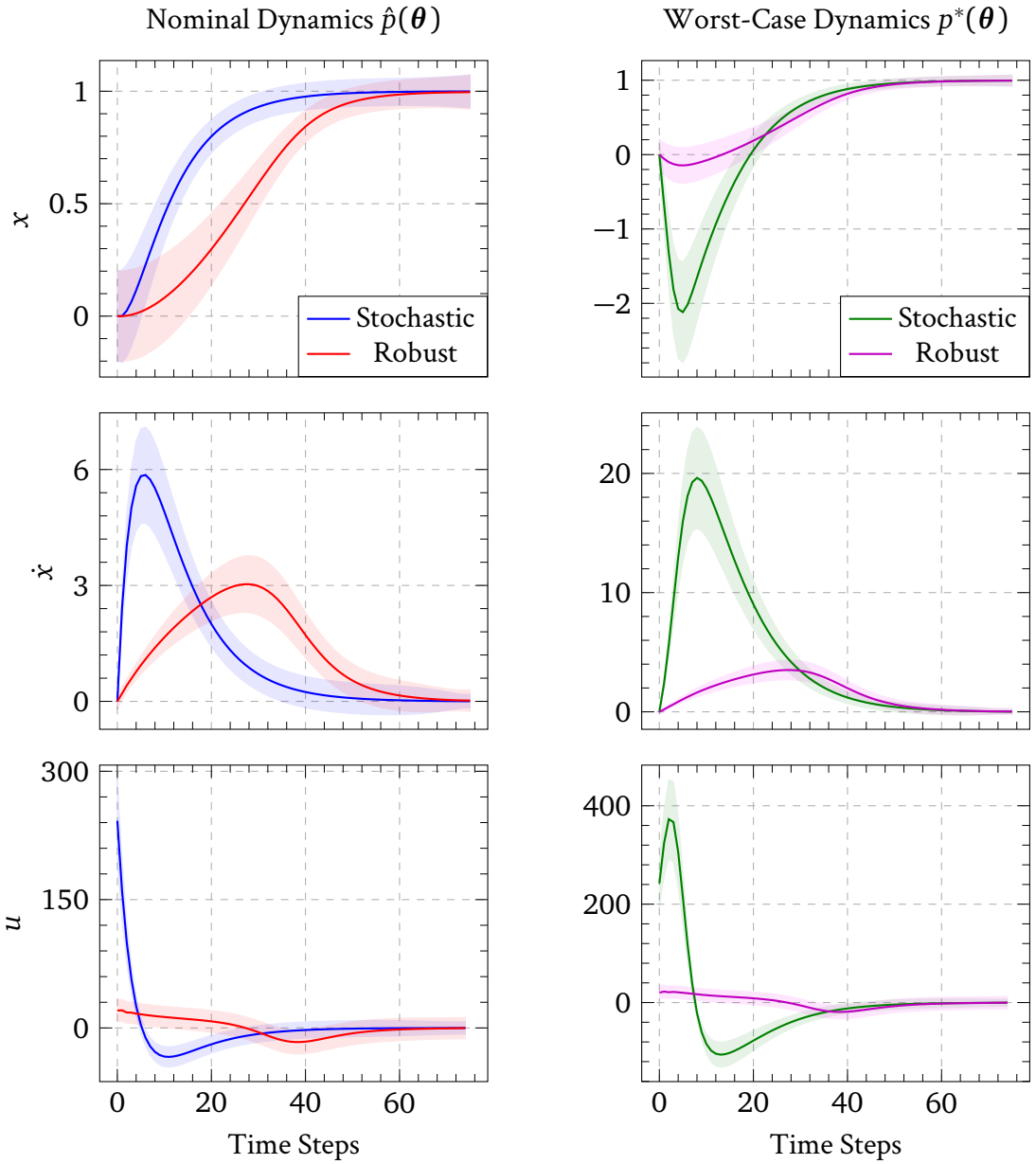


Figure 4.2: Uncertain linear system experiment. Comparison between the uncertainty-aware and distributionally robust controllers. Left, the trajectory distributions induced by standard (blue) and robust (red) controllers evaluated under the nominal dynamics distribution. The uncertainty-aware controller is aggressive and reaches the target faster. Right, the trajectory distributions induced by standard (green) and robust (magenta) controllers evaluated under the worst-case disturbance. The uncertainty-aware controller overshoots dramatically beyond the target, while the robust controller is barely affected.

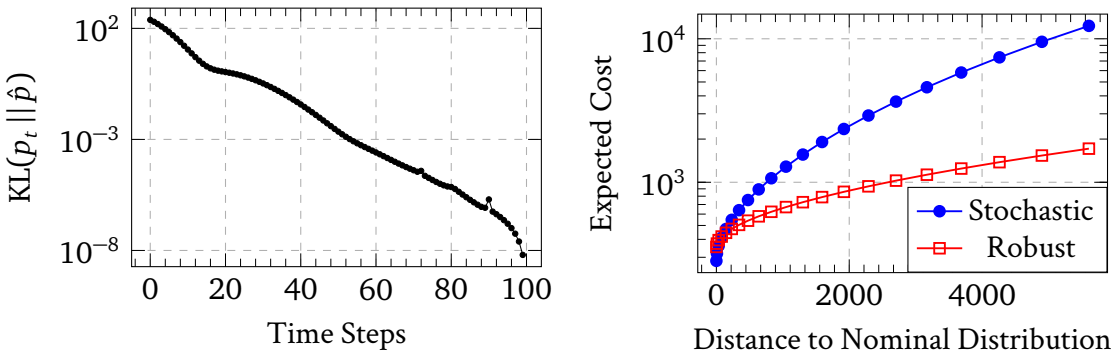


Figure 4.3: Uncertain nonlinear robot experiment. Right, allocation of the worst-case KL budget over time steps. Most of the deviation is concentrated toward the early phase of the trajectory. Left, the expected cost of the uncertainty-aware (blue) and robust (red) controllers evaluated on a range of distributions inter- and extrapolated from the nominal and worst-case distribution: The robust controller shows much lower sensitivity to changes in the disturbance.

tions and beyond using barycentric interpolation. The uncertainty-aware controller (blue) delivers better performance in a small region around the nominal distribution but very quickly worsens as the distance to that distribution increases.

4.6.2 Uncertain Nonlinear Robot Car

This experiment validates our approach for general nonlinear dynamical systems via iterative linearization of the dynamics around a trust region. We consider a nonholonomic robot moving in 2D-space. The state vector consists of the x, y-coordinates of the position, the speed v, and the orientation ψ, while the acceleration a and the steering angle ϕ are used for actuation. The global dynamics is nonlinear in state and action and given by

$$\begin{bmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{\psi}_t \\ \dot{v}_t \end{bmatrix} = \begin{bmatrix} v_t \sin \psi_t \\ v_t \cos \psi_t \\ v_t \tan(\phi_t)/d \\ a_t \end{bmatrix},$$

where the constant $d = 0.1\,\mathrm{m}$ is the car length. This ODE is integrated for a horizon $T = 100$ with a step size $\Delta t = 0.025\,\mathrm{s}$. The initial state distribution is centered at $[5, 5, 0, 0]^\top$ with a diagonal standard deviation $\sigma_{\mathbf{x}_0} = 1 \times 10^{-2}$ and the discrete-time process noise is zero-mean with a diagonal standard deviation $\sigma_{\mathbf{x}} = 1 \times 10^{-4}$. The goal state is $g = [0, 0, 0, 0]^\top$ and the quadratic cost matrices are $\mathbf{C}_{\mathbf{x}} = \mathrm{diag}([10, 10, 1, 1])$ and $\mathbf{C}_{\mathbf{u}} = \mathrm{diag}([0.1, 0.1])$. Analogous to the previous experiment, we assume the nominal parameter distribution to be centered at the linearized dynamics with a diagonal standard

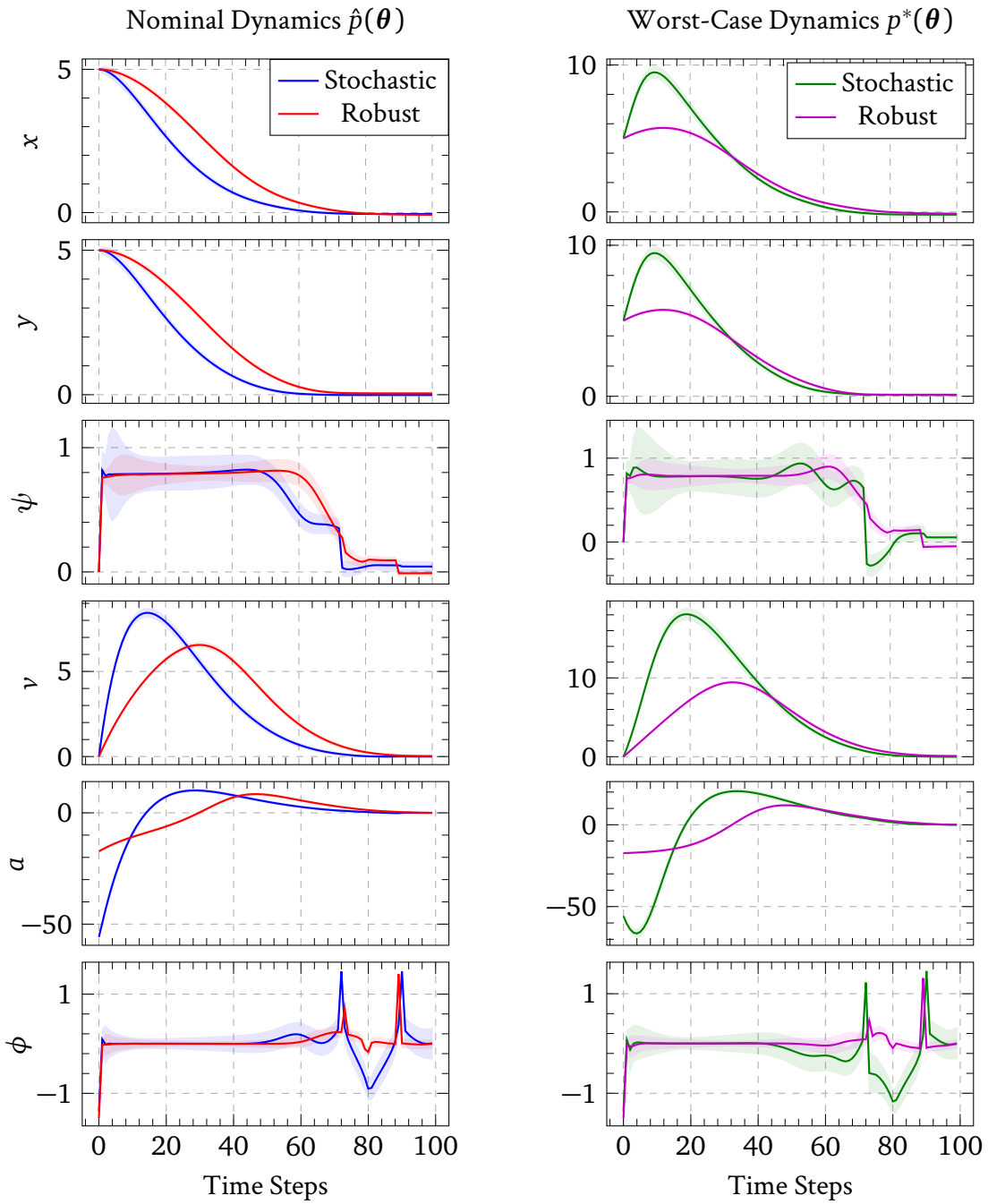


Figure 4.4: Uncertain nonlinear robot experiment. Comparison of standard and distributionally robust controllers. Left, the trajectory induced by the standard (blue) and robust (red) controllers evaluated under the nominal dynamics distribution. The uncertainty-aware controller takes advantage of the nominal dynamics and applies large controls to reach the target faster. Right, the trajectory distributions induced by standard (green) and robust (magenta) controllers evaluated under the worst-case disturbance. The uncertainty-aware controller shows clear sub-optimal behavior, while the robust controller is barely affected.

deviation $\sigma_{\theta} = 1 \times 10^{-3}$. We initialize a zero-mean controller with a diagonal standard deviation $\sigma_{\pi} = \sqrt{0.1}$ and set the trust regions to $\varepsilon = 0.25$ and $\delta = 500$.

Figure 4.4 depicts the results in a similar fashion to what we presented in the last experiment. Here again, the uncertainty-aware controller (blue) acts aggressively under the nominal dynamics, while the robust controller (red) is slower and applies smaller controls. When evaluating the controllers under the uncertainty-aware controller's optimal adversary, the uncertainty-aware controller (green) overshoots and shows sub-optimal behavior, while the trajectory distribution induced by the robust controller (magenta) is hardly affected. Lastly, the comparison of both controllers on a set of distributions interpolated between the nominal and the adversary highlights the overwhelming advantage of the robust controller, Figure 4.3.

4.7 Discussion

We have presented a technique to robustify data-driven stochastic optimal control approaches that rely on probabilistic models of the dynamics. Our approach consists of an iterative two-stage relative entropy trust region optimization. The first stage optimizes the maximum entropy worst-case Gaussian distributional dynamics in a KL-ball around a nominal distribution, while the second stage optimizes the policy with respect to the worst-case dynamics. We show that both stages admit closed-form backward value recursions and approximate cubature forward passes for probabilistic time-variant dynamics models. Furthermore, empirical results on linear and nonlinear dynamical systems validate the benefits of robustifying stochastic control against worst-case model disturbances.

Despite the encouraging initial results, our approach still has multiple limitations. The assumption of Gaussian densities for the nominal and worst-case distributions is rather limiting. Similarly, although reasonable, the restriction of the adversary to a time-variant form does not reflect the statistical errors that arise while approximating stationary representations of dynamics. In addition, long-horizon trajectory optimization is often prone to get stuck in local minima. An investigation of a nonlinear model predictive control formulation can prove very beneficial, despite the additional computational load it may require. Finally, the KL divergence is not a proper distance metric in the space of distributions. Analyzing the drawbacks of this design choice can inspire better alternatives, e.g., using kernel methods and optimal transport.

4.8 Filtering in Markov Jump Systems

Continuing the central theme of focusing on structured models, we discuss how the previously presented multi-stage trust region optimization inspires cost-oriented state inference methods and aids tractable control of stochastic switching systems.

We consider the discrete-time stochastic optimal control problem for Markov jump linear systems (MJLS) with quadratic costs. Although MJLS admit tractable closed-form optimal control computation (Fragoso, 1989), we propose a trust region formulation in order to account for an iterative data-driven process of dynamics learning. This formulation can be considered equivalent to the trust region regularized linear-quadratic case, albeit with an augmented state vector containing a discrete component. Remember, adding a trust region leads to a forward-backward solution scheme (Levine & Koltun, 2013; Arenz et al., 2016; Abdulsamad et al., 2017).

In this section, we do not focus on the whole control optimization problem. Instead, we are interested in the issue of mixture-state propagation during the forward recursion. Propagating a state distribution through discrete-continuous dynamics results in an explosion in the number of mixture components after a few steps. This effect poses a significant challenge for long-horizon planning in stochastic switching systems.

Common solutions to the mixture explosion problem rely on approximate Gaussian reduction techniques (Crouse et al., 2011) with the generalized pseudo Bayesian approximation (GPB) (Kim, 1994) and interacting multiple model (IMM) filtering (Blom & Bar-Shalom, 1988) being prominent examples. However, these methods are used mainly in state estimation scenarios and take no consideration of a cost minimization objective.

The following sections outline an approach that augments stochastic control optimization with a cost-oriented Gaussian mixture reduction technique. The overall method closely resembles the two-stage optimization problems discussed in this chapter. By incorporating the cost into the state estimation scheme, we can interpolate between variational approximations on a scale ranging from optimistic to pessimistic estimates with respect to cost.

4.8.1 Switching Stochastic Optimal Control

Let's assume a finite-horizon MDP with a continuous state space $\mathbf{X} \subseteq \mathbb{R}^d$, a set of one hot discrete state vectors $\mathbf{z} \in \{0,1\}^K : \sum_{k=1}^K z_k = 1$, and an action space $\mathbf{U} \subseteq \mathbb{R}^m$. The switching state transition follows the standard Markov jump process definition as a stochastic function with a density $p(\mathbf{x}', \mathbf{z}'|\mathbf{x}, \mathbf{u}, \mathbf{z}) = f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \mathbf{z}')h(\mathbf{z}'|\mathbf{z})$.

The set of stochastic hybrid policies $\pi_t(\mathbf{u}|\mathbf{x}, \mathbf{z})$ reflect the discrete-continuous structure and induce the state distribution mixture $\mu_t(\mathbf{x}, \mathbf{z}) = p_t(\mathbf{z})p_t(\mathbf{x}|\mathbf{z})$, where $p_t(\mathbf{z})$ is a categorical distribution and $p_t(\mathbf{x}|\mathbf{z})$ are the individual Gaussian state components. Finally, the cost function $c_t(\mathbf{x}, \mathbf{z}, \mathbf{u})$ is assumed to be quadratic in both state and action, with the dependency an optional on the discrete state $\mathbf{z}$.

The optimization problem solved at every iteration k can be written as

$$\begin{aligned}
\underset{\pi_t}{\text{minimize}} \quad & \sum_{t=1}^{T-1} \sum_{\mathbf{z}} \iint c_t(\mathbf{x}, \mathbf{u}, \mathbf{z})\, \mu_t(\mathbf{x}, \mathbf{z}) \pi_t^{k+1}(\mathbf{u}|\mathbf{x}, \mathbf{z})\, \mathrm{d}\mathbf{x}\, \mathrm{d}\mathbf{u} + \int c_T(\mathbf{x}, \mathbf{z}) \mu_T(\mathbf{x}, \mathbf{z})\, \mathrm{d}\mathbf{x}, \\
\text{subject to} \quad & \sum_{\mathbf{z}} \iiint \mu_t(\mathbf{x}, \mathbf{z})\, \pi_t^{k+1}(\mathbf{u}|\mathbf{x}, \mathbf{z})\, f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \mathbf{z}')\, h(\mathbf{z}'|\mathbf{z})\, \mathrm{d}\mathbf{u}\, \mathrm{d}\mathbf{x} = \mu_{t+1}(\mathbf{x}', \mathbf{z}'), \\
& \sum_{t=1}^{T-1} \sum_{\mathbf{z}} \int \mu_t(\mathbf{x}, \mathbf{z})\, \mathrm{KL}(\pi_t^{k+1}(\mathbf{u}|\mathbf{x}, \mathbf{z}) \,||\, \pi_t^k(\mathbf{u}|\mathbf{x}, \mathbf{z}))\, \mathrm{d}\mathbf{x} \leq \varepsilon, \\
& \int \pi_t^{k+1}(\mathbf{u}|\mathbf{x}, \mathbf{z})\, \mathrm{d}\mathbf{u} = 1, \quad \mu_1(\mathbf{x}, \mathbf{z}) = \hat{\mu}_1(\mathbf{x}, \mathbf{z}).
\end{aligned}$$

This problem can be solved via the method of Lagrangian multipliers (Boyd & Vandenberghe, 2004) and yields the forward-backward recursions to compute the value function $V_t(\mathbf{x}, \mathbf{z})$ and state distribution mixtures $\mu_t(\mathbf{x}, \mathbf{z})$. As previously highlighted, the forward recursion is computationally intractable for moderate and long horizons since the number of mixture components increases exponentially over time.

4.8.2 Optimistic and Pessimistic State Propagation

Nonetheless, to perform the forward recursion, we can fall back on Gaussian sum filtering methods such as IMM filters and GPB, which collapse the state distribution into an approximate Gaussian mixture with a smaller number of components. However, these approaches do not reflect the cost landscape of the control objective and may lead to unpredictable and large deviations from the true expected cost due to their coarse approximations.

We propose to explicitly incorporate the cost into Gaussian sum filtering algorithms by formulating a secondary optimization problem to find the worst- or best-case mixture weights $q_t(\mathbf{z})$ in a trust region around the nominal distribution $p_t(\mathbf{z})$. This approach leads to a more accentuated weighting of the individual components and potentially a complete suppression of some. Mathematically, we write the following

$$\begin{aligned}
\min_{q_t} / \max_{q_t} \quad & \iint c_t(\mathbf{x}, \mathbf{u}, \mathbf{z})\, q_t(\mathbf{z})\, p_t(\mathbf{x}|\mathbf{z})\, \pi_t^{k+1}(\mathbf{u}|\mathbf{x}, \mathbf{z})\, \mathrm{d}\mathbf{x}\, \mathrm{d}\mathbf{u}, \\
\text{subject to} \quad & \mathrm{KL}(q_t(\mathbf{z}) \,||\, p_t(\mathbf{z})) \leq \varepsilon, \\
& \sum_{\mathbf{z}} q_t(\mathbf{z}) = 1,
\end{aligned}$$

where $\mu_t(\mathbf{x}, \mathbf{z}) = p_t(\mathbf{z}) p_t(\mathbf{x}|\mathbf{z})$ is the state distribution approximation at the current time step, and ε controls the trust region size. Notice that we account for maximization and minimization objectives that result in pessimistic and optimistic weights, respectively.

The previous optimization is solved by constructing the Lagrangian and solving for $q_t(\mathbf{z})$ to get the optimal point

$$q_t^*(\mathbf{z}) \propto p_t(\mathbf{z}) \exp\left[\frac{1}{\lambda} \int\int c_t(\mathbf{x}, \mathbf{u}, \mathbf{z})\, \pi_t^{k+1}(\mathbf{u}|\mathbf{x}, \mathbf{z})\, d\mathbf{x}\, d\mathbf{u}\right],$$

where λ is the Lagrangian multiplier associated with the trust region constraint. Note that $\lambda \leq 0$ for the maximization objective and $\lambda \geq 0$ for the minimization variant. This solution is computed in closed-form for a given scalar value of λ, a quadratic cost $c_t(\mathbf{x}, \mathbf{u})$, and linear Gaussian hybrid policies $\pi_t(\mathbf{u}|\mathbf{x}, \mathbf{z})$.

4.8.3 Qualitative Examples

We illustrate the effect of the proposed weight optimization on a simple example of a stationary mixture of Gaussians at a time slice t. We assume a quadratic cost $c_t(\mathbf{x})$ centered at zero and analyze the trust region effect by modulating the variable λ directly. Without loss of generality, we can discard the role of the policy $\pi_t(\mathbf{x}, \mathbf{u})$ in this example.

The initial results confirm the motivation behind this approach. In Figure 4.5, we assume an optimistic approximation scenario and observe the tendency to suppress mixture components that are far from the minimum cost, around zero, and thus incur higher costs. In contrast, Figure 4.5 depicts the results of a pessimistic approximation which leads to the opposite effect. Gaussian components close to the minimum cost are heavily dampened, while the others are amplified.

Although these results are encouraging, further validation and more elaborate settings are required. For example, an evaluation of a long-horizon control task and comparing the cost objective under pessimistic, optimistic, and Monte Carlo estimates of the state distribution can reveal the magnitude of approximation errors induced by Gaussian mixture reduction techniques and their overall effects on planning.

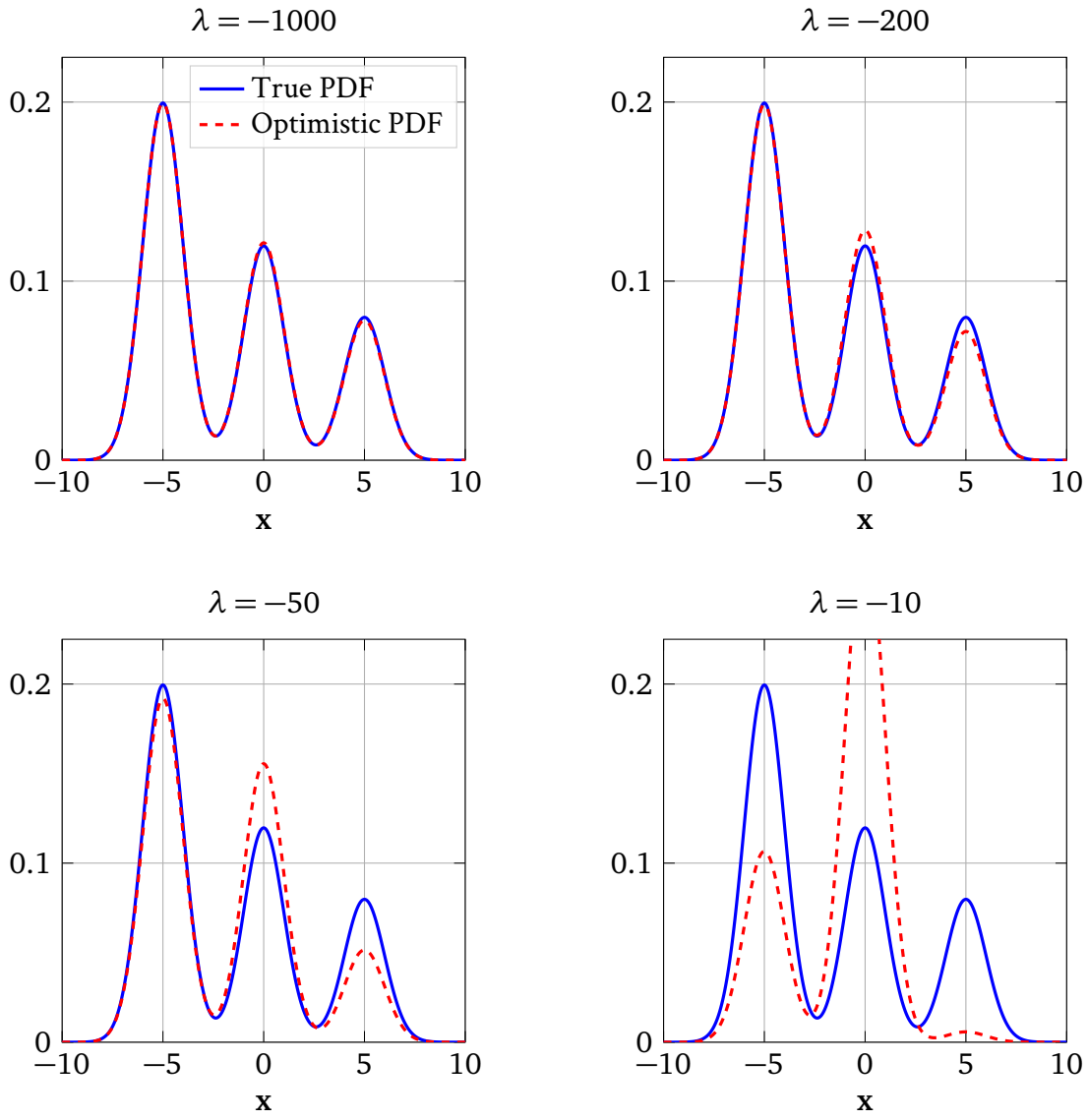


Figure 4.5: The effect of modulating the weights of Gaussian mixture through a cost-optimistic optimization. Large absolute values of λ correspond to a small trust region and small deviation from the reference weights and overall mixture. By lowering $|\lambda|$, we observe the gradual dampening of Gaussian components that correlate with higher costs, while components in lower-cost regions are amplified to account for the shifting of probability mass.

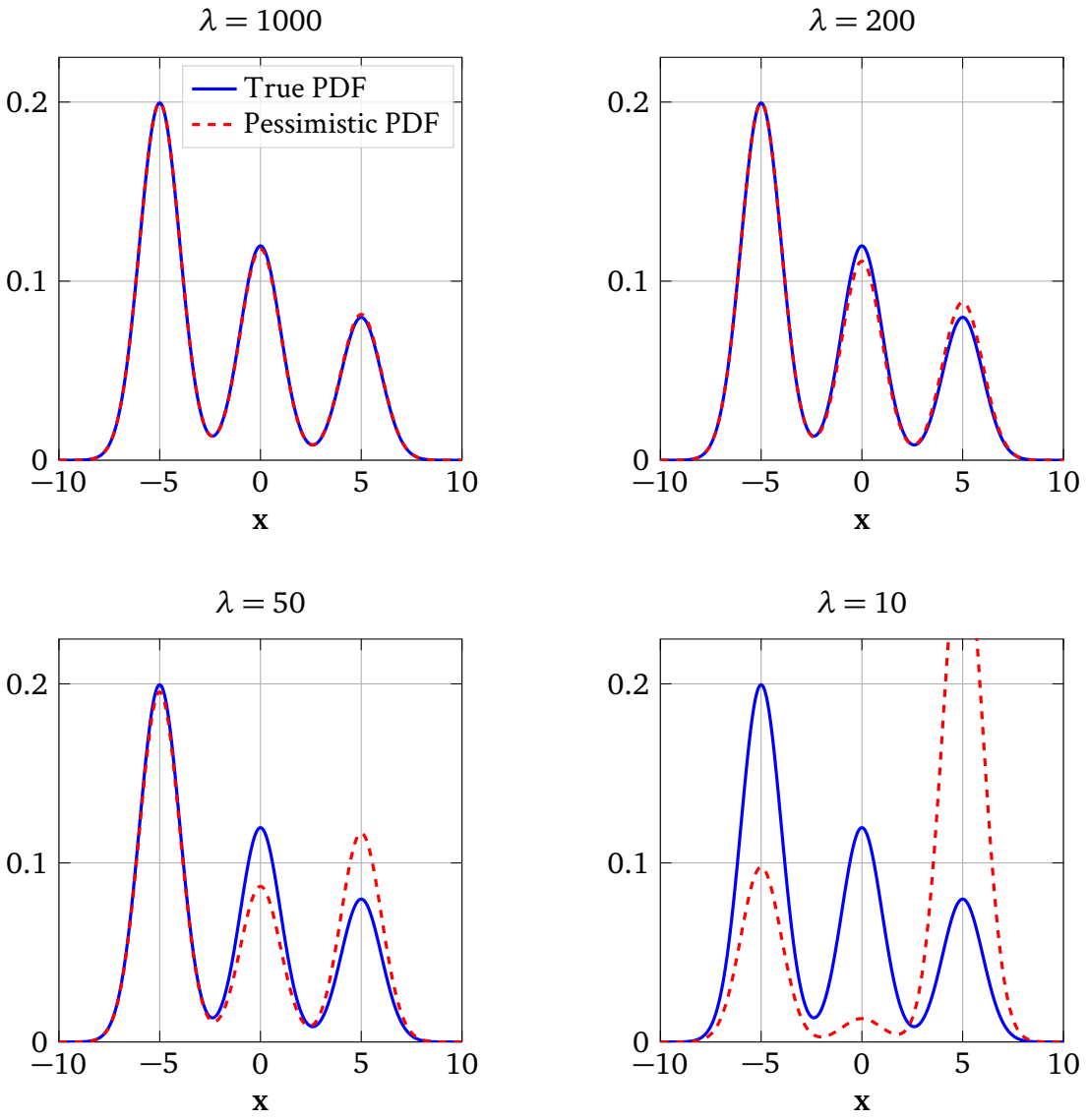


Figure 4.6: The effect of modulating the weights of Gaussian mixture through a cost-pessimistic optimization. Large values of λ correspond to a small trust region and small deviation from the reference weights and overall mixture. By lowering λ, we observe the gradual dampening of Gaussian components that correlate with lower costs, while components in higher-cost regions are amplified to account for the shifting of probability mass.

Chapter 5
Conclusion

This manuscript is the culmination of the main themes that dominated my doctoral studies. The topics covered in the previous chapters span a wide intersection area of Bayesian statistics, optimal control, hierarchical modeling, reinforcement learning, and robust optimization. These are foundational topics for achieving a level of intelligent automation that can effectively interact with the physical world.

The primary drive behind this research has often been grounded in a philosophical view of complexity that emerges from a transparent structure of simple cooperating local units and rules. This view, although fascinating, is exceptionally challenging, as it poses complex questions about the proper levels of abstraction. Furthermore, this hierarchical paradigm stands for an understanding of intelligence as a structured mixed program of symbolic and continuous processes, which, in our opinion, at the moment cannot rise to the level of achievements reached by massive black-box differentiable learning automata. One observation made throughout this research that may explain this current trend is the explosion in algorithmic complexity that accompanies structured modeling paradigms, in contrast to simple learning principles in opaque differentiable machines.

Nonetheless, hierarchical processes have a natural legitimacy that arises from their inherent ability to model discrete phenomena and compress representation by abstracting over repeating patterns. Moreover, a structured model delivers automatic complexity regularization by offering a mold for the data to fit in. However, this property is a double-edged sword, as it is often the case that a sub-optimal structure choice can severely restrict the power of a model to capture the underlying data patterns.

Reflecting on these aspects is and has been a matter of life-long research. The previous chapters are a first step focused on a collection of ideas to leverage structure and hierarchy in different scenarios of optimal control and reinforcement learning. In the following, we summarize the contributions of the individual chapters, draw conclusions, and highlight open questions and potential extensions.

5.1 Summary

In Chapter 2, we set out to construct large flexible regression models that can adapt their complexity according to the data. This objective implies the need for representations that grow their parametric structure to integrate information that the current state of the model cannot explain. Bayesian nonparametric statistics offers the most principled approach to tackling this problem by realizing infinite-dimensional stochastic processes such as the

Dirichlet process. Inspired by these concepts, we formulated two infinite mixtures of local regression models that upended the need for heuristics to extend or prune the structure and enjoy a set of amenable properties. On the one hand, these representations offer a compromise between memory-intensive kernel machines and common rigid parametric models. On the other hand, they maintain a probabilistic generative view of the data that supports a continual learning setting and avoids catastrophic forgetting. The Bayesian formulation is critical for regularizing large parametric models. This insight led us to rely on scalable variational inference techniques to infer the posterior parameters of the proposed representations. Furthermore, we presented a wide range of evaluations to highlight the capabilities of these models in approximating non-differentiable functions, dealing with heteroscedastic noise, and admitting sequential Bayesian updates. Finally, we used these models in large-scale experiments to learn the inverse dynamics of anthropomorphic manipulators and leveraged them in real-world control scenarios.

Chapter 3 focuses on modeling time series data and infinite-horizon control of nonlinear dynamical systems. This area of research is traditionally dominated by neural representations of value functions and policies. Our objective was to investigate the potential of employing a hierarchical approach to system identification and control. Thus, we adopted a data-driven view of hybrid systems as a structured modeling paradigm of general nonlinear dynamics. This decision was motivated by a certain interpretation of neural networks with rectified linear unit activations that views them as discriminative mixtures of interacting but often redundant local experts. Our approach relies on making hidden hierarchies visible in order to avoid over-parameterization. We used augmented hidden Markov models to capture and decompose the temporal dynamics of a control loop into linear sub-regions. In addition, we designed a reinforcement learning algorithm that explicitly integrates the hybrid dynamics and optimizes hierarchical polynomial policies and value functions. We empirically evaluated the ability of the proposed representation and demonstrated that it is able to capture long-horizon trends sufficiently well while drastically reducing parametric complexity. Moreover, the hierarchical reinforcement learning technique delivered encouraging results on common control tasks and carries potential as an alternative to common hybrid control techniques based on finite-horizon optimal control.

Lastly, Chapter 4 revolves around stochastic optimal control and the issues of its sensitivity w.r.t. to models learned from data. The current trend of incorporating statistical representations of dynamics into control frameworks has unleashed a significant potential for scalability to more complex environments. However, in scenarios where little data is available, the brittle validity of inferred models poses a severe challenge to optimality in general and risk-sensitive applications in specific. We proposed using the concept of distributional robustness as a mathematical framework for dealing with these concerns. We presented a formalism to robustify stochastic optimal control against statistical biases of probabilistic dynamics models. We formulated an iterative minimax relative entropy trust-region op-

timization. This approach alternates between finding the worst-case distribution over the dynamics in a trust region in the vicinity of a nominal distribution and optimizing a robust policy w.r.t. said the worst-case dynamics. Notably, we demonstrate that these steps admit closed-form backward recursions for time-variant linearized probabilistic dynamics. Our validation results illustrate the risk of optimizing controllers under model mismatch and how our treatment based on distributional robustness can mitigate the effects of worst-case statistical disturbances. Finally, we highlighted how a related two-stage optimization formulation could help tackle tractability issues of state estimation in stochastic switching linear systems and provided qualitative examples.

5.2 Outlook

We have successfully demonstrated the significance of previously presented concepts with a multitude of evaluations. However, as common in experimental machine learning, these approaches still lack the generality and scalability required for deployment and broad adoption. What follows is a critical view of this thesis' contributions and a reflection on the future steps that may build upon its ideas constructively.

A general point can be made on whether the algorithmic effort connected to constructing and learning hierarchical representations is always justifiable. The practical algorithms developed in this thesis are often considerably more complex than widely adopted standard solutions. Although we have highlighted the general benefits of each contribution, their importance might vary depending on the application domain.

On a technical level, there are critical challenges deeply rooted in the inference paradigms of structured models. For instance, expectation-maximization algorithms are notoriously sensitive to their initial conditions and do not scale well to higher dimensions and many components. In contrast, Variational Bayes approaches are less fragile due to the priors' regularizing effects. Nonetheless, they still suffer from approximation errors due to the mean-field assumptions. This concern can be addressed by adopting collapsed mixture formulations that reduce the posterior approximation gap (Kurihara et al., 2007).

Moreover, Bayesian model design, in general, is profoundly affected by prior misspecification. Too broad prior definitions lead to under-fitting and poor posterior predictive performance. Empirical Bayes approaches, although controversial, as they undermine the Bayesian principle, can remedy this issue by optimizing the priors after the fact. However, structured models often do not admit tractable empirical Bayes computation. Nevertheless, despite these significant difficulties, relaxation strategies of discrete variables may offer a way to perform prior optimization (Jang et al., 2016).

Concerning design decisions, most models presented in this thesis assume a structure to be directly available in the data space in one way or another. For example, the hierarchical infinite mixtures discussed in Chapter 2 cluster the activations in the input space, and the

output is modeled via a direct linear dependency on the input. Such assumptions may prove inefficient in some cases. One may envision a scenario in which the data is more compactly structured on a higher-dimensional manifold. Incorporating and automatically learning nonlinear projections of the data can lead to further compression of these representations (Iwata et al., 2013).

On the other hand, the hidden Markov models used in Chapter 3 do, in fact, rely on nonlinear embeddings of discrete switching probabilities. However, the individual linear regions still operate under the assumptions of a fully observable state, which is used to model continuous dynamics. A more general framework of hybrid control can be achieved by inferring the switching dynamics in a latent space (Becker-Ehmck et al., 2019). In addition, a fully Bayesian treatment of these models could make the learning process more reliable and improve scalability (Beal et al., 2002; Wenzel et al., 2019).

In addition, it is common to assume a data collection process independent of the learning process. Generating data may involve uniform sampling or sinusoidal excitation of a dynamical system. These approaches are often either expensive and inefficient or plainly sub-optimal. More principled methods couple data generation and learning by actively seeking information that improves the model's predictive accuracy (Aoki, 1967; Schultheis et al., 2020). We postulate that certain structured representations may be exceptionally compatible with such learning strategies.

One more interesting subject of research is to consider viewing hierarchical control through the lens of control-as-inference. The hybrid reinforcement learning approach we presented in Chapter 3 heavily relies on tractable inference in hidden Markov models. We posit that the corresponding graphical models can be readily extended to directly incorporate an external reward signal, thus creating a unifying framework for action optimization by relying on Bayesian inference principles (Toussaint & Storkey, 2006; Toussaint, 2009; Hoffman et al., 2013; Watson et al., 2020a).

Finally, the distributionally robust optimal control algorithm we presented in Chapter 4 can benefit from several extensions. On the one hand, the limitations of Gaussian nominal densities and time-varying linearized dynamics can be done without. Instead, an iterative model-free trajectory optimization in the style of (Akrour et al., 2016) appears to be within reach. On the other hand, forgoing the idea of a parametric nominal density in favor of empirical reference distributions that directly incorporate the data, although computationally challenging, may offer a more general optimization framework that circumvents the need for a parametric representation of the dynamics.

Appendix A
Bayesian Posteriors

A.1 Categorical with a Dirichlet Prior

Likelihood: Assuming a one-hot random variable $\mathbf{z}$ of size K

$$\begin{aligned} p(\mathbf{Z}|\boldsymbol{\pi}) &= \prod_{n=1}^{N} \text{Cat}(\mathbf{z}_n|\boldsymbol{\pi})^{w_{nk}} \\ &= \prod_{n=1}^{N}\prod_{k=1}^{K} (\pi_k^{z_{nk}})^{w_{nk}} \\ &\propto \exp\left\{ \begin{bmatrix} \log \pi_1 \\ \vdots \\ \log \pi_K \end{bmatrix} \cdot \begin{bmatrix} \sum_{n=1}^{N} w_{n,1} \\ \vdots \\ \sum_{n=1}^{N} w_{n,K} \end{bmatrix} \right\} \end{aligned}$$

Prior:

$$\begin{aligned} p(\boldsymbol{\pi}) &= \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\tau}_0) \\ &\propto \prod_{k=1}^{K} \pi_k^{\tau_{0,k}} \\ &= \exp\left\{ \begin{bmatrix} \tau_{0,1} - 1 \\ \vdots \\ \tau_{0,K} - 1 \end{bmatrix} \cdot \begin{bmatrix} \log \pi_1 \\ \vdots \\ \log \pi_K \end{bmatrix} \right\}, \end{aligned}$$

Posterior:

$$\begin{aligned} q(\boldsymbol{\pi}) &= \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\tau}) \\ &\propto \exp\left\{ \begin{bmatrix} \tau_{0,1} - 1 + \sum_{n=1}^{N} w_{n,1} \\ \vdots \\ \tau_{0,K} - 1 + \sum_{n=1}^{N} w_{n,K} \end{bmatrix} \cdot \begin{bmatrix} \log \pi_1 \\ \vdots \\ \log \pi_K \end{bmatrix} \right\}. \end{aligned}$$

A.2 Infinite Categorical with a Stick-Breaking Prior

Likelihood: Assuming a one-hot random variable $\mathbf{z}$ of infinite size

$$
\begin{aligned}
p(\mathbf{Z}|\pi(\mathbf{s})) &= \prod_{n=1}^{N} \text{Cat}(\mathbf{z}_n|\pi(\mathbf{s}))^{w_{nk}} \\
&= \prod_{n=1}^{N}\prod_{k=1}^{\infty}\left[\left(s_k\prod_{l=1}^{k-1}(1-s_l)\right)^{z_{nk}}\right]^{w_{nk}}
\end{aligned}
$$

Prior:

$$
\begin{aligned}
p(\mathbf{s}) &= \prod_{k=1}^{\infty}\text{Beta}(s_k|\gamma_0,\alpha_0) \\
&\propto \prod_{k=1}^{\infty} s_k^{\gamma_0-1}\left(1-s_k\right)^{\alpha_0-1}
\end{aligned}
$$

Truncated Posterior: (Blei & Jordan, 2006)

$$
q(\mathbf{s}) = \prod_{k=1}^{K-1}\text{Beta}(s_k|\gamma_k,\alpha_k)
$$

where

$$
\begin{aligned}
\gamma_k &= \gamma_0 + \sum_{n=1}^{N} w_{nk} \\
\alpha_k &= \alpha_0 + \sum_{n=1}^{N}\sum_{l=k+1}^{K} w_{nl}
\end{aligned}
$$

A.3 Gaussian with a Normal-Wishart Prior

Likelihood: Assuming a random variable $\mathbf{x} \in \mathbb{R}^m$

$$
\begin{aligned}
p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Lambda}) &= \prod_{n=1}^{N} \mathrm{N}(\mathbf{x}_n|\boldsymbol{\mu}, \boldsymbol{\Lambda})^{w_n} \\
&= \prod_{n=1}^{N} \left[(2\pi)^{-m/2} |\boldsymbol{\Lambda}|^{1/2} \exp\left\{ -\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\mathbf{x}_n - \boldsymbol{\mu}) \right\} \right]^{w_n} \\
&\propto |\boldsymbol{\Lambda}|^{N_w/2} \exp\left\{ -\frac{1}{2} \sum_{n}^{N} w_n (\mathbf{x}_n - \boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\mathbf{x}_n - \boldsymbol{\mu}) \right\}, \; N_w = \sum_{n=1}^{N} w_n, \\
&= |\boldsymbol{\Lambda}|^{N_w/2} \exp\left\{ -\frac{1}{2} \mathrm{tr}\left(\boldsymbol{\Lambda} \sum_{n=1}^{N} w_n \mathbf{x}_n \mathbf{x}_n^\top - 2\boldsymbol{\mu}^\top \boldsymbol{\Lambda} \sum_{n=1}^{N} w_n \mathbf{x}_n + N_w \boldsymbol{\mu}^\top \boldsymbol{\Lambda} \boldsymbol{\mu} \right) \right\} \\
&= \exp\left\{ \begin{bmatrix} \boldsymbol{\Lambda}\boldsymbol{\mu} \\ \boldsymbol{\Lambda} \end{bmatrix} : \begin{bmatrix} \sum_{n=1}^{N} w_n \mathbf{x}_n \\ -\frac{1}{2} \sum_{n=1}^{N} w_n \mathbf{x}_n \mathbf{x}_n^\top \end{bmatrix} + \begin{bmatrix} \boldsymbol{\mu}^\top \boldsymbol{\Lambda} \boldsymbol{\mu} \\ \log|\boldsymbol{\Lambda}| \end{bmatrix} : \begin{bmatrix} -\frac{1}{2} N_w \\ \frac{1}{2} N_w \end{bmatrix} \right\} \\
&= \exp\left\{ \begin{bmatrix} \boldsymbol{\Lambda}\boldsymbol{\mu} \\ \boldsymbol{\mu}^\top \boldsymbol{\Lambda} \boldsymbol{\mu} \\ \boldsymbol{\Lambda} \\ \log|\boldsymbol{\Lambda}| \end{bmatrix} : \begin{bmatrix} \sum_{n=1}^{N} w_n \mathbf{x}_n \\ -\frac{1}{2} N_w \\ -\frac{1}{2} \sum_{n=1}^{N} w_n \mathbf{x}_n \mathbf{x}_n^\top \\ \frac{1}{2} N_w \end{bmatrix} \right\}
\end{aligned}
$$

Prior:

$$
\begin{aligned}
p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) &= \mathrm{N}(\boldsymbol{\mu}|\mathbf{m}_0, \kappa_0 \boldsymbol{\Lambda}) \; \mathrm{W}(\boldsymbol{\Lambda}|\boldsymbol{\Psi}_0, \nu_0) \\
&\propto |\kappa_0 \boldsymbol{\Lambda}|^{1/2} \exp\left\{ -\frac{\kappa_0}{2} (\boldsymbol{\mu} - \mathbf{m}_0)^\top \boldsymbol{\Lambda} (\boldsymbol{\mu} - \mathbf{m}_0) \right\} |\boldsymbol{\Lambda}|^{\frac{\nu_0 - m - 1}{2}} \exp\left\{ -\frac{1}{2} \mathrm{tr}(\boldsymbol{\Psi}_0{}^{-1} \boldsymbol{\Lambda}) \right\} \\
&\propto \exp\left\{ -\frac{1}{2} \mathrm{tr}\left(\kappa_0 \boldsymbol{\mu}^\top \boldsymbol{\Lambda} \boldsymbol{\mu} - 2\kappa_0 \mathbf{m}_0 \boldsymbol{\mu}^\top \boldsymbol{\Lambda} + \kappa_0 \mathbf{m}_0 \mathbf{m}_0^\top \boldsymbol{\Lambda} + \boldsymbol{\Psi}_0^{-1} \boldsymbol{\Lambda} - (\nu_0 - m) \log|\boldsymbol{\Lambda}| \right) \right\} \\
&= \exp\left\{ \begin{bmatrix} \kappa_0 \mathbf{m}_0 \\ -\frac{1}{2} \kappa_0 \\ -\frac{1}{2} (\boldsymbol{\Psi}_0^{-1} + \kappa_0 \mathbf{m}_0 \mathbf{m}_0^\top) \\ \frac{1}{2} (\nu_0 - m) \end{bmatrix} : \begin{bmatrix} \boldsymbol{\Lambda}\boldsymbol{\mu} \\ \boldsymbol{\mu}^\top \boldsymbol{\Lambda} \boldsymbol{\mu} \\ \boldsymbol{\Lambda} \\ \log|\boldsymbol{\Lambda}| \end{bmatrix} \right\}
\end{aligned}
$$

Posterior:

$$
\begin{aligned}
q(\boldsymbol{\mu}, \boldsymbol{\Lambda}) &= \mathrm{N}(\boldsymbol{\mu}|\mathbf{m}, \kappa\boldsymbol{\Lambda})\ \mathrm{W}(\boldsymbol{\Lambda}|\boldsymbol{\Psi}, \nu) \\
&\propto \exp\left\{ \begin{bmatrix} \boldsymbol{\Lambda}\boldsymbol{\mu} \\ \boldsymbol{\mu}^\top\boldsymbol{\Lambda}\boldsymbol{\mu} \\ \boldsymbol{\Lambda} \\ \log|\boldsymbol{\Lambda}| \end{bmatrix} : \begin{bmatrix} \sum_{n=1}^N w_n\mathbf{x}_n \\ -\frac{1}{2}N_w \\ -\frac{1}{2}\sum_{n=1}^N w_n\mathbf{x}_n\mathbf{x}_n^\top \\ \frac{1}{2}N_w \end{bmatrix} \right\} \\
&\quad \times \exp\left\{ \begin{bmatrix} \kappa_0\mathbf{m}_0 \\ -\frac{1}{2}\kappa_0 \\ -\frac{1}{2}(\boldsymbol{\Psi}_0^{-1} + \kappa_0\mathbf{m}_0\mathbf{m}_0^\top) \\ \frac{1}{2}(\nu_0 - m) \end{bmatrix} : \begin{bmatrix} \boldsymbol{\Lambda}\boldsymbol{\mu} \\ \boldsymbol{\mu}^\top\boldsymbol{\Lambda}\boldsymbol{\mu} \\ \boldsymbol{\Lambda} \\ \log|\boldsymbol{\Lambda}| \end{bmatrix} \right\} \\
&= \exp\left\{ \begin{bmatrix} \kappa_0\mathbf{m}_0 + \sum_{n=1}^N w_n\mathbf{x}_n \\ -\frac{1}{2}(\kappa_0 + N_w) \\ -\frac{1}{2}(\boldsymbol{\Psi}_0^{-1} + \kappa_0\mathbf{m}_0\mathbf{m}_0^\top + \sum_{n=1}^N w_n\mathbf{x}_n\mathbf{x}_n^\top) \\ \frac{1}{2}(\nu_0 - m + N_w) \end{bmatrix} : \begin{bmatrix} \boldsymbol{\Lambda}\boldsymbol{\mu} \\ \boldsymbol{\mu}^\top\boldsymbol{\Lambda}\boldsymbol{\mu} \\ \boldsymbol{\Lambda} \\ \log|\boldsymbol{\Lambda}| \end{bmatrix} \right\}
\end{aligned}
$$

The operation $A{:}B$ stands for a double tensor contraction, or a double dot product, between two tensors A and B. This operation is a generalization of the trace $\mathrm{tr}(A^\top B) = A{:}B$.

A.4 Tied Gaussians with Normal-Wishart Priors

Likelihood: Assuming a random variable $\mathbf{x} \in \mathbb{R}^m$ governed by K precision-tied components

$$
\begin{aligned}
p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Lambda}) &= \prod_{k=1}^{K}\prod_{n=1}^{N} \mathrm{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda})^{w_{nk}} \\
&= \prod_{k=1}^{K}\prod_{n=1}^{N} \left[(2\pi)^{-d/2}|\boldsymbol{\Lambda}|^{1/2} \exp\left\{ -\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}(\mathbf{x}_n - \boldsymbol{\mu}_k) \right\} \right]^{w_{nk}} \\
&\propto \prod_{k=1}^{K} |\boldsymbol{\Lambda}|^{N_k/2} \exp\left\{ -\frac{1}{2}\sum_{n=1}^{N} w_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}(\mathbf{x}_n - \boldsymbol{\mu}_k) \right\}, \; N_k = \sum_{n=1}^{N} w_{nk} \\
&= \prod_{k=1}^{K} |\boldsymbol{\Lambda}|^{N_k/2} \exp\left\{ -\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Lambda}\sum_{n=1}^{N} w_n \mathbf{x}_n \mathbf{x}_n^\top - 2\boldsymbol{\mu}_k^\top \boldsymbol{\Lambda} \sum_{n=1}^{N} w_{nk}\mathbf{x}_n + N_k \boldsymbol{\mu}_k^\top \boldsymbol{\Lambda}\boldsymbol{\mu}_k \right) \right\} \\
&\propto \exp\left\{ \begin{bmatrix} \boldsymbol{\Lambda} \\ \log|\boldsymbol{\Lambda}| \end{bmatrix} : \begin{bmatrix} -\frac{1}{2K}\sum_{k=1}^{K}\sum_{n=1}^{N} w_{nk}\mathbf{x}_n\mathbf{x}_n^\top \\ \frac{1}{2K}\sum_{k=1}^{K} N_k \end{bmatrix} \right\} \\
&\quad \times \prod_{k=1}^{K} \exp\left\{ \begin{bmatrix} \boldsymbol{\Lambda}\boldsymbol{\mu}_k \\ \boldsymbol{\mu}_k^\top \boldsymbol{\Lambda}\boldsymbol{\mu}_k \end{bmatrix} : \begin{bmatrix} \sum_{n=1}^{N} w_{nk}\mathbf{x}_n \\ -\frac{1}{2}N_k \end{bmatrix} \right\}
\end{aligned}
$$

Prior:

$$
\begin{aligned}
p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) &= \mathrm{W}(\boldsymbol{\Lambda}|\boldsymbol{\Psi}_0, \nu_0) \prod_{k=1}^{K} \mathrm{N}(\boldsymbol{\mu}_k|\mathbf{m}_0, \kappa_0\boldsymbol{\Lambda}) \\
&\propto |\boldsymbol{\Lambda}|^{\frac{\nu_0 - d - 1}{2}} \exp\left\{ -\frac{1}{2}\mathrm{tr}(\boldsymbol{\Psi}_0{}^{-1}\boldsymbol{\Lambda}) \right\} \prod_{k=1}^{K} |\kappa_0\boldsymbol{\Lambda}|^{1/2} \exp\left\{ -\frac{\kappa_0}{2}(\boldsymbol{\mu}_k - \mathbf{m}_0)^\top \boldsymbol{\Lambda}(\boldsymbol{\mu}_k - \mathbf{m}_0) \right\} \\
&= \exp\left\{ \begin{bmatrix} -\frac{1}{2}(\boldsymbol{\Psi}_0^{-1} + \frac{\kappa_0}{K}\sum_{k=1}^{K}\mathbf{m}_0\mathbf{m}_0^\top) \\ \frac{1}{2}(\nu_0 - m) \end{bmatrix} : \begin{bmatrix} \boldsymbol{\Lambda} \\ \log|\boldsymbol{\Lambda}| \end{bmatrix} \right\} \\
&\quad \times \prod_{k=1}^{K} \exp\left\{ \begin{bmatrix} \kappa_0\mathbf{m}_0 \\ -\frac{1}{2}\kappa_0 \end{bmatrix} : \begin{bmatrix} \boldsymbol{\Lambda}\boldsymbol{\mu}_k \\ \boldsymbol{\mu}_k^\top \boldsymbol{\Lambda}\boldsymbol{\mu}_k \end{bmatrix} \right\}
\end{aligned}
$$

Posterior:

$$
\begin{aligned}
q(\boldsymbol{\mu}, \boldsymbol{\Lambda}) &= \mathrm{W}(\boldsymbol{\Lambda}|\boldsymbol{\Psi}, \nu) \prod_{k=1}^{K} \mathrm{N}(\boldsymbol{\mu}_k|\mathbf{m}_k, \kappa_k\boldsymbol{\Lambda}) \\
&\propto \exp\left\{ \begin{bmatrix} \boldsymbol{\Lambda} \\ \log|\boldsymbol{\Lambda}| \end{bmatrix} : \begin{bmatrix} -\frac{1}{2K}\sum_{k=1}^{K}\sum_{n=1}^{N} w_{nk}\mathbf{x}_n\mathbf{x}_n^\top \\ \frac{1}{2K}\sum_{k=1}^{K} N_k \end{bmatrix} \right\} \\
&\quad \times \prod_{k=1}^{K} \exp\left\{ \begin{bmatrix} \boldsymbol{\Lambda}\boldsymbol{\mu}_k \\ \boldsymbol{\mu}_k^\top\boldsymbol{\Lambda}\boldsymbol{\mu}_k \end{bmatrix} : \begin{bmatrix} \sum_{n=1}^{N} w_{nk}\mathbf{x}_n \\ -\frac{1}{2}N_k \end{bmatrix} \right\} \\
&\quad \times \exp\left\{ \begin{bmatrix} -\frac{1}{2}(\boldsymbol{\Psi}_0^{-1} + \frac{\kappa_0}{K}\sum_{k=1}^{K}\mathbf{m}_0\mathbf{m}_0^\top) \\ \frac{1}{2}(\nu_0 - m) \end{bmatrix} : \begin{bmatrix} \boldsymbol{\Lambda} \\ \log|\boldsymbol{\Lambda}| \end{bmatrix} \right\} \\
&\quad \times \prod_{k=1}^{K} \exp\left\{ \begin{bmatrix} \kappa_0\mathbf{m}_0 \\ -\frac{1}{2}\kappa_0 \end{bmatrix} : \begin{bmatrix} \boldsymbol{\Lambda}\boldsymbol{\mu}_k \\ \boldsymbol{\mu}_k^\top\boldsymbol{\Lambda}\boldsymbol{\mu}_k \end{bmatrix} \right\} \\
&= \exp\left\{ \begin{bmatrix} -\frac{1}{2}(\boldsymbol{\Psi}_0^{-1} + \frac{\kappa_0}{K}\sum_{k=1}^{K}\mathbf{m}_0\mathbf{m}_0^\top + \frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N} w_{nk}\mathbf{x}_n\mathbf{x}_n^\top) \\ \frac{1}{2}(\nu_0 - m + \frac{1}{K}\sum_{k=1}^{K} N_k) \end{bmatrix} : \begin{bmatrix} \boldsymbol{\Lambda} \\ \log|\boldsymbol{\Lambda}| \end{bmatrix} \right\} \\
&\quad \times \prod_{k=1}^{K} \exp\left\{ \begin{bmatrix} \kappa_0\mathbf{m}_0 + \sum_{n=1}^{N} w_{nk}\mathbf{x}_n \\ -\frac{1}{2}(\kappa_0 + N_k) \end{bmatrix} : \begin{bmatrix} \boldsymbol{\Lambda}\boldsymbol{\mu}_k \\ \boldsymbol{\mu}_k^\top\boldsymbol{\Lambda}\boldsymbol{\mu}_k \end{bmatrix} \right\}
\end{aligned}
$$

A.5 Linear Gaussian with a Matrix-Normal-Wishart Prior

Likelihood: Assuming a conditional model with an input variable $\mathbf{x} \in \mathbb{R}^m$ and a random response $\mathbf{y} \in \mathbb{R}^d$ according to a linear mapping $\mathbf{A} : \mathbb{R}^m \to \mathbb{R}^d$

$$\begin{aligned}
p(\mathbf{Y}|\mathbf{X}, \mathbf{A}, \mathbf{V}) &= \prod_{n=1}^{N} \mathrm{N}(\mathbf{y}_n|\mathbf{x}_n, \mathbf{A}, \mathbf{V})^{w_n} \\
&= \prod_{n=1}^{N} \left[(2\pi)^{-d/2} |\mathbf{V}|^{1/2} \exp\left\{ -\frac{1}{2}(\mathbf{y}_n - \mathbf{A}\mathbf{x}_n)^\top \mathbf{V}(\mathbf{y}_n - \mathbf{A}\mathbf{x}_n) \right\} \right]^{w_n} \\
&\propto |\mathbf{V}|^{N_w/2} \exp\left\{ -\frac{1}{2} \operatorname{tr}\left(\mathbf{V}\mathbf{Y}\mathbf{W}\mathbf{Y}^\top - 2\mathbf{A}^\top \mathbf{V}\mathbf{Y}\mathbf{W}\mathbf{X}^\top + \mathbf{A}^\top \mathbf{V}\mathbf{A}\mathbf{X}\mathbf{W}\mathbf{X}^\top \right) \right\} \\
&= \exp\left\{ \begin{bmatrix} \mathbf{V}\mathbf{A} \\ \mathbf{A}^\top \mathbf{V}\mathbf{A} \\ \mathbf{V} \\ \log|\mathbf{V}| \end{bmatrix} : \begin{bmatrix} \mathbf{Y}\mathbf{W}\mathbf{X}^\top \\ -\frac{1}{2}\mathbf{X}\mathbf{W}\mathbf{X}^\top \\ -\frac{1}{2}\mathbf{Y}\mathbf{W}\mathbf{Y}^\top \\ \frac{1}{2}N_w \end{bmatrix} \right\}
\end{aligned}$$

where $N_w = \sum_{n=1}^{N} w_n$ and $\mathbf{W} = \operatorname{diag}(w_n)$.

Prior:

$$\begin{aligned}
p(\mathbf{A}, \mathbf{V}) &= \mathrm{MN}(\mathbf{A}|\mathbf{M}_0, \mathbf{K}_0, \mathbf{V})\ \mathrm{W}(\mathbf{V}|\mathbf{\Psi}_0, \nu_0) \\
&\propto |\mathbf{V}|^{m/2} \exp\left\{ -\frac{1}{2} \operatorname{tr}\left(\mathbf{K}_0(\mathbf{A} - \mathbf{M}_0)^\top \mathbf{V}(\mathbf{A} - \mathbf{M}_0) \right) \right\} |\mathbf{V}|^{\frac{\nu_0 - d - 1}{2}} \exp\left\{ -\frac{1}{2} \operatorname{tr}(\mathbf{\Psi_0}^{-1}\mathbf{V}) \right\} \\
&= \exp\left\{ -\frac{1}{2} \operatorname{tr}\left(\mathbf{K}_0\mathbf{A}^\top \mathbf{V}\mathbf{A} - 2\mathbf{M}_0\mathbf{K}_0\mathbf{A}^\top \mathbf{V} + \mathbf{K}_0\mathbf{M}_0^\top \mathbf{V}\mathbf{M}_0 + \mathbf{\Psi}_0^{-1}\mathbf{V} - (\nu_0 - d)\log|\mathbf{V}| \right) \right\} \\
&= \exp\left\{ \begin{bmatrix} \mathbf{M}_0\mathbf{K}_0 \\ -\frac{1}{2}\mathbf{K}_0 \\ -\frac{1}{2}(\mathbf{\Psi}_0^{-1} + \mathbf{M}_0\mathbf{K}_0\mathbf{M}_0^\top) \\ \frac{1}{2}(\nu_0 - d - 1 + m) \end{bmatrix} : \begin{bmatrix} \mathbf{V}\mathbf{A} \\ \mathbf{A}^\top \mathbf{V}\mathbf{A} \\ \mathbf{V} \\ \log|\mathbf{V}| \end{bmatrix} \right\}
\end{aligned}$$

Posterior:

$$
\begin{aligned}
q(\mathbf{A},\mathbf{V}) &= \mathrm{MN}(\mathbf{A}|\mathbf{M},\mathbf{K},\mathbf{V})\ \mathrm{W}(\mathbf{V}|\boldsymbol{\Psi},\nu) \\
&\propto \exp\left\{ \begin{bmatrix} \mathbf{VA} \\ \mathbf{A}^\top\mathbf{VA} \\ \mathbf{V} \\ \log|\mathbf{V}| \end{bmatrix} : \begin{bmatrix} \mathbf{YWX}^\top \\ -\frac{1}{2}\mathbf{XWX}^\top \\ -\frac{1}{2}\mathbf{YWY}^\top \\ -\frac{1}{2}N_w \end{bmatrix} \right\} \\
&\quad \times \exp\left\{ \begin{bmatrix} \mathbf{M}_0\mathbf{K}_0 \\ -\frac{1}{2}\mathbf{K}_0 \\ -\frac{1}{2}(\boldsymbol{\Psi}_0^{-1} + \mathbf{M}_0\mathbf{K}_0\mathbf{M}_0^\top) \\ \frac{1}{2}(\nu_0 - d) \end{bmatrix} : \begin{bmatrix} \mathbf{VA} \\ \mathbf{A}^\top\mathbf{VA} \\ \mathbf{V} \\ \log|\mathbf{V}| \end{bmatrix} \right\} \\
&= \exp\left\{ \begin{bmatrix} \mathbf{M}_0\mathbf{K}_0 + \mathbf{YWX}^\top \\ -\frac{1}{2}(\mathbf{K}_0 + \mathbf{XWX}^\top) \\ -\frac{1}{2}(\boldsymbol{\Psi}_0^{-1} + \mathbf{M}_0\mathbf{K}_0\mathbf{M}_0^\top + \mathbf{YWY}^\top) \\ \frac{1}{2}(\nu_0 - d - 1 + m + N_w) \end{bmatrix} : \begin{bmatrix} \mathbf{VA} \\ \mathbf{A}^\top\mathbf{VA} \\ \mathbf{V} \\ \log|\mathbf{V}| \end{bmatrix} \right\}
\end{aligned}
$$

A.6 Tied Lin-Gauss with Matrix-Normal-Wishart Priors

Likelihood: Assuming a conditional model with an input $\mathbf{x} \in \mathbb{R}^m$ and a random response $\mathbf{y} \in \mathbb{R}^d$ according to a linear mapping $\mathbf{A} : \mathbb{R}^m \to \mathbb{R}^d$ and K precision-tied components

$$
\begin{aligned}
p(\mathbf{Y}|\mathbf{X}, \mathbf{A}, \mathbf{V}) &= \prod_{k=1}^{K} \prod_{n=1}^{N} \mathrm{N}(\mathbf{y}_n|\mathbf{x}_n, \mathbf{A}_k, \mathbf{V})^{w_{nk}} \\
&= \prod_{k=1}^{K} \prod_{n=1}^{N} \left[(2\pi)^{-d/2} |\mathbf{V}|^{1/2} \exp\left\{ -\frac{1}{2} (\mathbf{y}_n - \mathbf{A}_k \mathbf{x}_n)^\top \mathbf{V} (\mathbf{y}_n - \mathbf{A}_k \mathbf{x}_n) \right\} \right]^{w_{nk}} \\
&\propto \prod_{k=1}^{K} |\mathbf{V}|^{N_k/2} \exp\left\{ -\frac{1}{2} \operatorname{tr}\left(\mathbf{V}\mathbf{Y}\mathbf{W}_k\mathbf{Y}^\top - 2\mathbf{A}_k^\top \mathbf{V}\mathbf{Y}\mathbf{W}_k\mathbf{X}^\top + \mathbf{A}_k^\top \mathbf{V}\mathbf{A}_k\mathbf{X}\mathbf{W}_k\mathbf{X}^\top \right) \right\} \\
&\propto \exp\left\{ \begin{bmatrix} \mathbf{V} \\ \log|\mathbf{V}| \end{bmatrix} : \begin{bmatrix} -\frac{1}{2K} \sum_{k=1}^{K} \mathbf{Y}\mathbf{W}_k\mathbf{Y}^\top \\ \frac{1}{2K} N_k \end{bmatrix} \right\} \\
&\quad \times \prod_{k=1}^{K} \exp\left\{ \begin{bmatrix} \mathbf{V}\mathbf{A}_k \\ \mathbf{A}_k^\top \mathbf{V}\mathbf{A}_k \end{bmatrix} : \begin{bmatrix} \mathbf{Y}\mathbf{W}_k\mathbf{X}^\top \\ -\frac{1}{2}\mathbf{X}\mathbf{W}_k\mathbf{X}^\top \end{bmatrix} \right\},
\end{aligned}
$$

where $N_k = \sum_{n=1}^{N} w_{nk}$ and $\mathbf{W}_k = \operatorname{diag}(w_{nk})$.

Prior:

$$
\begin{aligned}
p(\mathbf{A}, \mathbf{V}) &= \mathrm{W}(\mathbf{V}|\boldsymbol{\Psi}_0, \nu_0) \prod_{k=1}^{K} \mathrm{N}(\mathbf{A}_k|\mathbf{M}_0, \mathbf{K}_0, \mathbf{V}) \\
&\propto |\mathbf{V}|^{\frac{\nu_0 - d - 1}{2}} \exp\left\{ -\frac{1}{2} \operatorname{tr}(\boldsymbol{\Psi}_0{}^{-1}\mathbf{V}) \right\} \\
&\quad \times \prod_{k=1}^{K} |\mathbf{V}|^{m/2} \exp\left\{ -\frac{1}{2} \operatorname{tr}\left(\mathbf{K}_0 (\mathbf{A}_k - \mathbf{M}_0)^\top \mathbf{V} (\mathbf{A}_k - \mathbf{M}_0) \right) \right\} \\
&= \exp\left\{ \begin{bmatrix} -\frac{1}{2}(\boldsymbol{\Psi}_0^{-1} + \frac{1}{K} \sum_{k=1}^{K} \mathbf{M}_0\mathbf{K}_0\mathbf{M}_0^\top) \\ \frac{1}{2}(\nu_0 - d - 1 + m) \end{bmatrix} : \begin{bmatrix} \mathbf{V} \\ \log|\mathbf{V}| \end{bmatrix} \right\} \\
&\quad \times \prod_{k=1}^{K} \exp\left\{ \begin{bmatrix} \mathbf{M}_0\mathbf{K}_0 \\ -\frac{1}{2}\mathbf{K}_0 \end{bmatrix} : \begin{bmatrix} \mathbf{V}\mathbf{A}_k \\ \mathbf{A}_k^\top \mathbf{V}\mathbf{A}_k \end{bmatrix} \right\}
\end{aligned}
$$

Posterior:

$$
\begin{aligned}
q(\mathbf{A},\mathbf{V}) &= \mathrm{W}(\mathbf{V}|\mathbf{\Psi},\nu)\prod_{k=1}^{K}\mathrm{N}(\mathbf{A}_k|\mathbf{M}_k,\mathbf{K}_k,\mathbf{V}) \\
&\propto \exp\left\{\begin{bmatrix}\mathbf{V}\\ \log|\mathbf{V}|\end{bmatrix} : \begin{bmatrix}-\frac{1}{2K}\sum_{k=1}^{K}\mathbf{Y}\mathbf{W}_k\mathbf{Y}^\top \\ \frac{1}{2K}N_k\end{bmatrix}\right\} \\
&\quad\times\prod_{k=1}^{K}\exp\left\{\begin{bmatrix}\mathbf{V}\mathbf{A}_k\\ \mathbf{A}_k^\top\mathbf{V}\mathbf{A}_k\end{bmatrix} : \begin{bmatrix}\mathbf{Y}\mathbf{W}_k\mathbf{X}^\top \\ -\frac{1}{2}\mathbf{X}\mathbf{W}_k\mathbf{X}^\top\end{bmatrix}\right\} \\
&\quad\times\exp\left\{\begin{bmatrix}-\frac{1}{2}(\mathbf{\Psi}_0^{-1}+\frac{1}{K}\sum_{k=1}^{K}\mathbf{M}_0\mathbf{K}_0\mathbf{M}_0^\top)\\ \frac{1}{2}(\nu_0-d-1+m)\end{bmatrix} : \begin{bmatrix}\mathbf{V}\\ \log|\mathbf{V}|\end{bmatrix}\right\} \\
&\quad\times\prod_{k=1}^{K}\exp\left\{\begin{bmatrix}\mathbf{M}_0\mathbf{K}_0\\ -\frac{1}{2}\mathbf{K}_0\end{bmatrix} : \begin{bmatrix}\mathbf{V}\mathbf{A}_k\\ \mathbf{A}^\top\mathbf{V}\mathbf{A}\end{bmatrix}\right\} \\
&= \exp\left\{\begin{bmatrix}-\frac{1}{2}(\mathbf{\Psi}_0^{-1}+\frac{1}{K}\sum_{k=1}^{K}\mathbf{M}_0\mathbf{K}_0\mathbf{M}_0^\top+\frac{1}{K}\sum_{k=1}^{K}\mathbf{Y}\mathbf{W}_k\mathbf{Y}^\top)\\ \frac{1}{2}(\nu_0-d-1+m+\frac{1}{K}\sum_{k=1}^{K}N_k)\end{bmatrix} : \begin{bmatrix}\mathbf{V}\\ \log|\mathbf{V}|\end{bmatrix}\right\} \\
&\quad\times\prod_{k=1}^{K}\exp\left\{\begin{bmatrix}\mathbf{M}_0\mathbf{K}_0+\mathbf{Y}\mathbf{W}_k\mathbf{X}^\top\\ -\frac{1}{2}(\mathbf{K}_0+\frac{1}{K}\sum_{k=1}^{K}N_k)\end{bmatrix} : \begin{bmatrix}\mathbf{V}\mathbf{A}_k\\ \mathbf{A}_k^\top\mathbf{V}\mathbf{A}_k\end{bmatrix}\right\}
\end{aligned}
$$

Appendix B
Infinite Linear Regression Mixtures

B.1 E-Step of Infinite Linear Regression

$$
\begin{aligned}
\log q(\mathbf{Z}) &= \mathbb{E}_{q(\mathbf{s})}\left[\log p(\mathbf{Z}|\boldsymbol{\pi}(\mathbf{s}))\right] + \mathbb{E}_{q(\boldsymbol{\mu},\boldsymbol{\Lambda})}\left[\log p(\mathbf{X}|\mathbf{Z})\right] \\
&\quad + \mathbb{E}_{q(\mathbf{A},\mathbf{c},\mathbf{V})}\left[\log p(\mathbf{Y}|\mathbf{Z},\mathbf{X})\right] + \text{const} \\
&= \text{const} + \sum_{n=1}^{N}\Bigg[\mathbb{E}_{q(\mathbf{s})}\left[\log \text{Cat}(\mathbf{z}_n|\boldsymbol{\pi}(\mathbf{s}))\right] \\
&\quad + \sum_{k=1}^{K} z_{nk}\mathbb{E}_{q(\boldsymbol{\mu},\boldsymbol{\Lambda})}\left[\log \text{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k)\right] \\
&\quad + \sum_{k=1}^{K} z_{nk}\mathbb{E}_{q(\mathbf{A},\mathbf{c},\mathbf{V})}\left[\log \text{N}(\mathbf{y}_n|\mathbf{A}_k\mathbf{x}_n + \mathbf{c}_k,\mathbf{V}_k)\right]\Bigg] \\
&= \sum_{k=1}^{K}\sum_{n=1}^{N} z_{nk}\log r_{nk},
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbb{E}_{q(\mathbf{s})}\left[\log \text{Cat}(\mathbf{z}_n|\boldsymbol{\pi}(\mathbf{s}))\right] &= \sum_{k=1}^{K} z_{nk}\mathbb{E}_{q(s_k)}\left[\log s_k\right] \\
&\quad + \sum_{k=1}^{K} z_{nk}\mathbb{E}_{q(s_k)}\left[\sum_{l=1}^{k-1}\log(1-s_l)\right].
\end{aligned}
$$

The individual expectations of the *log-sticks* under the beta posteriors are given by

$$
\begin{aligned}
\mathbb{E}_{q(s_k)}\left[\log s_k\right] &= \Psi\left(\gamma_k\right) - \Psi\left(\gamma_k + \alpha_k\right), \\
\mathbb{E}_{q(s_k)}\left[\log(1-s_k)\right] &= \Psi\left(\alpha_k\right) - \Psi\left(\gamma_k + \alpha_k\right),
\end{aligned}
$$

where Ψ is the Digamma function.

B.2 M-Step of Infinite Linear Regression

$$
\begin{aligned}
\log q(\mathbf{s}) &= \mathbb{E}_{q(\mathbf{Z})}\left[\log p(\mathbf{Z}|\pi(\mathbf{s}))\right] + \log p(\mathbf{s}) + \text{const} \\
&= \sum_{k=1}^{K}\sum_{n=1}^{N} r_{nk} \log \left[s_k \prod_{l=1}^{k-1}(1-s_l) \right] \\
&\quad + \sum_{k=1}^{K-1} \log \text{Beta}(s_k|1,\alpha_0) + \text{const}, \\
\log q(\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \mathbb{E}_{q(\mathbf{Z})}\left[\log p(\mathbf{X}|\mathbf{Z})\right] + \log p(\boldsymbol{\mu},\boldsymbol{\Lambda}) + \text{const} \\
&= \sum_{k=1}^{K}\sum_{n=1}^{N} r_{nk} \log \text{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k) \\
&\quad + \sum_{k=1}^{K} \log \text{N}(\boldsymbol{\mu}_k|\mathbf{m}_0,\kappa_0\boldsymbol{\Lambda}_k) \\
&\quad + \sum_{k=1}^{K} \log \text{W}(\boldsymbol{\Lambda}_k|\boldsymbol{\Psi}_0,\nu_0) + \text{const}, \\
\log q(\mathbf{A},\mathbf{c},\mathbf{V}) &= \mathbb{E}_{q(\mathbf{Z})}\left[\log p(\mathbf{Y}|\mathbf{Z},\mathbf{X})\right] + \log p(\mathbf{A},\mathbf{c},\mathbf{V}) + \text{const} \\
&= \sum_{k=1}^{K}\sum_{n=1}^{N} r_{nk} \log \text{N}(\mathbf{y}_n|\mathbf{A}_k\mathbf{x}_n + \mathbf{c}_k, \mathbf{V}_k) \\
&\quad + \sum_{k=1}^{K} \log \text{MN}(\mathbf{A}_k|\mathbf{M}_0,\mathbf{K}_0,\mathbf{V}_k) \\
&\quad + \sum_{k=1}^{K} \log \text{N}(\mathbf{c}_k|\boldsymbol{\theta}_0,\rho_0\mathbf{V}_k) \\
&\quad + \sum_{k=1}^{K} \log \text{W}(\mathbf{V}_k|\boldsymbol{\Phi}_0,\eta_0) + \text{const},
\end{aligned}
$$

where $r_{nk} = \mathbb{E}_{q(\mathbf{Z})}\left[z_{nk}\right]$ are the expected responsibilities computed in the E-step. Further details on how to perform these updates via general exponential family recipes are provided in Appendix A.

B.3 M-Step of Hierarchical Infinite Linear Regression

$$
\begin{aligned}
\log q(\mathbf{t}) &= \mathbb{E}_{q(\mathbf{H})}\left[\log p(\mathbf{H}|\boldsymbol{\omega}(\mathbf{t}))\right] + \log p(\mathbf{t}) + \text{const} \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N} \hat{g}\log\left[t_m\prod_{l=1}^{m-1}(1-t_l)\right] \\
&\quad + \sum_{m=1}^{M}\log \text{Beta}(t_m|1,\beta_0) + \text{const}, \\
\log q(\mathbf{s}) &= \mathbb{E}_{q(\mathbf{H},\mathbf{Z})}\left[\log p(\mathbf{Z}|\mathbf{H})\right] + \log p(\mathbf{s}) + \text{const} \\
&= \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N} \hat{g}\,\hat{r}\log\left[s_{mk}\prod_{l=1}^{k-1}(1-s_{ml})\right] \\
&\quad + \sum_{m=1}^{M}\sum_{k=1}^{K}\log \text{Beta}(s_{mk}|1,\alpha_0) + \text{const}, \\
\log q(\boldsymbol{\mu}) &= \mathbb{E}_{q(\mathbf{H},\mathbf{Z},\boldsymbol{\tau},\boldsymbol{\Lambda})}\left[\log p(\mathbf{X}|\mathbf{H},\mathbf{Z})\right] + \mathbb{E}_{q(\boldsymbol{\tau},\boldsymbol{\Lambda})}\left[\log p(\boldsymbol{\mu}|\boldsymbol{\tau},\boldsymbol{\Lambda})\right] + \text{const} \\
&= \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N} \hat{g}\,\hat{r}\,\mathbb{E}_{q(\boldsymbol{\Lambda})}\left[\log \text{N}(\mathbf{x}_n|\boldsymbol{\mu}_{mk},\boldsymbol{\Lambda}_m)\right] \\
&\quad + \sum_{m=1}^{M}\sum_{k=1}^{K}\mathbb{E}_{q(\boldsymbol{\tau},\boldsymbol{\Lambda})}\left[\log \text{N}(\boldsymbol{\mu}_{mk}|\boldsymbol{\tau}_m,\kappa_0\boldsymbol{\Lambda}_m)\right] + \text{const}, \\
\log q(\boldsymbol{\tau},\boldsymbol{\Lambda}) &= \mathbb{E}_{q(\mathbf{H},\mathbf{Z},\boldsymbol{\mu})}\left[\log p(\mathbf{X}|\mathbf{H},\mathbf{Z})\right] + \log p(\boldsymbol{\tau},\boldsymbol{\Lambda}) + \mathbb{E}_{q(\boldsymbol{\mu})}\left[\log p(\boldsymbol{\mu}|\boldsymbol{\tau},\boldsymbol{\Lambda})\right] + \text{const} \\
&= \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N} \hat{g}\,\hat{r}\,\mathbb{E}_{q(\boldsymbol{\mu})}\left[\log \text{N}(\mathbf{x}_n|\boldsymbol{\mu}_{mk},\boldsymbol{\Lambda}_m)\right] \\
&\quad + \sum_{m=1}^{M}\log \text{N}(\boldsymbol{\tau}_m|\mathbf{m}_0,\lambda_0\boldsymbol{\Lambda}_m) \\
&\quad + \sum_{m=1}^{M}\log \text{W}(\boldsymbol{\Lambda}_m|\boldsymbol{\Psi}_0,\nu_0) \\
&\quad + \sum_{m=1}^{M}\sum_{k=1}^{K}\mathbb{E}_{q(\boldsymbol{\mu})}\left[\log \text{N}(\boldsymbol{\mu}_{mk}|\boldsymbol{\tau}_m,\kappa_0\boldsymbol{\Lambda}_m)\right] + \text{const},
\end{aligned}
$$

$$
\begin{aligned}
\log q(\mathbf{c}) &= \mathbb{E}_{q(\mathbf{H},\mathbf{Z},\mathbf{A},\mathbf{V})}\Big[\log p(\mathbf{Y}|\mathbf{H},\mathbf{Z},\mathbf{X})\Big] + \mathbb{E}_{q(\mathbf{V})}\Big[\log p(\mathbf{c}|\mathbf{V})\Big] + \text{const} \\
&= \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N} \hat{g}\,\hat{r}\,\mathbb{E}_{q(\mathbf{A},\mathbf{V})}\Big[\log \mathrm{N}(\mathbf{y}_n|\mathbf{A}_m\mathbf{x}_n + \mathbf{c}_{mk}, \mathbf{V}_m)\Big] \\
&\quad + \sum_{m=1}^{M}\sum_{k=1}^{K} \mathbb{E}_{q(\mathbf{V})}\Big[\mathrm{MN}(\mathbf{c}_{mk}|\boldsymbol{\theta}_0, \rho_0\mathbf{V}_m)\Big] + \text{const}, \\
\log q(\mathbf{A},\mathbf{V}) &= \mathbb{E}_{q(\mathbf{H},\mathbf{Z},\mathbf{c})}\Big[\log p(\mathbf{Y}|\mathbf{H},\mathbf{Z},\mathbf{X})\Big] + \log p(\mathbf{A},\mathbf{V}) \\
&\quad + \mathbb{E}_{q(\mathbf{c})}\Big[\log p(\mathbf{c}|\mathbf{V})\Big] + \text{const} \\
&= \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{n=1}^{N} \hat{g}\,\hat{r}\,\mathbb{E}_{q(\mathbf{c})}\Big[\log \mathrm{N}(\mathbf{y}_n|\mathbf{A}_m\mathbf{x}_n + \mathbf{c}_{mk}, \mathbf{V}_m)\Big] \\
&\quad + \sum_{m=1}^{M}\sum_{k=1}^{K} \mathbb{E}_{q(\mathbf{c})}\Big[\mathrm{MN}(\mathbf{c}_{mk}|\boldsymbol{\theta}_0, \rho_0\mathbf{V}_m)\Big] + \text{const},
\end{aligned}
$$

where the quantities $\hat{g} = g_{nm} = \mathbb{E}_{q(\mathbf{H})}\big[h_{nm}\big]$ and $\hat{r} = r_{nmk} = \mathbb{E}_{q(\mathbf{Z})}\big[z_{nmk}\big]$. After computing the necessary expectations, these computations largely correspond to the posterior update recipes described in Appendix A.

Appendix C
Reinforcement Learning For Switching Systems

C.1 Hybrid Relative Entropy Policy Search

$$
\begin{aligned}
\underset{\pi,\mu}{\text{maximize}} \quad & J = \sum_{\mathbf{z}} \iint r(\mathbf{x},\mathbf{u})\pi(\mathbf{u}|\mathbf{x},\mathbf{z})\mu(\mathbf{x},\mathbf{z})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}, \\
\text{subject to} \quad & \mu(\mathbf{x}',\mathbf{z}') = (1-\vartheta)\mu_1(\mathbf{x}',\mathbf{z}') \\
& \qquad + \vartheta \sum_{\mathbf{z}} \iint \pi(\mathbf{u}|\mathbf{x},\mathbf{z})\mu(\mathbf{x},\mathbf{z})p(\mathbf{x}',\mathbf{z}'|\mathbf{x},\mathbf{u},\mathbf{z})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}, \\
& \mathrm{KL}(\pi(\mathbf{u}|\mathbf{x},\mathbf{z})\mu(\mathbf{x},\mathbf{z})\,||\,q(\mathbf{x},\mathbf{u},\mathbf{z})) \le \epsilon \\
& \sum_{\mathbf{z}} \iint \pi(\mathbf{u}|\mathbf{x},\mathbf{z})\mu(\mathbf{x},\mathbf{z})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x} = 1,
\end{aligned}
$$

The Lagrangian function

$$
\begin{aligned}
L = & \sum_{\mathbf{z}} \iint r(\mathbf{x},\mathbf{u})p(\mathbf{x},\mathbf{u},\mathbf{z})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x} + \lambda\left(1 - \sum_{\mathbf{z}} \iint p(\mathbf{x},\mathbf{u},\mathbf{z})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}\right) \\
& + \vartheta \sum_{\mathbf{z}'} \int V(\mathbf{x}',\mathbf{z}') \sum_{\mathbf{z}} \iint p(\mathbf{x},\mathbf{u},\mathbf{z})p(\mathbf{x}',\mathbf{z}'|\mathbf{x},\mathbf{u},\mathbf{z})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{x}' \\
& + (1-\vartheta) \sum_{\mathbf{z}'} \iint V_1(\mathbf{x}',\mathbf{z}')p_1(\mathbf{x}',\mathbf{u}',\mathbf{z}')\,\mathrm{d}\mathbf{x}'\,\mathrm{d}\mathbf{u}' - \sum_{\mathbf{z}'} \iint V(\mathbf{x}',\mathbf{z}')p(\mathbf{x}',\mathbf{u}',\mathbf{z}')\,\mathrm{d}\mathbf{x}'\,\mathrm{d}\mathbf{u}' \\
& + \eta\left(\epsilon - \sum_{\mathbf{z}} \iint p(\mathbf{x},\mathbf{u},\mathbf{z}) \log \frac{p(\mathbf{x},\mathbf{u},\mathbf{z})}{q(\mathbf{x},\mathbf{u},\mathbf{z})}\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}\right),
\end{aligned}
$$

assuming $p(\mathbf{x},\mathbf{u},\mathbf{z}) = \mu(\mathbf{x},\mathbf{z})\pi(\mathbf{u}|\mathbf{x},\mathbf{z})$ and $\mu(\mathbf{x},\mathbf{z}) = \int p(\mathbf{x},\mathbf{u},\mathbf{z})\,\mathrm{d}\mathbf{u}$

Taking the partial derivative of L with respect to $p(\mathbf{x}, \mathbf{u}, \mathbf{z})$

$$\frac{\partial L}{\partial p} = r(\mathbf{x}, \mathbf{u}) - \lambda + (1 - \vartheta) \sum_{\mathbf{z}'} \int V(\mathbf{x}', \mathbf{z}') \mu_1(\mathbf{x}', \mathbf{z}') \, \mathrm{d}\mathbf{x}' + \vartheta \sum_{\mathbf{z}'} \int V(\mathbf{x}', \mathbf{z}') p(\mathbf{x}', \mathbf{z}' | \mathbf{x}, \mathbf{u}, \mathbf{z}) \, \mathrm{d}\mathbf{x}' - V(\mathbf{x}, \mathbf{z}) - \eta \log \frac{p^*(\mathbf{x}, \mathbf{u}, \mathbf{z})}{q(\mathbf{x}, \mathbf{u}, \mathbf{z})} - \eta$$

and set it to zero to get the solution

$$p^*(\mathbf{x}, \mathbf{u}, \mathbf{z}) = q(\mathbf{x}, \mathbf{u}, \mathbf{z}) \exp\left[\frac{A(\mathbf{x}, \mathbf{u}, \mathbf{z})}{\eta} - \frac{\lambda}{\eta} - 1 \right],$$

where

$$A(\mathbf{x}, \mathbf{u}, \mathbf{z}) = r(\mathbf{x}, \mathbf{u}) + (1 - \vartheta) \sum_{\mathbf{z}'} \int V(\mathbf{x}', \mathbf{z}') \mu_1(\mathbf{x}', \mathbf{z}') \, \mathrm{d}\mathbf{x}' + \vartheta \sum_{\mathbf{z}'} \int V(\mathbf{x}', \mathbf{z}') p(\mathbf{x}', \mathbf{z}' | \mathbf{x}, \mathbf{u}, \mathbf{z}) \, \mathrm{d}\mathbf{x}' - V(\mathbf{x}, \mathbf{z}).$$

By solving for the gradient of λ

$$\begin{aligned} 1 &= \sum_{\mathbf{z}} \iint p^*(\mathbf{x}, \mathbf{u}, \mathbf{z}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x} \\ 1 &= \sum_{\mathbf{z}} \iint q(\mathbf{x}, \mathbf{u}, \mathbf{z}) \exp\left[\frac{A(\mathbf{x}, \mathbf{u}, \mathbf{z})}{\eta} - \frac{\lambda^*}{\eta} - 1 \right] \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x} \\ \lambda^* &= -\eta + \eta \log \sum_{\mathbf{z}} \iint q(\mathbf{x}, \mathbf{u}, \mathbf{z}) \exp\left[\frac{A(\mathbf{x}, \mathbf{u}, \mathbf{z})}{\eta} \right] \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x}. \end{aligned}$$

Plugging λ^* back into the optimal distribution p^*, we retrieve the softmax form

$$p^*(\mathbf{x}, \mathbf{u}, \mathbf{z}) = \frac{q(\mathbf{x}, \mathbf{u}, \mathbf{z}) \exp\left[A(\mathbf{x}, \mathbf{u}, \mathbf{z}) / \eta \right]}{\sum_{\mathbf{z}} \iint q(\mathbf{x}, \mathbf{u}, \mathbf{z}) \exp\left[A(\mathbf{x}, \mathbf{u}, \mathbf{z}) / \eta \right] \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x}}.$$

Substituting p^* and λ^* into the Lagrangian L, we get the dual G

$$G = \eta \epsilon + \eta \log \sum_{\mathbf{z}} \iint q(\mathbf{x}, \mathbf{u}, \mathbf{z}) \exp\left[\frac{A(\mathbf{x}, \mathbf{u}, \mathbf{z})}{\eta} \right] \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x}.$$

Appendix D
Distributionally Robust Optimal Control

D.1 Worst-Case Parameter Optimization

$$
\begin{aligned}
\underset{p_t^{k+1}(\boldsymbol{\theta})}{\text{maximize}} \quad & \sum_{t=1}^{T-1} \int \int c_t(\mathbf{x},\mathbf{u})\mu_t(\mathbf{x})\pi_t^k(\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x} + \int c_T(\mathbf{x})\mu_T(\mathbf{x})\,\mathrm{d}\mathbf{x}, \\
\text{subject to} \quad & \iiint \mu_t(\mathbf{x})\pi_t^k(\mathbf{u}|\mathbf{x}) f(\mathbf{x}'|\mathbf{x},\mathbf{u},\boldsymbol{\theta}) p_t^{k+1}(\boldsymbol{\theta})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta} = \mu_{t+1}(\mathbf{x}'), \quad \forall \mathbf{x}', \forall t > 1, \\
& \sum_{t=1}^{T-1} \int p_t^{k+1}(\boldsymbol{\theta}) \log \frac{p_t^{k+1}(\boldsymbol{\theta})}{\hat{p}(\boldsymbol{\theta})}\,\mathrm{d}\boldsymbol{\theta} \leq \delta, \\
& \int p_t^{k+1}(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} = 1, \quad \forall t < T, \\
& \mu_1(\mathbf{x}) = \hat{\mu}_1(\mathbf{x}), \quad \forall \mathbf{x}, t = 1.
\end{aligned}
$$

The Lagrangian

$$
\begin{aligned}
H = & \sum_{t=1}^{T-1} \iint c_t(\mathbf{x},\mathbf{u})\mu_t(\mathbf{x})\pi_t^k(\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x} + \int c_T(\mathbf{x})\mu_T(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
& + \sum_{t=1}^{T-1} \gamma_t \left(\int_\theta p_t^{k+1}(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} - 1 \right) + \int V_1^{\boldsymbol{\theta}}(\mathbf{x})\hat{\mu}_1(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
& + \sum_{t=1}^{T-1} \int V_{t+1}^{\boldsymbol{\theta}}(\mathbf{x}') \iiint \mu_t(\mathbf{x})\pi_t^k(\mathbf{u}|\mathbf{x}) f(\mathbf{x}'|\mathbf{x},\mathbf{u},\boldsymbol{\theta}) p_t^{k+1}(\boldsymbol{\theta})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta}\,\mathrm{d}\mathbf{x}' \\
& - \sum_{t=1}^{T-1} \int V_t^{\boldsymbol{\theta}}(\mathbf{x}')\mu_t(\mathbf{x}')\,\mathrm{d}\mathbf{x}' + \int V_T^{\boldsymbol{\theta}}(\mathbf{x}')\mu_T(\mathbf{x}')\,\mathrm{d}\mathbf{x}' \\
& + \beta \left(\sum_{t=1}^{T-1} \int p_t^{k+1}(\boldsymbol{\theta}) \log \frac{p_t^{k+1}(\boldsymbol{\theta})}{\hat{p}(\boldsymbol{\theta})}\,\mathrm{d}\boldsymbol{\theta} - \delta \right).
\end{aligned}
$$

Take partial derivative of H with respect to $p_t^{k+1}(\boldsymbol{\theta})$

$$\frac{\partial H}{\partial p_t^{k+1}} = \gamma_t + \int V_{t+1}^{\boldsymbol{\theta}}(\mathbf{x}') \iint \mu_t(\mathbf{x}) \pi_t^k(\mathbf{u}|\mathbf{x}) f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{x}' + \beta \left(\log \frac{p_t^*(\boldsymbol{\theta})}{\hat{p}(\boldsymbol{\theta})} + 1 \right).$$

and set it to zero to get the optimal solution $p_t^{k+1}(\boldsymbol{\theta})$

$$\begin{aligned} p_t^{k+1}(\boldsymbol{\theta}) &= \exp\Bigg[-\frac{1}{\beta} \Bigg(\gamma_t + \beta - \beta \log \hat{p}(\boldsymbol{\theta}) \\ &\qquad + \int V_{t+1}^{\boldsymbol{\theta}}(\mathbf{x}') \iint \mu_t(\mathbf{x}) \pi_t^k(\mathbf{u}|\mathbf{x}) f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{x}' \Bigg) \Bigg] \\ &= \hat{p}(\boldsymbol{\theta}) \exp\left[-\frac{1}{\beta} \left(\gamma_t + \beta + Q_t(\boldsymbol{\theta}) \right) \right], \end{aligned}$$

where

$$Q_t(\boldsymbol{\theta}) = \int V_{t+1}^{\boldsymbol{\theta}}(\mathbf{x}') \iint \mu_t(\mathbf{x}) \pi_t^k(\mathbf{u}|\mathbf{x}) f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{x}'.$$

Plug $p_t^*(\boldsymbol{\theta})$ into Lagrangian H to get the dual F

$$\begin{aligned} F = &\sum_{t=1}^{T-1} \iint c_t(\mathbf{x}, \mathbf{u}) \mu_t(\mathbf{x}) \pi_t^k(\mathbf{u}|\mathbf{x}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x} + \int c_T(\mathbf{x}) \mu_T(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ &+ \int V_1^{\boldsymbol{\theta}}(\mathbf{x}) \hat{\mu}_1(\mathbf{x}) \, \mathrm{d}\mathbf{x} - \sum_{t=1}^{T-1} \int V_t^{\boldsymbol{\theta}}(\mathbf{x}') \mu_t(\mathbf{x}') \, \mathrm{d}\mathbf{x}' - \int V_T^{\boldsymbol{\theta}}(\mathbf{x}') \mu_T(\mathbf{x}') \, \mathrm{d}\mathbf{x}' \\ &- \sum_{t=1}^{T-1} \gamma_t - \beta \sum_{t=1}^{T-1} \int p_t^{k+1}(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} - \beta \delta \end{aligned}$$

Take partial derivative with respect to γ_t and set it to zero

$$\begin{aligned} 1 &= \int p_t^{k+1}(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \\ 1 &= \int \hat{p}(\boldsymbol{\theta}) \exp\left[-\frac{1}{\beta} \left(\gamma_t^* + \beta + Q_t(\boldsymbol{\theta}) \right) \right] \mathrm{d}\boldsymbol{\theta} \\ \gamma_t^* &= -\beta + \beta \log \int \hat{p}(\boldsymbol{\theta}) \exp\left[-\frac{1}{\beta} Q_t(\boldsymbol{\theta}) \right] \mathrm{d}\boldsymbol{\theta} \end{aligned}$$

Plug γ_t^* into $p_t^{k+1}(\boldsymbol{\theta})$ to get the normalized distribution

$$p_t^{k+1}(\boldsymbol{\theta}) = \frac{\hat{p}(\boldsymbol{\theta}) \exp\left[-\frac{1}{\beta} Q_t(\boldsymbol{\theta})\right]}{\int \hat{p}(\boldsymbol{\theta}) \exp\left[-\frac{1}{\beta} Q_t(\boldsymbol{\theta})\right] \mathrm{d}\boldsymbol{\theta}}$$

Plug γ_t^* and $p_t^{k+1}(\boldsymbol{\theta})$ back into F

$$\begin{aligned} F = & \sum_{t=1}^{T-1} \iint c_t(\mathbf{x}, \mathbf{u}) \mu_t(\mathbf{x}) \pi_t^k(\mathbf{u}|\mathbf{x}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x} + \int c_T(\mathbf{x}) \mu_T(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ & + \int V_1^{\boldsymbol{\theta}}(\mathbf{x}) \hat{\mu}_1(\mathbf{x}) \, \mathrm{d}\mathbf{x} - \sum_{t=1}^{T-1} \int V_t^{\boldsymbol{\theta}}(\mathbf{x}') \mu_t(\mathbf{x}') \, \mathrm{d}\mathbf{x}' - \int V_T^{\boldsymbol{\theta}}(\mathbf{x}') \mu_T(\mathbf{x}') \, \mathrm{d}\mathbf{x}' - \beta \delta \\ & - \beta \sum_{t=1}^{T-1} \log \int \hat{p}(\boldsymbol{\theta}) \exp\left[-\frac{1}{\beta} Q_t(\boldsymbol{\theta})\right] \mathrm{d}\boldsymbol{\theta} \end{aligned}$$

Take partial derivatives with respect to μ_T and μ_t

$$\begin{aligned} \frac{\partial F}{\partial \mu_t} &= -V_T^{\boldsymbol{\theta}}(\mathbf{x}) + c_T(\mathbf{x}) \\ \frac{\partial F}{\partial \mu_t} &= -V_t^{\boldsymbol{\theta}}(\mathbf{x}) + \int c_t(\mathbf{x}, \mathbf{u}) \pi_t^k(\mathbf{u}|\mathbf{x}) \, \mathrm{d}\mathbf{u} \\ &\quad + \int V_{t+1}^{\boldsymbol{\theta}}(\mathbf{x}') \iint p_t^{k+1}(\boldsymbol{\theta}) \pi_t^k(\mathbf{u}|\mathbf{x}) f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x}', \end{aligned}$$

and set it to zero to get the backward recursion for $V_t^{\boldsymbol{\theta}}(\mathbf{x})$.

Take partial derivatives with respect to $V_1^{\boldsymbol{\theta}}$ and $V_t^{\boldsymbol{\theta}}$

$$\begin{aligned} \frac{\partial F}{\partial V_1^{\boldsymbol{\theta}}} &= \hat{\mu}_1(\mathbf{x}) - \mu_1(\mathbf{x}) \\ \frac{\partial F}{\partial V_t^{\boldsymbol{\theta}}} &= -\mu_t(\mathbf{x}') + \iiint p_{t-1}^{k+1}(\boldsymbol{\theta}) \mu_{t-1}(\mathbf{x}) \pi_{t-1}^k(\mathbf{u}|\mathbf{x}) f(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \, \mathrm{d}\mathbf{u} \, \mathrm{d}\mathbf{x} \, \mathrm{d}\boldsymbol{\theta}, \end{aligned}$$

and set it to zero to get the forward recursion for $\mu_t(\mathbf{x})$. Finally, insert μ_t and $V_t^{\boldsymbol{\theta}}$ into F

$$\begin{aligned} F = &\sum_{t=1}^{T-1} \iint c_t(\mathbf{x},\mathbf{u})\mu_t(\mathbf{x})\pi_t^k(\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x} + \int V_1^{\boldsymbol{\theta}}(\mathbf{x})\hat{\mu}_1(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &- \sum_{t=1}^{T-1} \int V_t^{\boldsymbol{\theta}}(\mathbf{x})\mu_t(\mathbf{x})\,\mathrm{d}\mathbf{x} - \beta\delta - \beta \sum_{t=1}^{T-1} \log \int \hat{p}(\boldsymbol{\theta}) \exp\left[-\frac{1}{\beta} Q_t(\boldsymbol{\theta})\right] \mathrm{d}\boldsymbol{\theta} \\ &= \int V_1^{\boldsymbol{\theta}}(\mathbf{x})\hat{\mu}_1(\mathbf{x})\,\mathrm{d}\mathbf{x} + \beta \left(\sum_{t=1}^{T-1} \mathrm{KL}\left(p_t^{k+1}(\boldsymbol{\theta}) \,||\, \hat{p}(\boldsymbol{\theta})\right) - \delta \right). \end{aligned}$$

The dual is optimized with respect to β via gradient descent where

$$\frac{\partial F}{\partial \beta} = \sum_{t=1}^{T-1} \mathrm{KL}\left(p_t^{k+1}(\boldsymbol{\theta}) \,||\, \hat{p}(\boldsymbol{\theta})\right) - \delta.$$

D.2 Worst-Case Policy Optimization

$$
\begin{aligned}
\underset{\pi_t^{k+1}(\mathbf{u}|\mathbf{x})}{\text{minimize}} \quad & \sum_{t=1}^{T-1} \iint c_t(\mathbf{x},\mathbf{u})\mu_t(\mathbf{x})\pi_t^k(\mathbf{u}|\mathbf{x})\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x} + \int c_T(\mathbf{x})\mu_T(\mathbf{x})\,\mathrm{d}\mathbf{x}, \\
\text{subject to} \quad & \iiint \mu_t(\mathbf{x})\pi_t^{k+1}(\mathbf{u}|\mathbf{x}) f(\mathbf{x}'|\mathbf{x},\mathbf{u},\boldsymbol{\theta}) p_t^{k+1}(\boldsymbol{\theta})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta} = \mu_{t+1}(\mathbf{x}'), \quad \forall \mathbf{x}', \forall t \geq 1, \\
& \sum_{t=1}^{T-1} \int \mu_t(\mathbf{x}) \int \pi_t^{k+1}(\mathbf{u}|\mathbf{x}) \log \frac{\pi_t^{k+1}(\mathbf{u}|\mathbf{x})}{\pi_t^k(\mathbf{u}|\mathbf{x})}\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x} \leq \varepsilon, \\
& \int \pi_t^{k+1}(\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u} = 1, \quad \forall \mathbf{x}, \forall t < T, \\
& \mu_1(\mathbf{x}) = \hat{\mu}_1(\mathbf{x}), \quad \forall \mathbf{x}, t = 1.
\end{aligned}
$$

The Lagrangian function

$$
\begin{aligned}
L = & \sum_{t=1}^{T-1} \iint c_t(\mathbf{x},\mathbf{u})\mu_t(\mathbf{x})\pi_t^{k+1}(\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x} + \int c_T(\mathbf{x})\mu_T(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
& + \sum_{t=1}^{T-1} \int \lambda_t(\mathbf{x}) \left(\int \pi_t^{k+1}(\mathbf{u}|\mathbf{x})\,\mathrm{d}\mathbf{u} - 1 \right) \mathrm{d}\mathbf{x} + \int V_1^{\pi}(\mathbf{x})\hat{\mu}_1(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
& + \sum_{t=1}^{T-1} \int V_{t+1}^{\pi}(\mathbf{x}') \iiint \mu_t(\mathbf{x})\pi_t^{k+1}(\mathbf{u}|\mathbf{x}) f(\mathbf{x}'|\mathbf{x},\mathbf{u},\boldsymbol{\theta}) p_t^{k+1}(\boldsymbol{\theta})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta}\,\mathrm{d}\mathbf{x}' \\
& - \sum_{t=1}^{T-1} \int V_t^{\pi}(\mathbf{x}')\mu_t(\mathbf{x}')\,\mathrm{d}\mathbf{x}' - \int V_T^{\pi}(\mathbf{x}')\mu_T(\mathbf{x}')\,\mathrm{d}\mathbf{x}' \\
& + \alpha \left(\sum_{t=1}^{T-1} \int \mu_t(\mathbf{x}) \int \pi_t^{k+1}(\mathbf{u}|\mathbf{x}) \log \frac{\pi_t^{k+1}(\mathbf{u}|\mathbf{x})}{\pi_t^k(\mathbf{u}|\mathbf{x})}\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x} - \varepsilon \right)
\end{aligned}
$$

Take the partial derivative of L with respect to π_t^{k+1}

$$
\begin{aligned}
\frac{\partial L}{\partial \pi_t^{k+1}} = & c_t(\mathbf{x},\mathbf{u})\mu_t(\mathbf{x}) + \lambda_t(\mathbf{x})\alpha \left(\mu_t(\mathbf{x}) \log \frac{\pi_t^{k+1}(\mathbf{u}|\mathbf{x})}{\pi_t^k(\mathbf{u}|\mathbf{x})} + \mu_t(\mathbf{x}) \right) \\
& + \int V_{t+1}^{\pi}(\mathbf{x}')\mu_t(\mathbf{x}) \int f(\mathbf{x}'|\mathbf{x},\mathbf{u},\boldsymbol{\theta}) p_t^{k+1}(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}\,\mathrm{d}\mathbf{x}',
\end{aligned}
$$

and set it to zero to get the solution

$$\pi_t^{k+1}(\mathbf{u}|\mathbf{x}) = \pi_t^k(\mathbf{u}|\mathbf{x}) \exp\left[-\frac{1}{\alpha}\left(Q_t^\pi(\mathbf{x},\mathbf{u}) + \frac{\lambda_t(\mathbf{x})}{\mu_t(\mathbf{x})} + \alpha\right)\right],$$

where

$$Q_t^\pi(\mathbf{x},\mathbf{u}) = c_t(\mathbf{x},\mathbf{u}) + \int V_{t+1}^\pi(\mathbf{x}') \int f(\mathbf{x}'|\mathbf{x},\mathbf{u},\boldsymbol{\theta}) p_t^{k+1}(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}\, \mathrm{d}\mathbf{x}'.$$

Substitute π_t^{k+1} back into the Lagrangian L to get the dual G

$$\begin{aligned} G = & \int c_T(\mathbf{x})\mu_T(\mathbf{x})\, \mathrm{d}\mathbf{x} - \sum_{t=1}^{T-1} \int \lambda_t(\mathbf{x})\, \mathrm{d}\mathbf{x} \\ & + \int V_1^\pi(\mathbf{x})\hat{\mu}_1(\mathbf{x})\, \mathrm{d}\mathbf{x} - \sum_{t=1}^{T-1} \int V_t^\pi(\mathbf{x}')\mu_t(\mathbf{x}')\, \mathrm{d}\mathbf{x}' - \int V_T^\pi(\mathbf{x}')\mu_T(\mathbf{x}')\, \mathrm{d}\mathbf{x}' \\ & - \alpha \sum_{t=1}^{T-1} \iint \mu_t(\mathbf{x})\pi_t^{k+1}(\mathbf{u}|\mathbf{x})\, \mathrm{d}\mathbf{u}\, \mathrm{d}\mathbf{x} - \alpha\varepsilon. \end{aligned}$$

Take partial derivative with respect to λ_t and set it to zero

$$\begin{aligned} 1 &= \int \pi_t^{k+1}(\mathbf{u}|\mathbf{x})\, \mathrm{d}\mathbf{u} \\ 1 &= \int \pi_t^k(\mathbf{u}|\mathbf{x}) \exp\left[-\frac{1}{\alpha}\left(Q_t^\pi(\mathbf{x},\mathbf{u}) + \frac{\lambda_t^*(\mathbf{x})}{\mu_t(\mathbf{x})} + \alpha\right)\right] \mathrm{d}\mathbf{u} \\ \lambda_t^*(\mathbf{x}) &= -\alpha\mu_t(\mathbf{x}) + \alpha\mu_t(\mathbf{x}) \log \int \pi_t^k(\mathbf{u}|\mathbf{x}) \exp\left[-\frac{1}{\alpha} Q_t^\pi(\mathbf{x},\mathbf{u})\right] \mathrm{d}\mathbf{u}. \end{aligned}$$

Plug λ_t^* into π_t^{k+1} to get the normalized softmax distribution

$$\pi_t^{k+1}(\mathbf{u}|\mathbf{x}) = \frac{\pi_t^k(\mathbf{u}|\mathbf{x}) \exp\left[-\frac{1}{\alpha} Q_t^\pi(\mathbf{x},\mathbf{u})\right]}{\int \pi_t^k(\mathbf{u}|\mathbf{x}) \exp\left[-\frac{1}{\alpha} Q_t^\pi(\mathbf{x},\mathbf{u})\right] \mathrm{d}\mathbf{u}}.$$

Plug λ_t^* and π_t^{k+1} back into dual

$$\begin{aligned} G = &\int c_T(\mathbf{x})\mu_T(\mathbf{x})\,\mathrm{d}\mathbf{x} + \int V_1^\pi(\mathbf{x})\hat{\mu}_1(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &- \sum_{t=1}^{T-1}\int V_t^\pi(\mathbf{x}')\mu_t(\mathbf{x}')\,\mathrm{d}\mathbf{x}' - \int V_T^\pi(\mathbf{x}')\mu_T(\mathbf{x}')\,\mathrm{d}\mathbf{x}' - \alpha\varepsilon \\ &- \alpha\sum_{t=1}^{T-1}\pi_t^k(\mathbf{u}|\mathbf{x})\exp\left[-\frac{1}{\alpha}Q_t^\pi(\mathbf{x},\mathbf{u})\right]\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}. \end{aligned}$$

Take partial derivatives with respect to μ_T and μ_t

$$\begin{aligned} \frac{\partial G}{\partial \mu_t} &= -V_T^\pi(\mathbf{x}) + c_T(\mathbf{x}), \\ \frac{\partial G}{\partial \mu_t} &= -V_t^\pi(\mathbf{x}) - \alpha\log\int \pi_t^k(\mathbf{u}|\mathbf{x})\exp\left[-\frac{1}{\alpha}Q_t^\pi(\mathbf{x},\mathbf{u})\right]\mathrm{d}\mathbf{u}, \end{aligned}$$

and set it to zero to get a backward recursion for $V_t^\pi(\mathbf{x})$.

Take partial derivatives with respect to V_1^π and V_t^π

$$\begin{aligned} \frac{\partial G}{\partial V_1^\pi} &= \hat{\mu}_1(\mathbf{x}) - \mu_1(\mathbf{x}), \\ \frac{\partial G}{\partial V_t^\pi} &= -\mu_t(\mathbf{x}') + \iiint p_{t-1}^{k+1}(\boldsymbol{\theta})\mu_{t-1}(\mathbf{x})\pi_{t-1}^{k+1}(\mathbf{u}|\mathbf{x})f(\mathbf{x}'|\mathbf{x},\mathbf{u},\boldsymbol{\theta})\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta}, \end{aligned}$$

and set it to zero to get a forward recursion for $\mu_t(\mathbf{x})$.

Insert μ_t and V_t^π back into G

$$G = \int V_1^\pi(\mathbf{x})\hat{\mu}_1(\mathbf{x})\,\mathrm{d}\mathbf{x} - \alpha\varepsilon.$$

The dual is optimized with respect to α via gradient descent where

$$\frac{\partial G}{\partial \alpha} = \sum_{t=1}^{T-1}\int \mu_t(\mathbf{x})\,\mathrm{KL}\left(\pi_t^{k+1}(\mathbf{u}|\mathbf{x})\,||\,\pi_t^k(\mathbf{u}|\mathbf{x})\right)\mathrm{d}\mathbf{x} - \varepsilon.$$

Bibliography

Abdulsamad, H. and Peters, J. Hierarchical decomposition of nonlinear dynamics and control for system identification and policy distillation. In *Learning for Dynamics and Control*, 2020.

Abdulsamad, H., Arenz, O., Peters, J., and Neumann, G. State-regularized policy search for linearized dynamical systems. In *International Conference on Automated Planning and Scheduling*, 2017.

Abdulsamad, H., Nickl, P., Klink, P., and Peters, J. A variational infinite mixture for probabilistic inverse dynamics learning. *IEEE International Conference on Robotics and Automation*, 2021.

Abel, D., Hershkowitz, D. E., and Littman, M. L. Near optimal behavior via approximate state abstraction. *arXiv preprint arXiv:1701.04113*, 2017.

Ackerson, G. and Fu, K. On state estimation in switching environments. *IEEE Transactions on Automatic Control*, 1970.

Akametalu, A. K., Fisac, J. F., Gillula, J. H., Kaynama, S., Zeilinger, M. N., and Tomlin, C. J. Reachability-based safe learning with Gaussian processes. In *IEEE Conference on Decision and Control*, 2014.

Akrour, R., Neumann, G., Abdulsamad, H., and Abdolmaleki, A. Model-free trajectory optimization for reinforcement learning. In *International Conference on Machine Learning*, 2016.

Akrour, R., Veiga, F., Peters, J., and Neumann, G. Regularizing reinforcement learning with state abstraction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.

Amari, S.-I. Natural gradient works efficiently in learning. *Neural Computation*, 1998.

Anderson, B. D. and Moore, J. B. *Optimal Control: Linear Quadratic Methods*. 2007.

Andre, D. and Russell, S. J. State abstraction for programmable reinforcement learning agents. In *National Conference on Artificial Intelligence*, 2002.

Antoniak, C. E. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 1974.

Aoki, M. *Optimization of Stochastic Systems: Topics in Discrete-Time Systems*. 1967.

Arenz, O., Abdulsamad, H., and Neumann, G. Optimal control and inverse optimal control by distribution matching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

Arora, R., Basu, A., Mianjy, P., and Mukherjee, A. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*, 2016.

Arulkumaran, K., Deisenroth, M., Brundage, M., and Bharath, A. A. A brief survey of deep reinforcement learning. *arXiv preprint arXiv:1708.05866*, 2017.

Atkeson, C. G., Moore, A. W., and Schaal, S. Locally weighted learning for control. *Artificial Intelligence Review*, 1997a.

Atkeson, C. G., Moore, A. W., and Schaal, S. Locally weighted learning. *Artificial Intelligence Review*, 1997b.

Attias, H. A variational Bayesian framework for graphical models. In *Advances in Neural Information Processing Systems*, 2000.

Bacon, P.-L., Harb, J., and Precup, D. The option-critic architecture. In *AAAI Conference on Artificial Intelligence*, 2017.

Bako, L., Boukharouba, K., and Lecoeuche, S. An l_0-l_1 norm based optimization procedure for the identification of switched nonlinear systems. In *IEEE Conference on Decision and Control*, 2010.

Bar-Shalom, Y. and Li, X.-R. *Estimation and tracking- Principles, techniques, and software*. 1993.

Barber, D. Expectation correction for smoothed inference in switching linear dynamical systems. *Journal of Machine Learning Research*, 2006.

Barber, D. *Bayesian Reasoning and Machine Learning*. 2012.

Barto, A. G. and Mahadevan, S. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 2003.

Bauer, M., van der Wilk, M., and Rasmussen, C. E. Understanding probabilistic sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems*, 2016.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. A maximization technique occurring in

the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 1970.

Beal, M. J. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University College London, 2003.

Beal, M. J. and Ghahramani, Z. Variational Bayesian learning of directed graphical models with hidden variables. *Bayesian Analysis*, 2006.

Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. The infinite hidden Markov model. In *Advances in Neural Information Processing Systems*, 2002.

Beck, A. and Teboulle, M. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 2003.

Becker-Ehmck, P., Peters, J., and Van Der Smagt, P. Switching linear dynamics for variational Bayes filtering. In *International Conference on Machine Learning*, 2019.

Belousov, B. and Peters, J. f-Divergence constrained policy improvement. *arXiv preprint arXiv:1801.00056*, 2017.

Bemporad, A. and Di Cairano, S. Optimal control of discrete hybrid stochastic automata. In *International Workshop on Hybrid Systems: Computation and Control*, 2005.

Bemporad, A. and Morari, M. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 1999.

Bemporad, A., Borrelli, F., and Morari, M. Piecewise linear optimal controllers for hybrid systems. In *American Control Conference*, 2000.

Bemporad, A., Roll, J., and Ljung, L. Identification of hybrid systems via mixed-integer programming. In *IEEE Conference on Decision and Control*, 2001.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. *Robust Optimization*. 2009.

Bengio, Y. and Frasconi, P. An input-output HMM architecture. In *Advances in Neural Information Processing Systems*, 1995.

Bennett, C. L., Halpern, M., Hinshaw, G., Jarosik, N., Kogut, A., Limon, M., Meyer, S. S., Page, L., Spergel, D. N., Tucker, G. S., et al. First year Wilkinson microwave anisotropy probe (WMAP) observations: Preliminary maps and basic results. *The Astrophysical Journal Supplement Series*, 2003.

Bertsekas, D. P. and Shreve, S. *Stochastic Optimal Control: The Discrete-time Case*. 2004.

Bertsimas, D. and Sim, M. The price of robustness. *Operations Research*, 2004.

Bishop, C. M. *Pattern Recognition and Machine Learning*. 2006.

Bishop, C. M. and Svensén, M. Bayesian hierarchical mixtures of experts. In *Conference on Uncertainty in Artificial Intelligence*, 2003.

Blackwell, D., MacQueen, J. B., et al. Ferguson distributions via pólya urn schemes. *The Annals of Statistics*, 1973.

Blei, D. M. and Jordan, M. I. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 2006.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 2017.

Blom, H. A. and Bar-Shalom, Y. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE transactions on Automatic Control*, 1988.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In *International Conference on Machine Learning*, 2015.

Borrelli, F., Baotic, M., Bemporad, A., and Morari, M. An efficient algorithm for computing the state feedback optimal control law for discrete time hybrid systems. In *American Control Conference*, 2003.

Borrelli, F., Bemporad, A., Fodor, M., and Hrovat, D. An MPC/hybrid system approach to traction control. *IEEE Transactions on Control Systems Technology*, 2006.

Borrelli, F., Bemporad, A., and Morari, M. *Predictive Control for Linear and Hybrid Systems*. 2017.

Bottou, L. Online learning and stochastic approximations. *Online Learning in Neural Networks*, 1998.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. 2004.

Bradtke, S. J. and Duff, M. O. Reinforcement learning methods for continuous-time Markov decision problems. In *Advances in Neural Information Processing Systems*, 1995.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. *Handbook of Markov chain Monte Carlo*. 2011.

Büchler, D., Calandra, R., Schölkopf, B., and Peters, J. Control of musculoskeletal systems using learned dynamics models. *IEEE Robotics and Automation Letters*, 2018.

Burke, M., Hristov, Y., and Ramamoorthy, S. Hybrid system identification using switching density networks. In *Conference on Robot Learning*, 2020.

Calandra, R., Peters, J., Rasmussen, C. E., and Deisenroth, M. P. Manifold Gaussian processes for regression. In *International Joint Conference on Neural Networks*, 2016.

Calinon, S., D'halluin, F., Sauser, E. L., Caldwell, D. G., and Billard, A. G. Learning and reproduction of gestures by imitation. *IEEE Robotics & Automation Magazine*, 2010.

Camacho, E. F., Ramírez, D. R., Limón, D., De La Peña, D. M., and Alamo, T. Model predictive control techniques for hybrid systems. *Annual Reviews in Control*, 2010.

Cao, Y. and Fleet, D. J. Generalized product of experts for automatic and principled fusion of Gaussian process predictions. *arXiv preprint arXiv:1410.7827*, 2014.

Cassandras, C. G. and Lygeros, J. *Stochastic Hybrid Systems*. 2006.

Charalambous, C. D. and Rezaei, F. Stochastic uncertain systems subject to relative entropy constraints: Induced norms and monotonicity properties of minimax games. *IEEE Transactions on Automatic Control*, 2007.

Chu, C., Blanchet, J., and Glynn, P. Probability functional descent: A unifying perspective on GANs, variational inference, and reinforcement learning. In *International Conference on Machine Learning*, 2019.

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *arXiv preprint arXiv:1805.12114*, 2018.

Cleveland, W. S. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 1979.

Cleveland, W. S. and Loader, C. Smoothing by local regression: Principles and methods. In *Statistical Theory and Computational Aspects of Smoothing*, 1996.

Cohen, N., Sharir, O., and Shashua, A. On the expressive power of deep learning: A tensor analysis. In *Conference on Learning Theory*, 2016.

Coppens, P. and Patrinos, P. Data-driven distributionally robust MPC for constrained stochastic systems. *arXiv preprint arXiv:2103.03006*, 2021.

Coppens, P., Schuurmans, M., and Patrinos, P. Data-driven distributionally robust lqr with multiplicative noise. In *Learning for Dynamics and Control*, 2020.

Coulson, J., Lygeros, J., and Dörfler, F. Regularized and distributionally robust data-enabled predictive control. In *IEEE Conference on Decision and Control*, 2019.

Crouse, D. F., Willett, P., Pattipati, K., and Svensson, L. A look at Gaussian mixture reduction algorithms. In *International Conference on Information Fusion*, 2011.

Daniel, C., Van Hoof, H., Peters, J., and Neumann, G. Probabilistic inference for determining options in reinforcement learning. *Machine Learning*, 2016.

Davis, M. H. *Markov Models and Optimization*. 1993.

Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., and Hennig, P. Laplace redux-effortless Bayesian deep learning. *Advances in Neural Information Processing Systems*, 2021.

De Boer, P.-T., Kroese, D. P., Mannor, S., and Rubinstein, R. Y. A tutorial on the cross-entropy method. *Annals of Operations Research*, 2005.

Deisenroth, M. and Ng, J. W. Distributed Gaussian processes. In *International Conference on Machine Learning*, 2015.

Deisenroth, M. and Rasmussen, C. E. PILCO: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning*, 2011.

Deisenroth, M., Neumann, G., and Peters, J. A survey on policy search for robotics. *Foundations and Trends ® in Robotics*, 2013.

Delage, E. and Ye, Y. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 2010.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 1977.

Dietterich, T. G. State abstraction in MAXQ hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, 2000.

Duchi, J. and Namkoong, H. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018.

Eldan, R. and Shamir, O. The power of depth for feedforward neural networks. In *Conference on Learning Theory*, 2016.

Escobar, M. D. and West, M. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 1995.

Fan, J. and Gijbels, I. Variable bandwidth and local linear regression smoothers. *The Annals of Statistics*, 1992.

Fantoni, I. and Lozano, R. *Nonlinear Control for Underactuated Mechanical Systems*. 2002.

Farshidian, F. and Buchli, J. Risk sensitive, nonlinear optimal control: Iterative linear exponential-quadratic optimal control with Gaussian noise. *arXiv preprint arXiv:1512.07173*, 2015.

Ferguson, T. S. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1973.

Fleming, W. H. and Rishel, R. W. *Deterministic and Stochastic Optimal Control*. 2012.

Foong, A. Y., Burt, D. R., Li, Y., and Turner, R. E. On the expressiveness of approximate inference in Bayesian neural networks. *Advances in Neural Information Processing Systems*, 2020.

Forestier, J.-P. and Varaiya, P. Multilayer control of large Markov chains. *IEEE Transactions on Automatic Control*, 1978.

Foster, D. and Dayan, P. Structure in the space of value functions. *Machine Learning Jounral*, 2002.

Fox, E. *Bayesian nonparametric learning of complex dynamical phenomena*. PhD thesis, Massachusetts Institute of Technology, 2009.

Fox, E., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. Nonparametric Bayesian learning of switching linear dynamical systems. In *Advances in Neural Information Processing Systems*, 2009.

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. Bayesian nonparametric methods for learning Markov switching processes. *IEEE Signal Processing Magazine*, 2010.

Fragoso, M. Discrete-time jump LQG problem. *International Journal of Systems Science*, 1989.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

Gadd, C., Wade, S., and Boukouvalas, A. Enriched mixtures of generalised Gaussian process experts. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Garulli, A., Paoletti, S., and Vicino, A. A survey on switched and piecewise affine system identification. *International Federation of Automatic Control*, 2012.

Gelman, A. Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 2006.

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., and Rubin, D. B. *Bayesian Data Analysis*. 2013.

Geman, S. and Geman, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Pattern Analysis and Machine Intelligence*, 1984.

Ghahramani, Z. and Hinton, G. E. Variational learning for switching state-space models. *Neural Computation*, 2000.

Ghahramani, Z. and Jordan, M. I. Factorial hidden Markov models. *Machine Learning*, 1997.

Goebel, R., Sanfelice, R. G., and Teel, A. R. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. 2012.

Goldberg, P. W., Williams, C. K. I., and Bishop, C. M. Regression with input-dependent noise: A Gaussian process treatment. In *Advances in Neural Information Processing Systems*, 1997.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. 2016.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 2018.

Haddad, W. M., Chellaboina, V., and Nersesov, S. G. Impulsive and hybrid dynamical systems. *Princeton Series in Applied Mathematics*, 2006.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 2019.

Hamilton, J. D. Analysis of time series subject to changes in regime. *Journal of Econometrics*, 1990.

Hannah, L., Blei, D. M., and Powell, W. B. Dirichlet process mixtures of generalized linear models. *The Journal of Machine Learning Research*, 2011.

Hastie, T. and Loader, C. Local regression: Automatic kernel carpentry. *Statistical Science*, 1993.

Hastings, W. K. Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, 1970.

Hauskrecht, M., Meuleau, N., Kaelbling, L. P., Dean, T. L., and Boutilier, C. Hierarchical

solution of Markov decision processes using macro-actions. In *Conference on Uncertainty in Artificial Intelligence*, 1998.

Herbrich, R., Lawrence, N. D., and Seeger, M. Fast sparse Gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems*, 2003.

Hewing, L., Liniger, A., and Zeilinger, M. N. Cautious nmpc with Gaussian process dynamics for autonomous miniature race cars. In *IEEE European Control Conference*, 2018.

Hewing, L., Wabersich, K. P., Menner, M., and Zeilinger, M. N. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2020.

Hinton, G. and van Camp, D. Keeping neural networks simple by minimising the description length of weights. In *Conference on Learning Theory*, 1993.

Hjort, N. L., Holmes, C., Müller, P., and Walker, S. G. *Bayesian Nonparametrics*. 2010.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 1997.

Hoffman, M., Freitas, N., Doucet, A., and Peters, J. An expectation maximization algorithm for continuous markov decision processes with arbitrary reward. In *International Conference on Artificial Intelligence and*, 2009.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

Hu, Z. and Hong, L. J. Kullback-Leibler divergence constrained distributionally robust optimization. *Optimization Online*, 2013.

Huber, M. *A hybrid architecture for adaptive robot control*. PhD thesis, University of Massachusetts Amherst, 2000.

Huber, M. and Grupen, R. A. Learning to coordinate controllers-reinforcement learning on a control basis. In *International Joint Conferences on Artificial Intelligence*, 1997.

Huynh, V., Phung, D. Q., Venkatesh, S., Nguyen, X., Hoffman, M. D., and Bui, H. H. Scalable nonparametric bayesian multilevel clustering. In *Conference on Uncertainty in Artificial Intelligence*, 2016.

Ishwaran, H. and James, L. F. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 2001.

Iverson, K. E. A programming language. In *Joint Computer Conference*, 1962.

Iwata, T., Duvenaud, D., and Ghahramani, Z. Warped mixtures for nonparametric cluster shapes. *Conference on Uncertainty in Artificial Intelligence*, 2013.

Jaakkola, T. S. Tutorial on variational approximation methods. In *Advanced Mean Field Methods: Theory and Practice*, 2000.

Jaakkola, T. S. and Jordan, M. I. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 2000.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. Adaptive mixtures of local experts. *Neural Computation*, 1991.

Jacobson, D. H. and Mayne, D. Q. *Differential Dynamic Programming*. 1970.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Jensen, J. L. W. V. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 1906.

Jordan, M. I. and Jacobs, R. A. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 1994.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine Learning*, 1999.

Juditsky, A., Nemirovski, A., and Tauvel, C. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 2011.

Juloski, A. L., Weiland, S., and Heemels, W. A Bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, 2005.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, 2002.

Kakade, S. M. A natural policy gradient. *Advances in Neural Information Processing Systems*, 2001.

Kamthe, S. and Deisenroth, M. Data-efficient reinforcement learning with probabilistic model predictive control. In *International Conference on Artificial Intelligence and Statistics*, 2018.

Kersting, K., Plagemann, C., Pfaff, P., and Burgard, W. Most likely heteroscedastic Gaussian process regression. In *International Conference on Machine Learning*, 2007.

Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. Fast and scalable Bayesian deep learning by weight-perturbation in adam. *International Conference on Machine Learning*, 2018.

Kim, C.-J. Dynamic linear models with markov-switching. *Journal of Econometrics*, 1994.

Kim, K. and Yang, I. Minimax control of ambiguous linear stochastic systems using the Wasserstein metric. In *IEEE Conference on Decision and Control*, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kober, J., Bagnell, J. A., and Peters, J. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 2013.

Koller, D., Friedman, N., and Bach, F. *Probabilistic Graphical Models: Principles and Techniques*. 2009.

Koller, T., Berkenkamp, F., Turchetta, M., and Krause, A. Learning-based model predictive control for safe exploration. In *IEEE Conference on Decision and Control*, 2018.

Konidaris, G. and Barto, A. G. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in Neural Information Processing Systems*, 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

Kuhn, D., Esfahani, P. M., Nguyen, V. A., and Shafieezadeh-Abadeh, S. Wasserstein distributionally robust optimization: Theory and applications in machine learning. In *Operations Research & Management Science in the Age of Analytics*. 2019.

Kullback, S. and Leibler, R. A. On information and sufficiency. *The Annals of Mathematical Statistics*, 1951.

Kurihara, K., Welling, M., and Vlassis, N. Accelerated variational Dirichlet process mixtures. In *Advances in Neural Information Processing Systems*, 2006.

Kurihara, K., Welling, M., and Teh, Y. W. Collapsed variational Dirichlet process mixture models. In *International Joint Conferences on Artificial Intelligence*, 2007.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive un-

certainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

Lauer, F., Bloch, G., and Vidal, R. Nonlinear hybrid system identification with kernel models. In *IEEE Conference on Decision and Control*, 2010.

Le, Q. V., Smola, A. J., and Canu, S. Heteroscedastic Gaussian process regression. In *International Conference on Machine Learning*, 2005.

Lerner, U. N. *Hybrid Bayesian networks for reasoning about complex systems*. PhD thesis, Stanford University, 2002.

Levine, S. and Abbeel, P. Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems*, 2014.

Levine, S. and Koltun, V. Guided policy search. In *International Conference on Machine Learning*, 2013.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 2016.

Li, L., Walsh, T. J., and Littman, M. L. Towards a unified theory of state abstraction for MDPs. In *International Symposium on Artificial Intelligence and Mathematics*, 2006.

Liberzon, D. *Switching in Systems and Control*. 2003.

Liberzon, D. *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. 2011.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Lin, H. W., Tegmark, M., and Rolnick, D. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 2017.

Lin, S.-H. *Exploiting structure for planning and control*. PhD thesis, Brown University, 1997.

Linderman, S. W., Johnson, M. J., Miller, A. C., Adams, R. P., Blei, D. M., and Paninski, L. Bayesian learning and inference in recurrent switching linear dynamical systems. In *International Conference on Artificial Intelligence and Statistics*, 2017.

Liu, H., Ong, Y.-S., and Cai, J. Large-scale heteroscedastic regression via Gaussian process. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Lu, S. and You, F. Multistage model predictive control based on data-driven distributionally robust optimization. In *IEEE American Control Conference*, 2020.

Mankowitz, D. J., Mann, T. A., and Mannor, S. Adaptive skills adaptive partitions (ASAP). In *Advances in Neural Information Processing Systems*, 2016.

Marcucci, T. and Tedrake, R. Mixed-integer formulations for optimal control of piecewise-affine systems. In *ACM International Conference on Hybrid Systems: Computation and Control*, 2019.

Maritz, J. S. and Lwin, T. Empirical Bayes methods. *Monographs on Statistics and Applied Probability*, 1989.

Matthews, A. G. d. G. *Scalable Gaussian Process Inference Using Variational Methods*. PhD thesis, University of Cambridge, 2017.

Mayne, D. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *International Journal of Control*, 1966.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*. 1989.

McCullagh, P. and Nelder, J. A. *Generalized Linear Models*. 1989.

McLachlan, G. J. and Basford, K. E. *Mixture Models: Inference and Applications to Clustering*. 1988.

Meeds, E. and Osindero, S. An alternative infinite mixture of Gaussian process experts. In *Advances in Neural Information Processing Systems*, 2006.

Meier, F., Hennig, P., and Schaal, S. Incremental local Gaussian regression. In *Advances in Neural Information Processing Systems*, 2014.

Menchinelli, P. and Bemporad, A. Hybrid model predictive control of a solar air conditioning plant. *European Journal of Control*, 2008.

Mertikopoulos, P., Lecouat, B., Zenati, H., Foo, C., Chandrasekhar, V., and Piliouras, G. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *International Conference on Learning Representations*, 2019.

Mesot, B. and Barber, D. Switching linear dynamical systems for noise robust speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 2007.

Milz, J. and Ulbrich, M. An approximation scheme for distributionally robust nonlinear optimization. *SIAM Journal on Optimization*, 2020.

Minka, T. Bayesian linear regression, 2000a.

Minka, T. Estimating a Dirichlet distribution, 2000b.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

Moerland, P. Classification using localized mixtures of experts. 1999.

Montufar, G. F., Pascanu, R., Cho, K., and Bengio, Y. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, 2014.

Moody, J. and Darken, C. J. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1989.

Morari, M., Baotic, M., and Borrelli, F. Hybrid systems modeling and control. *European Journal of Control*, 2003.

Mueller, P., Erkanli, A., and West, M. Bayesian curve fitting using multivariate normal mixtures. *Biometrika*, 1996.

Murphy, K. P. Switching Kalman filters, 1998.

Murphy, K. P. Bayesian linear regression, 2007a.

Murphy, K. P. Conjugate Bayesian analysis of the Gaussian distribution, 2007b.

Murphy, K. P. *Machine Learning: A Probabilistic Perspective*. 2012.

Nakka, Y. K., Liu, A., Shi, G., Anandkumar, A., Yue, Y., and Chung, S.-J. Chance-constrained trajectory optimization for safe exploration and learning of nonlinear systems. *IEEE Robotics and Automation Letters*, 2020.

Nass, D., Belousov, B., and Peters, J. Entropic risk measure in policy search. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

Neal, R. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1994.

Neal, R. M. *Bayesian learning for neural networks*. 1996.

Neal, R. M. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 2000.

Nelles, O. and Isermann, R. Basis function networks for interpolation of local linear models. *Conference on Decision and Control*, 1996.

Nemirovski, A. Prox-method with rate of convergence o (1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 2004.

Nguyen, T. V., Phung, D., Nguyen, X., Venkatesh, S., and Bui, H. Bayesian nonparametric multilevel clustering with group-level contexts. In *International Conference on Machine Learning*, 2014.

Nguyen-Tuong, D. and Peters, J. Using model knowledge for learning inverse dynamics. In *IEEE International Conference on Robotics and Automation*, 2010.

Nguyen-Tuong, D., Peters, J., and Seeger, M. Local Gaussian process regression for real time online model learning and control. In *Advances in Neural Information Processing Systems*, 2008.

Nguyen-Tuong, D., Seeger, M. W., and Peters, J. Model learning with local Gaussian process regression. *Advanced Robotics*, 2009.

Nishimura, H., Mehr, N., Gaidon, A., and Schwager, M. RAT-iLQR: A risk auto-tuning controller to optimally account for stochastic model mismatch. *IEEE Robotics and Automation Letters*, 2021.

Nocedal, J. and Wright, S. *Numerical Optimization*. 2006.

Oh, S. M., Rehg, J. M., Balch, T., and Dellaert, F. Data-driven MCMC for learning and inference in switching linear dynamic systems. In *National Conference on Artificial Intelligence*, 2005.

Opper, M. and Saad, D. *Advanced Mean Field Methods: Theory and Practice*. 2001.

Pajarinen, J., Kyrki, V., Koval, M., Srinivasa, S., Peters, J., and Neumann, G. Hybrid control trajectory optimization under uncertainty. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

Pan, X. and Srikumar, V. Expressiveness of rectifier networks. In *International Conference on Machine Learning*, 2016.

Paoletti, S., Juloski, A. L., Ferrari-Trecate, G., and Vidal, R. Identification of hybrid systems: A tutorial. *European Journal of Control*, 2007.

Parr, R. E. *Hierarchical control and learning for Markov decision processes*. PhD thesis, University of California Berkeley, 1998.

Pavlovic, V., Rehg, J. M., and MacCormick, J. Learning switching linear models of human motion. In *Advances in Neural Information Processing Systems*, 2001.

Peters, J., Mülling, K., and Altun, Y. Relative entropy policy search. In *AAAI Conference on Artificial Intelligence*, 2010.

Petersen, I. R., James, M. R., and Dupuis, P. Minimax optimal control of stochastic uncertain systems with relative entropy constraints. *IEEE Transactions on Automatic Control*, 2000.

Petersen, K. B. and Pedersen, M. S. The matrix cookbook, 2012.

Petersen, P. and Voigtlaender, F. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks*, 2018.

Pignat, E. and Calinon, S. Bayesian Gaussian mixture model for robotic policy imitation. *IEEE Robotics and Automation Letters*, 2019.

Pignat, E., Lembono, T. S., and Calinon, S. Variational inference with mixture model approximation for applications in robotics, 2019.

Pirotta, M., Restelli, M., Pecorino, A., and Calandriello, D. Safe policy iteration. In *International Conference on Machine Learning*, 2013.

Precup, D. *Temporal abstraction in reinforcement learning*. PhD thesis, University of Massachusetts Amherst, 2000.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. 2014.

Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. *Ieee*, 1989.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2008.

Rahimian, H. and Mehrotra, S. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

Rasmussen, C. E. The infinite Gaussian mixture model. In *Advances in Neural Information Processing Systems*, 1999.

Rasmussen, C. E. and Ghahramani, Z. Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems*, 2002.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. 2006.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, 1951.

Robert, C. P., Casella, G., and Casella, G. *Monte Carlo Statistical Methods*. 1999.

Rodriguez, A., Dunson, D. B., and Gelfand, A. E. Bayesian nonparametric functional data analysis through density estimation. *Biometrika*, 2009.

Roll, J., Bemporad, A., and Ljung, L. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 2004.

Ross, S. M. *Introduction to Probability Models*. 2014.

Rousseau, J. and Mengersen, K. Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society*, 2011.

Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., and Hadsell, R. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

Salimbeni, H. and Deisenroth, M. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, 2017.

Samuelson, S. and Yang, I. Data-driven distributionally robust control of energy storage to manage wind power fluctuations. In *IEEE Conference on Control Technology and Applications*, 2017.

Särkkä, S. *Bayesian filtering and smoothing*. 2013.

Sato, M.-a. and Ishii, S. Online EM algorithm for the normalized Gaussian network. *Neural Computation*, 2000.

Scarf, H. A min-max solution of an inventory problem. *Studies in The Mathematical Theory of Inventory and Production*, 1958.

Schaal, S. and Atkeson, C. G. Constructive incremental learning from only local information. *Neural Computation*, 1998.

Schaal, S., Atkeson, C. G., and Vijayakumar, S. Scalable techniques from nonparametric statistics for real time robot learning. *Applied Intelligence*, 2002.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Schultheis, M., Belousov, B., Abdulsamad, H., and Peters, J. Receding horizon curiosity. In *Conference on Robot Learning*, 2020.

Serra, T., Tjandraatmadja, C., and Ramalingam, S. Bounding and counting linear regions of deep neural networks. *arXiv preprint arXiv:1711.02114*, 2017.

Sethuraman, J. A constructive definition of Dirichlet priors. *Statistica Sinica*, 1994.

Shahbaba, B. and Neal, R. M. Nonlinear models using Dirichlet process mixtures. *The Journal of Machine Learning Research*, 2009.

Smith, A. Bayesian detection and estimation of jumps in linear systems. *Bayesian Statistics*, 1985.

Smith, M., Hoof, H., and Pineau, J. An inference-based policy gradient method for learning options. In *International Conference on Machine Learning*, 2018.

Solin, A. Cubature integration methods in nonlinear Kalman filtering and smoothing. 2010.

Sontag, E. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 1981.

Sosic, A., Zoubir, A. M., and Koeppl, H. A Bayesian approach to policy recognition and state representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Sridharan, K. and Tewari, A. Convex games in Banach spaces. In *Conference on Learning Theory*, 2010.

Sudderth, E. B. *Graphical Models for Visual Object Recognition and Tracking*. PhD thesis, Massachusetts Institute of Technology, 2006.

Sun, S., Zhang, G., Shi, J., and Grosse, R. Functional variational Bayesian neuralnetworks. In *International Conference on Learning Representations*, 2019a.

Sun, S., Zhang, G., Shi, J., and Grosse, R. Functional variational Bayesian neural networks. *International Conference on Learning Representations*, 2019b.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. 2018.

Sutton, R. S., Precup, D., and Singh, S. Intra-option learning about temporally abstract actions. In *International Conference on Machine Learning*, 1998.

Sutton, R. S., Precup, D., and Singh, S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 1999.

Tassa, Y., Erez, T., and Todorov, E. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

Teh, Y. W. Dirichlet process. In *Encyclopedia of Machine Learning and Data Mining*. 2010.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. Sharing clusters among related groups: Hierarchical Dirichlet processes. In *Advances in Neural Information Processing Systems*, 2005.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 2006.

Ting, J. A., Kalakrishnan, M., Vijayakumar, S., and Schaal, S. Bayesian kernel shaping for learning control. In *Advances in Neural Information Processing Systems*, 2009.

Ting, J. A., Meier, F., Vijayakumar, S., and Schaal, S. Locally weighted regression for control. In *Encyclopedia of Machine Learning and Data Mining*. 2010.

Titsias, M. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, 2009.

Todorov, E. Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system. *Neural Computation*, 2005.

Todorov, E. and Li, W. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *IEEE American Control Conference*, 2005.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

Toussaint, M. Robot trajectory optimization using approximate inference. In *International Conference on Machine Learning*, 2009.

Toussaint, M. and Storkey, A. Probabilistic inference for solving discrete and continuous state markov decision processes. In *International Conference on Machine learning*, 2006.

Van Hoof, H., Peters, J., and Neumann, G. Learning of non-parametric control policies with high-dimensional state features. In *International Conference on Artificial Intelligence and Statistics*, 2015.

Van Parys, B. P., Kuhn, D., Goulart, P. J., and Morari, M. Distributionally robust control of constrained stochastic systems. *IEEE Transactions on Automatic Control*, 2015.

Vidal, R., Soatto, S., Ma, Y., and Sastry, S. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *IEEE International Conference on Decision and Control*, 2003.

Vijayakumar, S. and Schaal, S. Locally weighted projection regression: Incremental real

time learning in high dimensional space. In *International Conference on Machine Learning*, 2000.

Vijayakumar, S., D'Souza, A., and Schaal, S. Incremental online learning in high dimensions. *Neural Computation*, 2005.

Von Rosen, D. Moments for matrix normal variables. *Statistics: A Journal of Theoretical and Applied Statistics*, 1988.

Wainwright, M. J. and Jordan, M. I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 2008.

Wan, E. A., Van Der Merwe, R., and Haykin, S. The unscented Kalman filter. *Kalman Filtering and Neural Networks*, 2001.

Wasserman, L. *All of Nonparametric Statistics*. 2006.

Watson, J. and Peters, J. Advancing trajectory optimization with approximate inference: Exploration, covariance control and adaptive risk. *arXiv preprint arXiv:2103.06319*, 2021.

Watson, J., Abdulsamad, H., and Peters, J. Stochastic optimal control as approximate input inference. In *Conference on Robot Learning*, 2020a.

Watson, J., Lin, J. A., Klink, P., and Peters, J. Neural linear models with functional gaussian process priors. In *Symposium on Advances in Approximate Bayesian Inference*, 2020b.

Watson, J., Lin, J. A., Klink, P., Pajarinen, J., and Peters, J. Latent derivative Bayesian last layer networks. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Wenzel, F., Galy-Fajou, T., Donner, C., Kloft, M., and Opper, M. Efficient gaussian process classification using pòlya-gamma data augmentation. In *AAAI Conference on Artificial Intelligence*, 2019.

Wenzel, F., Roth, K., Veeling, B., Swiatkowski, J., Tran, L., Mandt, S., Snoek, J., Salimans, T., Jenatton, R., and Nowozin, S. How good is the Bayes posterior in deep neural networks really? *International Conference on Machine Learning*, 2020.

West, M., M, P., and Escobar, M. Hierarchical priors and mixture models, with application in regression and density estimation. In *Aspects of Uncertainty: A Tribute to DV Lindley*, 1994.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. Deep kernel learning. In *International Conference on Artificial Intelligence and Statistics*, 2016.

Xu, L., Jordan, M., and Hinton, G. E. An alternative model for mixtures of experts. *Advances in Neural Information Processing Systems*, 1994.

Xu, T., Liang, Y., and Lan, G. A primal approach to constrained policy optimization: Global optimality and finite-time analysis. *arXiv preprint arXiv:2011.05869*, 2020.

Yang, I. Wasserstein distributionally robust stochastic control: A data-driven approach. *IEEE Transactions on Automatic Control*, 2020.

Yerebakan, H. Z., Rajwa, B., and Dundar, M. The infinite mixture of infinite Gaussian mixtures. In *Advances in Neural Information Processing Systems*, 2014.

Yu, X. Gibbs sampling methods for Dirichlet process mixture model: Technical details, 2009.

Yuan, C. and Neubauer, C. Variational mixture of Gaussian process experts. In *Advances in Neural Information Processing Systems*, 2009.

Yuksel, S. E., Wilson, J. N., and Gader, P. D. Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 2012.

Zhou, K., Doyle, J. C., Glover, K., et al. *Robust and Optimal Control*. 1996.

Zhu, F. and Antsaklis, P. J. Optimal control of hybrid switched systems: A brief survey. *Discrete Event Dynamic Systems*, 2015.

Zhu, J.-J., Jitkrittum, W., Diehl, M., and Schölkopf, B. Worst-case risk quantification under distributional ambiguity using kernel mean embedding in moment problem. In *IEEE Conference on Decision and Control*, 2020.

List of Acronyms

ANN artificial neural network.

ARHMM autoregressive hidden Markov model.

BNN Bayesian neural network.

BNP Bayesian nonparametrics.

Cat categorical.

CEM cross-entropy method.

CMB cosmic microwave background.

DDP differential dynamic programming.

Dir Dirichlet.

DP Dirichlet process.

DRO distributionally robust optimization.

E-step expectation step.

EB-step empirical Bayes step.

ELBO evidence lower bound.

EM expectation-maximization.

FHMM factorial hidden Markov model.

FNN feed-forward neural net.

GMM Gaussian mixture model.

GP Gaussian process.

GPB generalized pseudo Bayesian approximation.

gPoE generalized product of experts.

GPR Gaussian process regression.

GPS guided policy search.

Hb-REPS hybrid relative entropy policy search.

HDBN hybrid dynamic Bayesian networks.

HILR hierarchical infinite local regression.

HMM hidden Markov model.

HRL hierarchical reinforcement learning.

HSMM semi-hidden Markov model.

iLEQG iterative linear-exponential-quadratic Gaussian.

iLQR iterative linear-quadratic regulator.

ILR infinite local regression.

IMM interacting multiple model.

KL Kullback-Leibler divergence.

LGR local Gaussian regression.

LR local regression.

LSTM long-short-term memory network.

LWPR locally weighted projection regression.

M-step maximization step.

MAP maximum a posteriori.

MCMC Markov chain Monte Carlo.

MDP Markov decision processes.

MIQP mixed-integer quadratic program.

MJLS Markov jump linear systems.

MN matrix-normal.

MNW matrix-normal-Wishart.

MoE mixture of experts.

MPC model predictive control.

MSE mean squared error.

NMSE normalized mean squared error.

NW normal-Wishart.

PGM probabilistic graphical models.

PoE product of experts.

PWA piecewise affine.

PWARX piecewise autoregressive exogenous systems.

PWL piecewise linear.

rARHMM recurrent autoregressive hidden Markov model.

rBCM robust Bayesian committee machine.

ReLU rectified linear units.

REPS relative entropy policy search.

RFF random Fourier feature.

RFWR receptive field weighted regression.

RL reinforcement learning.

RNN recurrent neural network.

rSLDS recurrent switching linear dynamical systems.

SAC soft actor-critic.

SDS switching dynamical systems.

SGPR sparse Gaussian process regression.

SLDS switching linear dynamical systems.

SMDP semi-Markov decision processes.

SSM switching state-space models.

SVI stochastic variational inference.

TD temporal difference.

VB variational Bayes.

VBEM variational Bayes expectation-maximization.

VI variational inference.

List of Figures

Publications

Journal Articles

Akrour, R., Abdolmaleki, A., Abdulsamad, H., and Neumann, G. Model-free trajectory optimization with monotonic improvement. *Journal for Machine Learning Research*, 2018.

Klink, P., Abdulsamad, H., Belousov, B., D'Eramo, C., Peters, J., and Pajarinen, J. A probabilistic interpretation of self-paced learning with applications to reinforcement learning. *Journal for Machine Learning Research*, 2021.

Watson, J., Abdulsamad, H., and Peters, J. Stochastic control through approximate Bayesian input inference. *IEEE Transactions on Automatic Control*, 2021, Submitted.

Abdulsamad, H. and Peters, J. Model-based reinforcement learning for stochastic hybrid systems. *IEEE Transactions on Automatic Control*, 2021, Submitted.

Abdulsamad, H., Nickl, P., Klink, P., and Peters, J. Variational structured mixtures for learning probabilistic inverse dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, Submitted.

Abdulsamad, H., Dorau, T., Belousov, B., Zhu, J-J., and Peters, J. Distributionally robust trajectory optimization under uncertain dynamics via relative entropy trust regions. *IEEE Transactions on Automatic Control*, 2021, Submitted.

Conferences and Workshops

Parisi, S., Abdulsamad, H., Paraschos, A., Daniel, C., and Peters, J. Reinforcement learning vs human programming in Tetherball robot games. *IEEE/RSJ Conference on Intelligent Robots and Systems*, 2015.

Akrour, R., Abdolmaleki, A., Abdulsamad, H., and Neumann, G. Model-free trajectory optimization for reinforcement learning. *International Conference on Machine Learning*, 2016.

Arenz, O., Abdulsamad, H., and Neumann, G. Optimal control and inverse optimal control by distribution matching. *IEEE/RSJ Conference on Intelligent Robots and Systems*, 2016.

Abdulsamad, H., Arenz, O., Peters, J., and Neumann, G. State-regularized policy search for linearized dynamical systems. *International Conference on Automated Planning and Scheduling*, 2017.

Celik, O., Abdulsamad, H., and Peters, J. Chance-constrained trajectory optimization for nonlinear dynamical system with unknown stochastic dynamics. *IEEE/RSJ Conference on Intelligent Robots and Systems*, 2019.

Abdulsamad, H., Naveh, K., and Peters, J. Model-based relative entropy policy search for stochastic hybrid systems. *Multidisciplinary Conference on Reinforcement Learning and Decision Making*, 2019.

Belousov, B., Abdulsamad, H., Schultheis, M., and Peters, J. Belief space model predictive control for approximately optimal system identification. *Multidisciplinary Conference on Reinforcement Learning and Decision Making*, 2019.

Schultheis, M., Belousov, B., Abdulsamad, H., and Peters, J. Receding horizon curiosity. *Conference on Robot Learning*, 2019.

Klink, P., Abdulsamad, H., Belousov, B., and Peters, J. Self-paced contextual reinforcement learning. *Conference on Robot Learning*, 2019.

Watson, J., Abdulsamad, H., and Peters, J. Stochastic optimal control as approximate input inference. *Conference on Robot Learning*, 2019.

Tosatto, S., Carvalho, J., Abdulsamad, H., and Peters, J. A nonparametric off-policy policy gradient. *International Conference on Artificial Intelligence and Statistics*, 2020.

Abdulsamad, H. and Peters, J. Hierarchical decomposition of nonlinear dynamics and control for system identification and policy distillation. *Conference on Learning for Dynamics and Control*, 2020.

Abdulsamad, H. and Peters, J. Learning hybrid dynamics and control. *ECML/PKDD Workshop on Deep Continuous-Discrete Machine Learning*, 2020.

Abdulsamad, H., Nickl, P., Klink, P., and Peters, J. A variational infinite mixture for probabilistic inverse dynamics learning. *IEEE International Conference on Robotics and Automation*, 2021.

Books

Belousov, B., Abdulsamad, H., Klink, P., Parisi, S., and Peters, J. (Eds.). Reinforcement learning algorithms: Analysis and applications. *Studies in Computational Intelligence, Springer International Publishing*, 2021.

Curriculum Vitae

Hany Abdulsamad
hany@robot-learning.de
abdulsamad.ias.tu-darmstadt.de
hanyas.github.com

Education

since 04/2016 **Ph.D. Student.** Machine learning and robotics at the Intelligent Autonomous Systems Lab, Technische Universität Darmstadt.

2012–2016 **Master of Science.** Electrical Engineering and Information Technology, Technische Universität Darmstadt

2008–2012 **Bachelor of Science.** Electrical Engineering and Information Technology, Technische Universität Darmstadt.

Research Interest

Machine Learning. Focus on hierarchical representation for modeling and control of dynamical systems, including inference of sequence models and Bayesian nonparametric paradigms.

Optimal Control. Model-based reinforcement learning founded in optimal control and trajectory optimization, and a focus optimal input synthesis for model identification.

Reinforcement Learning. Leveraging hierarchies in reinforcement learning to achieve simpler policy representation that enable interpretability and avoid over-parameterized models.

Robotics. Development and application of model-free learning approaches for complex real-world robotic tasks with a focus on contextual, imitation and curriculum learning.

Invited Talks

10/2019 **Preferred Networks, Tokyo, Japan.** *The Hybrid-System Paradigm: Learning Decomposition and Control of Nonlinear Systems.*

10/2019 **RIKEN Institute, Tokyo, Japan.** *The Hybrid-System Paradigm: Learning Decomposition and Control of Nonlinear Systems.*

11/2019 **ATR Institute, Kyoto, Japan.** *The Hybrid-System Paradigm: Learning Decomposition and Control of Nonlinear Systems.*

Teaching and Mentoring

2019 **Teaching Assistant.** *Statistical Machine Learning Lecture*, Intelligent Autonomous Systems Lab, Technische Universität Darmstadt

2018–2019 **Teaching Assistant.** *Reinforcement Learning Lecture: From Foundations to Deep Approaches*, Intelligent Autonomous Systems Lab, Technische Universität Darmstadt.

2018–2019 **Teaching Assistant.** *Reinforcement Learning Seminar: Algorithms and Platforms*, Intelligent Autonomous Systems Lab, Technische Universität Darmstadt.

2018–2019 **Teaching Assistant.** *Applications of Reinforcement Learning Lab*, Intelligent Autonomous Systems Lab, Technische Universität Darmstadt.

2018 **Teaching Assistant.** *Robot Learning Integrated Project*, Intelligent Autonomous Systems Lab, Technische Universität Darmstadt.

2016–2017 **Lecture Assistant.** *Operating Systems Lecture*, Computer Science Department, Technische Universität Darmstadt.

2015 **Student Mentor.** Electrical Engineering Department, Technische Universität Darmstadt.

2014 **Lecture Assistant.** *Intelligent Multi-Agent Systems Lecture*, Intelligent Autonomous Systems Lab, Technische Universität Darmstadt.

2011–2013 **Lab Assistant.** *Control Theory Lab*, Institute for Control and Mechatronics, Technische Universität Darmstadt.

Project Supervision

2019–2020 **Student Research Project.** Janosch Moos, Kay Hansel, *Foundations of Adversarial and Robust Learning.*

2017–2018 **Student Research Project.** Markus Semmler, Stefan Fabian, *Bayesian Inference for Switching Systems.*

2017 **Student Research Project.** Pascal Klink, *Online Dynamics Model Learning.*

2016–2017 **Student Research Project.** Martin Seiler, Max Kreischer, *Optimal Control for Biped Locomotion.*

2016 **Student Research Project.** Elvir Sabic, Alexander Wölker, *Juggeling with Robots.*

2016 **Student Research Project.** Manuel Bied, Felix Treede, Felix Pels, *Learning and Control for a Bipedal Walker.*

Thesis Supervision

2021 **Master Thesis.** Tom Buchholz,
Infinite Mixtures for Dimensionality Reduction.
2021 **Master Thesis.** Tim Schneider,
Tactile Active Inference.
2021 **Master Thesis.** Yannick Eich,
Distributionally Robust Hybrid Control.
2021 **Master Thesis.** Kay Hansel,
Probabilistic Dynamic Mode Primitives.
2021 **Master Thesis.** Janosch Moos,
Approximate Variational Inference For Mixture Models.
2020 **Master Thesis.** Tim Dorau,
Distributionally Robust Optimization for Optimal Control.
2020 **Master Thesis.** Thomas Lautenschläger,
Variational Inference for Switching Dynamics.
2020 **Master Thesis.** Markus Semmler,
Sequential Bayesian Optimal Experimental Design.
2019 **Master Thesis.** Peter Nickl,
Bayesian Inference using Nonparametric Infinite Mixtures.
2019 **Master Thesis.** Pascal Klink,
Generalization and Transferability in Reinforcement Learning.
2019 **Master Thesis.** Matthias Schultheis,
Approximate Bayesian RL for System Identification.
2018 **Master Thesis.** Onur Celik,
Chance-Constraints for Stochastic Optimal Control.
2018 **Bachelor Thesis.** Tim Schneider,
Guided Policy Search for In-Hand Manipulation.
2018 **Bachelor Thesis.** Ana Borg,
Infinite Mixture Policies in Reinforcement Learning.
2018 **Bachelor Thesis.** Nourhan Khaled,
Benchmarking RL Algorithms on Tetherball Games.

Internships

2014 **Honda Research Institute Europe.** 6 Months,
Learning Tactile Models for Object Localization.

Awards

2017 *Datenlotsen Award.* Prize for an outstanding master thesis,
Technische Universität Darmstadt.

Master Thesis

2016 *Trajectory Optimization and Stochastic Optimal Control with Linearized Dynamics.* Supervisors: Prof. Gerhard Neumann, Dr. Oleg Arenz, Prof. Jan Peters, Intelligent Autonomous Systems Lab, Technische Universität Darmstadt.

Scientific Reviewing

Conferences

2021 International Conference on Intelligent Robots and Systems
2021 International Conference on Automated Planning and Scheduling
2021 International Conference on Robotics and Automation
2020 International Conference on Automated Planning and Scheduling
2020 Conference on Neural Information Processing Systems
2020 Conference on Robot Learning
2019 Conference on Artificial Intelligence
2019 International Conference on Decision and Control
2019 American Control Conference
2019 International Conference on Intelligent Robots and Systems
2018 International Conference on Humanoid Robots
2018 International Conference on Decision and Control
2018 International Conference on Robotics and Automation
2018 International Conference on Intelligent Robots and Systems
2017 International Conference on Humanoid Robots
2017 International Conference on Decision and Control
2017 International Conference on Intelligent Robots and Systems
2016 International Conference on Humanoid Robots
2015 International Conference on Automation Science and Engineering
2015 International Conference on Robotics and Automation

Journals

2021 Robotics and Automation Letters
2020 Robotics and Automation Letters
2019 Robotics and Automation Letters
2019 Autonomous Robots