ESTIMATION AND CONTROL OF JUMP STOCHASTIC SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

Wee Chin Wong

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Chemical & Biomolecular Engineering

Georgia Institute of Technology
December 2009

ESTIMATION AND CONTROL
OF JUMP STOCHASTIC SYSTEMS

Approved by:

Professor Jay H. Lee, Advisor
School of Chemical & Biomolecular Engineering
Georgia Institute of Technology

Professor Erik Verriest
School of Electrical & Computer Engineering
Georgia Institute of Technology

Professor Matthew J. Realff
School of Chemical & Biomolecular Engineering
Georgia Institute of Technology

Professor Magnus Egerstedt
School of Electrical & Computer Engineering
Georgia Institute of Technology

Professor Martha Grover
School of Chemical & Biomolecular Engineering
Georgia Institute of Technology

Professor Alexander Gray
College of Computing
Georgia Institute of Technology

Date Approved: 17 August 2009

To my family and friends...

ACKNOWLEDGEMENTS

I find myself indebted to a number of people, without whom I could not have possibly come this far. Without a doubt, the first person who comes to mind is my advisor Professor Jay H. Lee, for all the guidance, advice, patient and support that he has given me in the past five years. Professor Lee has been a great motivation in terms of pursuit of academic excellence as well as professionalism. A primary reason for my attending Georgia Tech has been him. In this regard, I must say I have made an excellent choice.

I am also thankful to Professor Alexander Gray for the immense help, encouragement and guidance that he provided me as a committee member and more importantly as a friend. I have thoroughly enjoyed learning from him and the members of Fastlab. I wish them all the best in their future endeavors. The help I received from the other committee members Professors Matthew Realff, Martha Grover, Erik Verriest and Magnus Egerstedt is also greatly appreciated. I thank all these professors for agreeing to be on my thesis committee and giving me invaluable advice and time.

It is my sincere belief that my the friends I have made in the Lee research group are ones that I will keep forever. Special thanks goes to Nikolaos Pratikakis (my dear Greek brother), Farminder Anand, Ugur Guner, Prabuddha Bansal and Rakshita Agrawal for their friendship, for tolerating me and for seeing me through all the trials and tribulations of the past several years. We have spent important, formative years together and I am thankful for that. I have learnt so much from each and every single one of them and wish them well in everything they do.

I would also like to thank previous students of the Lee Group: Jongmin Lee and Niket Kaisare, both of whom are Professors in their own right for their help and research advice. I am also grateful to have had significant and meaningful academic and non-academic interactions with Thidarat (Tina) Tosukhowong and Anshul Dubey, who are also my seniors in this group.

I would like to show my appreciation to Owens Corning where I spent two fruitful summers as an intern. I would like to acknowledge Jim Beilstein, Chad Farschman, Wei Li, Mike Lewis, Yandong Pan, Bethany Ruiz, Mark Rindler and everyone else in Manufacturing Technology and Information Systems for their guidance and friendship.

Last but not least, I would like to thank my parents and dear family members. It is through trying times that I have learnt to appreciate the meaning of unconditional love and unwavering support. I sincerely apologize for being away from home for extended periods of time and for missing important milestones in the lives of my kin and closest friends.

I conclude by saying that I will miss Georgia Tech but am simultaneously eager to contribute to a better age.

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGEMENTS iv

LIST OF TABLES x

LIST OF FIGURES xi

SUMMARY xiii

I INTRODUCTION 1

1.1 Disturbance modeling 2

1.2 Stochastic control via approximate dynamic programming based on the post-decision state variable 2

1.3 Thesis outline 3

II BACKGROUND 5

2.1 Disturbance modeling and system identification for stochastic systems . . 5

2.2 Control of nonlinear stochastic systems 6

2.2.1 Model predictive control 6

2.2.2 Dynamic programming 8

III REALISITIC DISTURBANCE MODELING USING HIDDEN MARKOV MODELS 11

3.1 Introduction 11

3.2 Generalizations via a Hidden Markov Model framework 14

3.2.1 Hidden Markov Models 14

3.2.2 Mixture-of-Gaussians and Markov jump linear systems 14

3.2.3 Disturbance modeling using HMMs 15

3.3 System identification & state estimation of jump systems 17

3.3.1 System identification & practical considerations 18

3.3.2 State estimation 19

3.4 Example 1: HMM disturbance framework for offset-free linear model predictive control 24

3.4.1 Numerical example 27

3.5 Example 2: Rejecting & detecting deterministic step changes 31

3.5.1 Comparing closed-loop performance: different disturbance models 32

3.6 Conclusions, limitations & future work 36

IV HYBRID CYBERNETIC MODEL-BASED SIMULATION OF CONTINUOUS PRODUCTION OF LIGNOCELLULOSIC ETHANOL: REJECTING ABRUPTLY CHANGING FEED CONDITIONS 37

4.1 Introduction 37

4.2 Modeling abruptly changing feed disturbances via HMMs 41

4.2.1 Disturbance modeling via HMMs: unknown mean values of feed concentrations 43

4.2.2 State estimation for systems with Markovian jump parameters . . 44

4.3 A hybrid cybernetic model-based description of the continuous fermentor 45

4.3.1 Overview of a hybrid cybernetic chemostat model seeded with Saccharomyces yeast 1400(pLNH33) 46

4.4 Control strategy: successive linearization-based Model Predictive Control (slMPC) 49

4.4.1 p-step ahead prediction 50

4.4.2 Control move implementation 51

4.5 Selection of an appropriate controlled variable: simulation studies 52

4.5.1 Productivity as the CV 52

4.5.2 Investigating the incorporation of a soft constraint on conversion . 55

4.5.3 Conversion as the controlled variable of choice 56

4.6 Rejecting unmeasured feed disturbances 57

4.6.1 Various disturbance models: comparing closed-loop performance . 58

4.7 Conclusions 60

V FAULT DETECTION IN PROCESS SYSTEMS USING HIDDEN MARKOV DISTURBANCE MODELS 63

5.1 Introduction 63

5.2 Fault modeling using Hidden Markov Models 65

5.2.1 Proposed fault model: intermittent drifts & abrupt jumps 66

5.2.2 Fault detection via state estimation of jump Markov systems . . . 68

5.2.3 A-posteriori regime estimation 69

5.3 Example 1: Fault tracking in a shell & tube heat exchanger 70

5.3.1 Simulation conditions 70

5.3.2 Proposed HMM method to handle abrupt jumps & intermittent drifts 72
5.3.3 Alternative MOG description 73
5.3.4 Example 1: Results 74
5.4 Example 2: Valve stiction 76
5.4.1 Simulation studies: mixing tank 77
5.4.2 Results: estimating valve output & detecting valve stiction 78
5.5 Conclusions & future work 78
VI CONTROLLING JUMP STOCHASTIC SYSTEMS VIA POST-DECISION-STATE-BASED APPROXIMATE DYNAMIC PROGRAMMING 79
6.1 Introduction 79
6.1.1 Dynamic programming for solving multi-stage stochastic control problems 81
6.2 Pre and post-decision-state-based dynamic programming 82
6.3 Approximate dynamic programming based on the post-decision state . . 85
6.3.1 Function approximation & stable learning 88
6.3.2 Kernel regression 90
6.3.3 Cautious learning: preventing over-extrapolations 92
6.4 Post-decision-state-based Bellman equations for jump stochastic systems 94
6.4.1 Example 1: Control of a constrained Markov Jump Linear System 95
6.5 Control of Stochastic Systems: other examples 97
6.5.1 Example 2: Interaction between state estimation and control . . . 97
6.5.2 Example 3: Dual control of an integrator with an unknown step change in gain 102
6.5.3 Example 4: Constrained linear stochastic system- double integrator problem 106
6.5.4 Example 5: Constrained chemostat - maximizing productivity subject to conversion constraints 108
6.6 Conclusions 111
VII CONTRIBUTIONS & FUTURE RESEARCH 113
7.1 Summary of contributions 113
7.1.1 Disturbance modeling 113
7.1.2 Post-decision-state-based approximate dynamic programming . . . 114

7.2 Future work 114

APPENDIX A DETAILS FOR HYBRID CYBERNETIC MODEL-BASED SIMULATION OF CONTINUOUS PRODUCTION OF LIGNOCELLULOSIC ETHANOL 116

APPENDIX B PROOF: CONVERGENCE OF POST-DECISION-STATE-BASED VI USING LOCAL FUNCTION APPROXIMATORS 119

REFERENCES 120

VITA 128

LIST OF TABLES

1 Noise parameters used in simulation studies 28

2 Mean of normalized squared error over 500 realizations 30

3 Tracking error corresponding to various disturbance models, in response to the introduction of a step disturbance at $t = t^* = 175$. The operator, $< \cdot >$, denotes sample average over 100 realizations. 34

4 Re-running experiments in Table 3 with a de-tuned GPB2 estimator. 36

5 Entering mean feed concentration for various regimes 40

6 Description of subscripts used in Eq. (45) 47

7 Rate-of-investment for each elementary flux mode. 48

8 Nominal steady state operating conditions 71

9 2-norm of state-estimation error (Average of 100 realizations) 74

10 Interaction between estimation & control: performance over 100 realizations 102

11 Dual control of integrator with an unknown gain: mean performance over 100 realizations 106

12 Double integrator example: comparing closed-loop performance 108

13 Kinetic parameters for Eqs. (148)-(149). 118

LIST OF FIGURES

1 Realistic disturbance patterns modeled by proposed framework. 12

2 Schematic illustrating the GPB2 algorithm for a two-regime problem. . . . 21

3 Comparing state estimators reacting to a regime change. 24

4 Benchmarked output plots for various combinations of employed controller/ estimator schemes and simulation scenarios. 30

5 Comparing closed loop performance of various controller/ estimator pairs: plots of $||z_t||_1$ vs. time for a typical stochastic realization. 35

6 Estimating presence of step disturbance. 35

7 Continuous mode ethanol production (adapted from [57]). Input glucose (GLC) and xylose (XYL) concentrations may experience severe fluctuations due to the diversity of raw materials and/ or changes in intermediate processing steps. 38

8 Postulated highly varying feed conditions (manifested in entering glucose and xylose concentrations). 39

9 Steady state productivity and conversion profiles for various entering glucose/ xylose (g/L) concentrations. The faint (red) lines correspond to the 95% conversion limit. 53

10 Plots of productivity, conversion and dilution rate in response to a sudden decrease (from regime 1 (80/40) to regime 3 (60/30)) in feed conditions at $t = 5$. Note the constraint violation despite decreasing dilution rate to 0 (1/h). 54

11 Investigating closed loop performance as a function of penalty weights in response to a shift in feed conditions from regime 1 to regime 3. 56

12 State profiles corresponding to various disturbance models: Proposed HMM method vs. IWN(lo) 60

13 State profiles corresponding to various disturbance models: Proposed HMM moethod vs. IWN(hi) 61

14 Possible disturbance signals (θ). 64

15 Tracking θ^u and θ^y. Comparing the proposed HMM vs. MOG approaches. Legend: solid line - actual fault signal; Dots ($\cdot$) - HMM; Crosses (x) - MOG 75

16 Tracking unmeasured valve output in mixing-tank example. 78

17 Proposed ADP algorithm based on the post-decision state. 87

18 Comparing post-decision state based ADP vs. LMPC. The Markov state switched from $r = 1$ to $r = 2$ midway through the simulation. 97

19 Interaction between estimation & control: plot of learning error vs. VI iterations. 100

20 Interaction between estimation & control: x vs. t for a typical realization . 101

21 Interaction between estimation & control: u vs. t for a typical realization . 101

22 Interaction between estimation & control: $P_{t|t}$ vs. t for a typical realization. 101

23 Dual control: a typical realization. b jumps from an initial value of 0.5 to 15 at $t = 3$ 105

24 Double integrator example: x_2 vs. t for 500 realizations. Lower bound for x_2 is 0. 109

25 Steady state profiles 110

26 Constrained non-linear chemostat example: closed-loop performance of a typical realization. ADP: solid line (-); slMPC: dotted line(..); lower bound on conversion: dash-dot (-.) 112

SUMMARY

This thesis deals with the estimation and control of jump stochastic systems as a result of the novel proposition of a framework based on switching hidden modes to enhance the treatment of uncertainty within the process control field. Uncertainty is typically manifested as uncontrolled, exogenous disturbance signals affecting the input and/ or output channels as well as parameters. In the area of disturbance modeling, the assumption of stationary, filtered white noise has been the norm. The description of non-stationary signals, which are common in process industries, is limited to random walk-type signals. Understandably, the lack of sophistication in modeling has led to inadequacies along the control front. This is typified by Model-based Predictive Control (MPC), the de facto advanced process control solution. There, the common practice is to solve a rolling-horizon open-loop optimal mathematical program by making deterministic assumptions about the future trajectory of noise signals and uncertainty. Although the rolling-horizon mechanism imparts some robustness against uncertainty, the general situation is that overall performance falls short of closed-loop optimality. This lack of a systematic treatment of uncertainty is a fundamental limitation of MPC.

This thesis begins with the realistic modeling of non-stationary disturbance signals typically witnessed in process industries. Such disturbances are characterized by probabilistic switches between distinct regimes. To capture the effect of such multiple modes, a Hidden Markov Model (HMM) framework is employed. The main disturbance patterns of concern considered in this thesis include intermittent drifts and abrupt jumps. The main idea is to superimpose a Markov chain, whose parameters govern the latent temporal regime transitions on top of a discrete-time equation that governs the underlying system dynamics to generate a (concatenated) Markov Jump System (MJS). Several examples are given in the thesis which demonstrate the usefulness of the HMM framework in the context of describing an array of disturbance patterns, providing integral action and the provision for

model-based fault detection. The main contribution is practical in that with the adoption of a more sophisticated disturbance model, and the consequential use of an existing state estimator suited for MJS's, superior tracking performance is possible. Enhanced closed-loop control, without having to re-design the controller (which remains as MPC for this part of the thesis), is the result.

Having dealt with the issue of modeling, the thesis then proceeds to the optimal control of stochastic systems, including the aforementioned jump systems. The control framework employed is Approximate Dynamic Programming (ADP) based on a "post-decision state", as opposed to the more common "pre-decision" state. ADP is a promising framework for systematically and practically obtaining good closed-loop policies for multi-stage, stochastic control problems. The main advantage of ADP over MPC is the former's ability to account for uncertainty in a systematic fashion. Furthermore, for ADP, the bulk of the computation burden is shifted *off-line* (during a process termed Value Iteration (VI), via which "value-functions" are computed). *Online* computations are oftentimes swifter, since the solution to a single-stage optimization problem (as opposed to a multi-stage one, in the case of MPC) is required. The contributions in this part of the thesis are multi-fold.

Most previous works on ADP, as applied in the context of process control, have focused on solving deterministic problems. For stochastic control problems, the pre-decision ADP formulation involves a computationally cumbersome optimization over an expected quantity during on-line and off-line calculations. Through the use of the post-decision state, the typically non-commutative optimization and expectation operations are interchanged to yield an equivalent problem. The post-decision stage formulation also allows the efficient use of off-the-shelf optimization solvers which form the cornerstone of MPC technology. A further benefit is that off-line VI calculations may be run in parallel. This thesis extends previous ADP methodologies (involving simulations to uncover relevant parts, over which VI is performed, of the state-space and function approximation) to the post-decision analogue.

The proposed post-decision-state-based ADP approach is demonstrated on but not limited to the control of stochastic jump systems. Several examples are used to demonstrate the

algorithm and to highlight the inadequacy of MPC (and/ or other popular control methodologies) in providing good closed-loop control due to its ad-hoc treatment of uncertainty. These examples include a dual control problem (of an integrator with an unknown gain) and a case study highlighting the importance of considering the oftentimes over-looked interaction between state-estimation and control. Another example is one of a constrained double integrator, where MPC leads to frequent violations of the constraints due to unbounded prediction errors. The last pertains to a bioreactor chemostat where the aim of maximizing productivity whilst ensuring high conversion results in operations at the constraint boundary. The proposed ADP solution is able to automatically "back-off" from the constraint boundary in the face of disturbances.

The following peer-reviewed publications have resulted from this work:

- Wong, W. C. and J. H. Lee, "Realistic disturbance modeling using Hidden Markov Models: applications in model-based process control," *Journal of Process Control*, accepted.

- Wong, W. C., H. S. Song, J. H. Lee, and D. Ramkrishna, "Hybrid cybernetic model-based simulation of continuous production of lignocellulosic ethanol: rejecting abruptly changing feed conditions," *Control Engineering Practice*, under review

- Wong, W. C. and J. H. Lee, "A reinforcement learning-based Scheme for direct adaptive optimal control of linear stochastic systems," *Optimal Control Applications and Methods*, accepted

- Wong, W.C. and J. H. Lee, "Approximate dynamic programming for process control," International Symposium on Advanced Control of Chemical Processes 2009[1].

The first two publications are related to the disturbance modeling part of the thesis whereas the last two are associated with the latter half.

[1] An extended version of this is under preparation for submission to a peer-reviewed journal

CHAPTER I

INTRODUCTION

The appropriate treatment of uncertainty in the context of process control (and other similar multi-stage decision making) problems is of utmost importance. Indeed, the presence of disturbances (a typical manifestation of uncertainty) is one of the primary reasons for adopting advanced control solutions [3] such as Linear Quadratic Gaussian controllers (LQG) and Model Predictive Control (MPC). Although it is undeniable that the widespread adoption of Advanced Process Control (APC) solutions has yielded much economic benefit [73] to process industries, existing solutions do not address uncertainty adequately.

First and foremost, current approaches towards the *modeling* of disturbances (in either the input and/ or output channels or that pertaining to parameter values) are still dominated by the additive, stationary noise paradigm [82]. With regards to the subsequent issue of *control*, MPC (the de-facto APC solution) is fundamentally limited in that it solves a math program at every time instant, assuming that future disturbances and uncertainties follow a pre-determined trajectory. The result is an open-loop optimal deterministic formulation. This means that future feedback, this being generally beneficial for control performance, is not considered in the formulation. Poor closed-loop performance is the result. Furthermore, it is well known that MPC is oftentimes plagued by exorbitant on-line computational requirements in the course of solving large-scale, non-convex mathematical programs.

As such, the main objective of the thesis is first to explore and provide solutions for superior disturbance modeling (see Section 1.1). The overall result is a Markov Jump description of the system of concern which is better able to model commonly witnessed disturbance signals.

Subsequent to this, the optimal control of (jump and non-jump) stochastic systems based on the rigorous framework of Dynamic Programming (DP) (see Section 1.2) is explored.

Brief descriptions are given in the following sections.

1.1 Disturbance modeling

Understanding and modeling disturbances play a critical part in designing effective advanced model-based control solutions. Existing linear, stationary disturbance models are oftentimes limiting in the face of time-varying characteristics typically witnessed in process industries. These include intermittent drifts, abrupt changes, temporary oscillations, outliers and the likes. This work proposes a Hidden-Markov-Model-based framework to deal with such situations that exhibit discrete, modal behavior. The usefulness of the proposed disturbance framework - from modeling to ensuring the integral action under a wide variety of scenarios - is demonstrated through several examples. The HMM framework is also applied to the problem of fault detection and the production of second generation bio-ethanol subject to highly varying feed conditions. Fermenting various sugars derived from lignocellulosic biomass promises to be attractive for producing ethanol, an important alternative fuel. Diversity of lignocellulosic biomass sources and pre-processing variations mean entering sugars are expected to experience large, though infrequent, changes. Recent developments in hybrid cybernetic modeling allow efficient in-silico studies. This enables studying sequential linearization-based model predictive control for ensuring high productivity and conversion, for a chemostat seeded with yeast capable of co-fermentation. An appropriate controlled variable (conversion) and control formulation are ascertained.

1.2 Stochastic control via approximate dynamic programming based on the post-decision state variable

Dynamic Programming (DP) [7] represents a unified framework for solving stochastic, multi-stage control problems found in process industries and other application areas. Central to DP is the cost-to-go function (which scores the desirability of any arbitrary state) which can be theoretically obtained by solving (usually off-line) for the fixed point of Bellman's optimality equations [7]. Optimal control is achieved through the on-line solution of a single-stage problem which reflects the tradeoff between immediate costs (manifested through a single-stage term) and future costs (reflected by the value function of a candidate next

state).

Systems with large state and action spaces suffer from a 'curse of dimensionality', where representing and obtaining the cost-to-go compactly and efficiently becomes highly non-trivial. To circumvent this, the authors of [47, 48, 49, 44, 46] proposed an Approximate Dynamic Programming (ADP) method for solving process control problems, which suffer all the more from the said curse due to the presence of continuous state and action spaces. The basic idea is to use carefully-designed simulations to uncover a control-relevant part of the state space (which is finite-sized subset of the original state space) and employ an appropriate function approximator for generalization. The focus there was mainly on the control of deterministic, nonlinear systems. For stochastic systems, however, the off-line and on-line computations involving Bellman's equations require a minimization over the sum of a single-stage cost and the expected value function of a candidate next state. Since an analytical expression for the expectation is usually unknown, solving such a problem may be cumbersome.

The second part of the thesis involves the systematic extension of the ADP framework to stochastic systems. In doing so, it will be demonstrated that the introduction of an intermediate 'post-decision' state [80, 71] allows the generally non-commutative min and $\mathbb{E}$ operators to be interchanged. This means the off-line Bellman computations may be executed as various single-stage deterministic optimizations running in parallel through the use of off-the-shelf solvers. It is noted that the latter have been the cornerstone of MPC technology.

1.3 Thesis outline

The thesis is organized as follows. Chapter 2 provides a brief background as well as a literature review regarding commonly employed modeling and control strategies. Chapter 3 explores a Hidden Markov Model (HMM) framework for effectively describing commonly witnessed disturbance patterns. The application of the proposed HMM framework to the production of second generation lignocellulosic bioethanol, a major alternative fuel, is explored in Chapter 4. Work on fault detection in process systems using Hidden Markov

Disturbance Models is presented in Chapter 5. A post-decision-state-based Approximate Dynamic Programming (ADP) framework is described in Chapter 6. Concluding remarks summarizing the contributions and future research are given in the final Chapter (Chapter 7).

CHAPTER II

BACKGROUND

2.1 Disturbance modeling and system identification for stochastic systems

System identification plays a vital role in model-based control. For example, Model Predictive Control (MPC) relies on an appropriate description of plant dynamics so that future behavior may be predicted and optimized. Within this, disturbance modeling is crucial for it accounts for the effect of unmeasured signals, parameter changes, as well as other unmodeled phenomena (the residuals). Indeed, one of the main aims of control is to mitigate the impact of disturbances. Appropriate disturbance modeling leads to more accurate predictions of key variables' behavior, and consequently, superior control [3]. It can also allow for the better capturing of cross-correlations among the various output channels, which is useful for soft-sensing and inferential control. Furthermore, offset-free control is often imparted by appending integrating disturbance states [78, 65, 68] to the process model.

Eqs. (1)- (2) together form a general representation of a dynamical system. Here, $x_t \in \mathbb{R}^{n_x}$ is the system state at discrete-time index t, $u_t \in \mathbb{R}^{n_u}$, a vector of manipulated variables, $\omega_t \in \mathbb{R}^{n_x}$, additive state noise, $v_t \in \mathbb{R}^{n_y}$, additive measurement noise, and $y_t \in \mathbb{R}^{n_y}$, output measurements. $f(.)$, $g(.)$ are models of the state transition and measurement dynamics respectively.

$$x_{t+1} = f(x_t, u_t, \theta_{t+1}) + \omega_t \tag{1}$$

$$y_t = g(x_t, \theta_t) + v_t \tag{2}$$

$\theta_t \in \mathbb{R}^{n_\theta}$ is a vector of disturbance signals. The use of θ provides flexibility as it is sufficiently general to represent a variety of possible disturbance patterns, which, for practical purposes, is usually determined by the user. For example, the authors of [37], use θ as a fault parameter vector to describe changes in process and disturbance parameters as well as actuator and sensor problems. Common deterministic signals for θ_t include the pulse,

step changes (e.g. in Dynamic Matrix Control [18]), ramps, and sinusoids [3]. Stochastic processes lend themselves to an additional level of sophistication and generalization, given the typically-assumed probabilistic nature of disturbances. Eq. (3) (see [45]) is capable of modeling common stochastic signals (and indeed deterministic ones), through appropriate choices of $\mathbb{A} \in \mathbb{R}^{n_\gamma \times n_\gamma}$, $\mathbb{B} \in \mathbb{R}^{n_\gamma \times n_\varphi}$, $\mathbb{C} \in \mathbb{R}^{n_\theta \times n_\gamma}$, and the mathematical forms of $\varphi \in \mathbb{R}^{n_\varphi}$ and $e \in \mathbb{R}^{n_\theta}$.

$$\begin{aligned} \gamma_{t+1} &= \mathbb{A}\gamma_t + \mathbb{B}\varphi_{t+1} \\ \theta_t &= \mathbb{C}\gamma_t + e_t \end{aligned} \tag{3}$$

In the stochastic processes literature, θ_t is commonly approximated as a stationary noise, a white noise driven through a linear filter parameterized by $\mathbb{A}, \mathbb{B}, \mathbb{C}$. Eq. (3) is extended to allow for 'random walk'-type behavior when $\mathbb{A} = \mathbb{B} = \mathbb{C} = I$, $e_t = 0, \forall t$ and φ is modeled as a white noise. Such low frequency changes are commonly found in chemical process industries [62, 63, 16].

2.2 Control of nonlinear stochastic systems

2.2.1 Model predictive control

Model predictive control (MPC) is a technique in which the current control action is obtained by minimizing online, a cost criterion defined on a finite time interval. Nominal deterministic trajectories of future disturbance signals and uncertainties are necessarily assumed in order to obtain an optimization problem amenable to on-line solution via math programming. The solution generates a control sequence from which the first element is extracted and implemented. The procedure is repeated at the next time instant. Owing to its ability to handle constrained, multi-variable control problems in an optimal manner, MPC has become the de-facto advanced process control solution for the process industries today. Consider a generic process model:

$$\begin{aligned} \tfrac{dx}{dt} &= f_c(x, u, d) \\ y &= g(x) \end{aligned} \tag{4}$$

where $x \in \mathbb{R}^{n_x}$ refers to the system state, u, the input vector, y, an output vector and d a disturbance signal. Since MPC is typically implemented digitally, a discrete-time equivalent is given by:

$$x_{t+1} \quad = \quad f_h(x_t, u_t, d_t) \tag{5}$$

where $f_h(x_t, u_t)$ denotes the terminal state vector obtained by integrating f_c across a time interval of duration h with the initial condition x_t and u_t and d_t held constant across the interval.

The on-line control action is calculated by solving the following open-loop optimal control problem at each sample time:

$$\min_{\{\nu_k\}_{k=0}^{m-1}} \left\{ \sum_{k=0}^{p-1} \phi(\tilde{x}_k, \nu_k) + \phi_p(\tilde{x}_p) \right\} \tag{6}$$

where p refers to the prediction horizon, $m \leq p$, the control horizon ($\nu_m, \nu_{m+1}, \ldots, \nu_{p-1}$ are all constrained to be zero), ϕ a single-stage cost, and ϕ_p, the terminal stage cost. $\tilde{x}_0$ is initialized as x_t (or $x_{t|t}$ if a state estimator is used). The relationship between $\tilde{x}$ and one time instant to the next is governed by f_h. A deterministic trajectory (e.g. one might assume that $d_k = 0$ for the entire prediction horizon) of the disturbance vector is necessarily assumed. The first element of the sequence $\{\nu_t\}_{t=0}^{m-1}$ is extracted and injected into the plant. Evidently, the policy instructed by the MPC is implicitly given by the above mentioned optimization problem.

MPC is by now, considered to be a mature technology owing to the plethora of research and industrial experiences during the past three decades. Despite this, it has some fundamental limitations, which prevents it from being a panacea for all process control problems. One well-known limitation is the potentially exorbitant on-line computation required for solving a large-scale, and potentially non-convex math program that scales with the dimension of the state as well as the length of prediction horizon. Recent developments ([43]) have made some headway in tackling this problem although nontrivial computational challenges still exist.

The second limitation arises from the fact that the deterministic formulation adopted by MPC is inherently incapable of addressing uncertainty in a closed-loop optimal fashion.

Its open-loop optimal control formulation used to find the control moves at each sample time means the fact that information about future uncertainty will be revealed, this being generally beneficial for control performance, is not considered. Most of the past attempts at ameliorating the impact of uncertainty has been reflected in robust MPCs formulations based on the objective of minimizing the worst-case scenarios ([86]) at the expense of overly conservative policies. Multi-scenario formulations ([43]) have also been developed but the number of scenarios is limited and they do not give closed-loop optimal policies in general. Stochastic programming based methodologies ([70]) allow for recourse actions at the computational expense of enumerating an exponentially growing number of scenarios.

2.2.2 Dynamic programming

Consider the optimal control of the following discrete-time stochastic system:

$$x_{t+1} \quad = \quad f(x_t, u_t, \omega_t) \tag{7}$$

where $x_t \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ refers to the system state at discrete time index t, $u_t \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ a control or action vector, and ω_t an exogenous, unmeasured, stochastic signal. x may contain physically meaningful states as well as measured disturbances, and parameters subject to uncertainty. f refers to the single-stage transition function. For problems where the system's dynamics are represented by ordinary differential equations, f is then the result of numerical integration across a single sample-time, with vectors u and ω held constant. Throughout this chapter, it is assumed that full state feedback is available. In the event that only output feedback is available, x is interpreted as an information vector that contains the sufficient statistics of the state estimate's probability density function. Such lifting is possible as the information vector is governed by another related set of equations (*i.e.*, the filter dynamics).

Let $\mu \in \Gamma$ be a 'state-feedback policy' that maps the state vector to the action vector, where Γ represents the set of all admissible (stationary) such policies. $J^\mu(x)$ will be used to denote the 'cost-to-go' function, which is defined as the infinite horizon, discounted sum of the stage-wise costs under the policy μ starting from an arbitrary state x:

$$J^\mu(x) \quad = \quad \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \phi(x_k, u_k = \mu(x_k)) | x_0 = x\right] \tag{8}$$

where ϕ represents a pre-specified stage-wise cost (*e.g.* $\phi(x,u) := ||x||_Q^2 + ||x||_R^2$) and $\gamma \in [0,1)$ is a discount factor. The goal then is to find the optimal (stationary) policy $\mu^* : \mathcal{X} \rightarrow \mathcal{U}$, that yields the minimum cost-to-go function as below:

$$J^{\mu^*}(x) = \min_{\mu \in \Gamma} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \phi(x_{t+k}, u_{t+k} = \mu(x_{t+k})) | x_t = x\right] \tag{9}$$

$J^{\mu^*} : \mathcal{X} \rightarrow \mathbb{R}^{0+}$ is the *optimal* 'cost-to-go' function and is an indication of the attractiveness of a given state in terms of future rewards. By definition, $J^{\mu^*}(x) \leq J^{\mu}(x), \forall x$ and $\forall \mu \in \Gamma$.

Based on the principle of optimality [7], one is able to re-write (9), thereby obtaining Bellman's optimality equations:

$$\begin{aligned} J^{\mu^*}(x) &= \min_{u \in \mathcal{U}} \left\{\phi(x,u) + \gamma \mathbb{E}_{(\omega|x)}[J^{\mu^*}(f(x,u,\omega))]\right\} \\ &= \left(TJ^{\mu^*}\right)(x) \end{aligned} \tag{10}$$

T above represents the single-pass DP operator represented by the minimization operation. The optimal policy is implicitly obtained through the solution of the associated single-stage optimization:

$$\mu^*(x) = \arg\min_{u \in \mathcal{U}} \left\{\phi(x,u) + \gamma \mathbb{E}_{(\omega|x)}[J^{\mu^*}(f(x,u,\omega))]\right\} \tag{11}$$

In principle, the optimal control problem is solved once J^{μ^*} is known. The repeated application of T on an arbitrarily initialized cost-to-go leads to convergence and underpins the idea behind Value Iteration (VI).

$$J^{\mu^*}(x) = TJ^{\mu^*}(x) = \lim_{i \rightarrow \infty} (T)^i J^{\mu}(x),\ \forall \mu, x \tag{12}$$

VI's computational requirements grow as $O(|\mathcal{X}|^2|\mathcal{U}|)$. Therefore for continuous states, as is typical for process control, DP methods are computationally intractable. An exception is LQG control, a method of designing feedback control laws for linear systems with additive Gaussian noise processes that minimize a given quadratic cost functional. Due to the aforementioned 'Curse-of-Dimensionality', approximate methods making judicious use of simulations, discretization and function approximation have been explored. [35] contains

comprehensive references on this topic. The adaptation of these ideas for the purpose of chemical process control was explored in [47, 48, 44, 46]. Applications included data-based nonlinear process control, dual control, and project scheduling.

CHAPTER III

REALISITIC DISTURBANCE MODELING USING HIDDEN MARKOV MODELS

3.1 Introduction

Eqs. (13)-(14) together form a general representation of a dynamical system. Here, $x_t \in \mathbb{R}^{n_x}$ is the system state at discrete-time index t, $u_t \in \mathbb{R}^{n_u}$, a vector of manipulated variables, $\omega_t \in \mathbb{R}^{n_x}$, additive state noise, $v_t \in \mathbb{R}^{n_y}$, additive measurement noise, and $y_t \in \mathbb{R}^{n_y}$, output measurements. $f(.)$, $g(.)$ are models of the state transition and measurement dynamics respectively.

$$x_{t+1} = f(x_t, u_t, \theta_{t+1}) + \omega_t \tag{13}$$

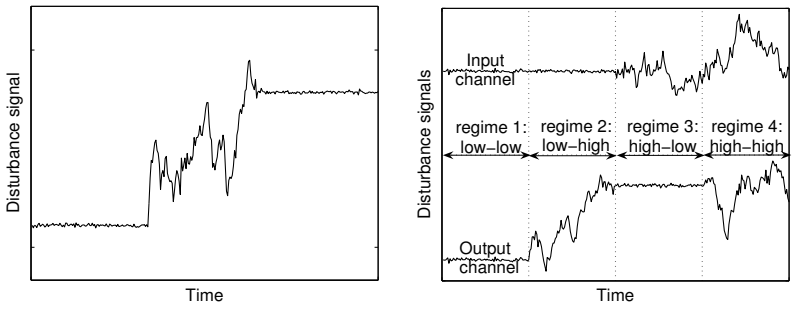
$$y_t = g(x_t, \theta_t) + v_t \tag{14}$$

$\theta_t \in \mathbb{R}^{n_\theta}$ is a vector of disturbance signals. The use of θ provides flexibility as it is sufficiently general to represent a variety of possible disturbance patterns, which, for practical purposes, is usually determined by the user. For example, the authors of [37], use θ as a fault parameter vector to describe changes in process and disturbance parameters as well as actuator and sensor problems. Common deterministic signals for θ_t include the pulse, step changes (e.g. in Dynamic Matrix Control [18]), ramps, and sinusoids [3]. Stochastic processes lend themselves to an additional level of sophistication and generalization, given the typically-assumed probabilistic nature of disturbances. Eq. (15) (see [45]) is capable of modeling common stochastic signals (and indeed deterministic ones), through appropriate choices of $\mathbb{A} \in \mathbb{R}^{n_\gamma \times n_\gamma}$, $\mathbb{B} \in \mathbb{R}^{n_\gamma \times n_\varphi}$, $\mathbb{C} \in \mathbb{R}^{n_\theta \times n_\gamma}$, and the mathematical forms of $\varphi \in \mathbb{R}^{n_\varphi}$ and $e \in \mathbb{R}^{n_\theta}$.

$$\begin{aligned} \gamma_{t+1} &= \mathbb{A}\gamma_t + \mathbb{B}\varphi_{t+1} \\ \theta_t &= \mathbb{C}\gamma_t + e_t \end{aligned} \tag{15}$$

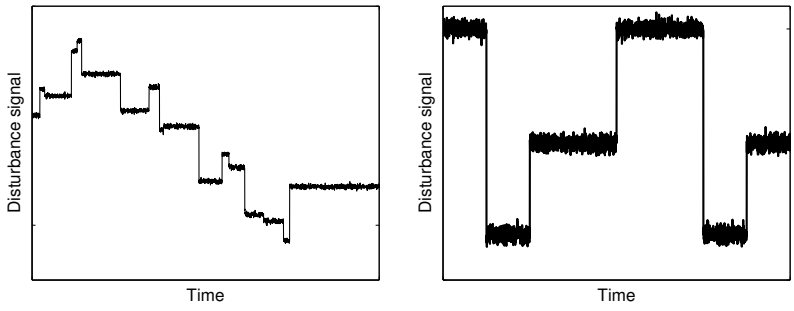
In the stochastic processes literature, θ_t is commonly approximated as a stationary noise, a white noise driven through a linear filter parameterized by $\mathbb{A}, \mathbb{B}, \mathbb{C}$. Eq. (15) is extended to allow for 'random walk'-type behavior when $\mathbb{A} = \mathbb{B} = \mathbb{C} = I$, $e_t = 0, \forall t$ and φ is modeled as a white noise. Such low frequency changes are commonly found in chemical process industries [62, 63, 16].

Though routinely adopted, the classical approach towards disturbance modeling is limiting in the face of more complex disturbance patterns seen in real processes. Such behavior includes intermittent drifts, abrupt jumps and temporary oscillations - all commonly witnessed patterns in process industries. For the purpose of illustration, consider that depicted in Figure 1(a), a time series of an entering disturbance signal (θ). It clearly exhibits dual-regime behavior. Namely, there exists "quiet" periods of relatively small high frequency process noise with occasional injections of intervals of random walk-like behavior. Another



(a) White noise probabilistically interspersed with integrated white noise.

(b) Cross-correlated input and output disturbances, each exhibiting intermittent drifts.



(c) Abrupt step changes of random duration and magnitude.

(d) Abrupt shifts in mean values.

Figure 1: Realistic disturbance patterns modeled by proposed framework.

scenario, characterized by the existence of multiple regimes or modes, is shown in Figure

1(b). Specifically, it reveals four regime permutations where input and/ or output disturbances are dominant. For example, 'low:low' (regime 1) indicates that both input and output disturbances enter as white noises. 'low:high' (regime 2) suggests that significant non-stationary, random-walk type disturbances are entering the output channel whereas input disturbances are relatively quiescent, and so on.

Other commonly seen disturbance signals include step changes of arbitrary duration and magnitude, as depicted in Figure 1(c). A slight variation is depicted in Figure 1(d) where the disturbance signal assumes, randomly, mean values/ biases from elements of a fixed set. In either case, regime 2 (say) may correspond to the relatively rare jump event.

The provision of a generalized framework for the modeling of the aforementioned behavior, revealing continuous and discrete[1] or modal dynamics, is the focus of this work. Accordingly, the use of Hidden Markov Models (HMMs) in providing a significant generalization of the current model form for disturbance modeling is explored. Another contribution of this chapter is to demonstrate that a wide range of existing model-based control formulations can be easily adjusted to integrate the proposed HMM-based disturbance framework. Such ease-of-use allows control practitioners to reap the benefits of realistic disturbance modeling in an immediate and relevant way.

The details of how the generalization may be achieved via HMMs are given in Section 3.2. Comments on system identification and state estimation for systems perturbed by disturbances modeled through HMMs are provided in Section 3.3. Three examples are presented in Sections 3.4 - 3.5 to demonstrate the usefulness of the proposed framework in a wide range of process control methods and applications. Section 3.4 shows that the HMM-based disturbance model, used to impart integral action to Linear Model Predictive Control (LMPC), is robust to a variety of stochastic disturbance scenarios. Section 3.5 demonstrates the flexibility of the proposed framework by extending it outside stochastic disturbances, to the detection and rejection of deterministic step changes. Section 3.6 concludes the chapter and provides comments on outstanding issues.

[1] not to be confused with discrete-time dynamics

3.2 Generalizations via a Hidden Markov Model framework

3.2.1 Hidden Markov Models

HMMs represent a useful class of statistical models where a latent state, taking values from an alphabet $\mathcal{J} \triangleq \{1, 2, \ldots, J \in \mathbb{Z}_+\}$[2] of cardinality J, transitions probabilistically in a Markovian fashion[3] from one sampling time to the next. Mathematically, a finite-state Markov chain is a sequence of random variables $(r_0, r_1, \ldots, r_t, \ldots)$, where matrix $\Pi = (\pi_{ij}) = (pr(r_t = j | r_{t-1} = i), i, j \in \mathcal{J}) : \sum_{j=1}^{J} \pi_{ij} = 1, \forall i \in \mathcal{J}$, governs the probabilistic temporal transitions. The term 'Hidden' signifies that the actual regime label is usually not known with complete certainty and must be inferred from available noisy measurements of itself or other related states. In the simplest case, each latent state has a probability distribution over a finite set of possible output symbols.

Hidden Markov Models have found widespread applications in science and engineering since the 1960s. Speech recognition [74] is one such area. Each candidate word has an HMM associated to it; word recognition consists of choosing the model that generates the largest likelihood. Kim [38] applied an HMM on top of a state-space model for modeling econometric time-series. There, the hidden states corresponded to different structural periods of the economy and determined the model parameters accordingly. In the field of bioinformatics, DNA sequencing via HMM analysis is now ubiquitous. In process control, Morales-Menendez et al. [61] modeled an industrial heat exchanger where different Markovian states corresponded to different flow-rate regimes, each of which was represented by a linear state-space approximation. Bar-Shalom and Li [4] describe the tracking of targets whose motions switch between models along their trajectories via Markov jump systems.

3.2.2 Mixture-of-Gaussians and Markov jump linear systems

In the literature, the modeling of disturbances with modal behavior has typically been achieved via a Mixture-of-Gaussians (MOG) formalism, which can be considered to be a

[2] The choice of using positive integers to denote all possible realizations of the latent state is made for notational convenience and generality. Therefore, a set containing labels composed of strings, characters, or other data types is also possible.

[3] *i.e.*, the transitions depend only upon the immediate past

special case of HMMs. At each time instant, each latent state would occur with a certain probability that is independent of past realizations, and would be realized from a particular Gaussian distribution. Robertson and Lee [79], for instance, modeled abrupt changes in such a manner albeit with an eventual focus on state estimation. For the purpose of data rectification, Singhal and Seborg [88] used an MOG approach to model infrequent outliers in measurement noise. By allowing for arbitrary probabilistic transitions among the hidden states, the proposed HMM framework is more general than the MOG methodology.

Following the successes in other fields, a generalization of Eq. (15) is considered by allowing the statistics of φ, e and potentially the parameters of $\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}$ to vary according to a hidden Markov chain. Namely, one obtains Eq. (16):

$$\begin{aligned} \gamma_{t+1} &= \mathbb{A}_{r_{t+1}}\gamma_t + \mathbb{B}_{r_{t+1}}\varphi_{t+1} \\ \theta_t &= \mathbb{C}_{r_t}\gamma_t + e_t \\ pr(r_t = j | r_{t-1} = i) &= \pi_{ij} \end{aligned} \tag{16}$$

The notation $\mathbb{A}_{r_t}$ means that at time t, $\mathbb{A}$ may take values from J possible candidates according to the realization of the hidden state, and so on. In addition to the system parameters, the probability distribution from which φ_t and e_t are sampled may vary according to the realization of the hidden state r_t.

In this chapter, φ_t is restricted to be a Gaussian variable, i.e., $\varphi_t \sim \mathcal{N}(\mu^{\varphi}_{r_t}, Q^{\varphi}_{r_t})$. A similar restriction is imposed on $e_t \sim \mathcal{N}(\mu^{e}_{r_t}, Q^{e}_{r_t})$. This resulting framework is termed a 'Markov Jump Linear System' (MJLS) [17] and is of sufficient flexibility to describe the scenarios mentioned earlier.

3.2.3 Disturbance modeling using HMMs

The form of Eq. (16) affords a great deal of flexibility in disturbance modeling. Whilst the values assumed by ($\mathbb{A}, \mathbb{B}, \mathbb{C}$, μ, Q) depict disturbance patterns within a given mode, the dynamics of the hidden state r, governed by Π, provides an added degree of freedom for describing a wide array of switching behavior.

3.2.3.1 *Intermittent drifts*

For one-dimensional intermittent drifts (Figure 1(a)), one sets $\mathbb{A} = \mathbb{B} = \mathbb{C} = 1$:

$$\begin{aligned} \gamma_{t+1} &= \gamma_t + \varphi_{t+1} \\ \theta_t &= \gamma_t + e_t \\ \pi_{ij} &< 1 \\ \pi_{ij} &\approx 1, \text{ if } i = j \\ i, j &\in \{1\,(\text{`white noise'}), 2\,(\text{`random walk'})\} \end{aligned} \tag{17}$$

Here, φ_t and e_t are uncorrelated, zero-mean Gaussian signals with the covariances of $Q^{\varphi}_{r_t}$ and $Q^{e}_{r_t}$, respectively. When $r_t = 1$ (*i.e.*, the white noise disturbance regime), $Q^{\varphi}_{r_t} \approx 0$ and $Q^{e}_{r_t} \gg 0$. Random-walk type behavior occurs when the hidden state switches to $r_t = 2$, where $Q^{\varphi}_{r_t} \gg 0$, and $Q^{e}_{r_t} \approx 0$. Since it is common that there is low probability of switching once the system enters a particular regime, a diagonally-dominant Π is employed. With some abuse of notation, the dependency of the noise statistics on the hidden state is denoted as such:

$$\begin{aligned} \gamma_{t+1} &= \gamma_t + \varphi_{r_{t+1}} \\ \theta_t &= \gamma_t + e_{r_t} \end{aligned} \tag{18}$$

One can certainly incorporate additional behavior (e.g., linear drifts) into the model by introducing more hidden states and associating them with appropriate stochastic models to those states. Multi-dimensional intermittent drifts (Figure 1(b)) can be described likewise by adding additional dimensions to θ. Correlations between the different channels can be introduced through the addition of an extra number of hidden states (see Section 3.4 for details).

3.2.3.2 *Abrupt step change, mean-shifts and outliers*

Many systems suffer from infrequent but abrupt changes [101, 5]. By changing the transition probability matrix described in Section 3.2.3.1 to have much higher weight on the first (say) column, a signal typified by Figure 1(c) may be effectively modeled since self-transitions of

the 'jump' state are assumed to be rare. Doing so enables the underlying Markov chain to behave somewhat similarly to a MOG (as described in Section 3.2.2), thus demonstrating the flexibility of the proposed HMM methodology.

By setting $\mathbb{C} = 0$, assigning appropriate values to $(\mu^e_{r_t}, Q^e_{r_t})$ and assuming a diagonally dominant Π, infrequent jumps between various mean levels (Figure 1(d)) is described. Outliers may be suitably modeled by assuming a proper form of Π, *i.e.*, $\pi_{ij} \approx 1$, $\forall i$ for some arbitrary column j.

3.2.3.3 Intermittent oscillations and other disturbance signals

Oscillations may be described as an appropriately modeled linear filter parameterized by $\mathbb{A}, \mathbb{B}, \mathbb{C}$ (see [3]), driven by an external forcing function. Therefore, intermittent oscillations come about by allowing the dynamics of the exogenous forcing function to change according to r.

The use of multiple Markov chains and superposition affords additional flexibility in terms of describing realistic disturbances patterns. Therefore, situations such as the appearance of intermittent drifts in addition to infrequent outliers are within the realm of the proposed HMM framework.

3.3 System identification & state estimation of jump systems

Typically, model-based advanced process control solutions operate by applying a policy (e.g. an open loop optimization in the context of Model Predictive Control) to on-line state estimates, by virtue of the certainty equivalence principle. For this reason, a brief mention of system identification of the overall system (Eq. (19) characterizes a general Markov Jump System (MJS)) composed of a concatenation of Eqs. (13), (14), (16), as well as practical considerations due to the difficulties associated with exact identification, is appropriate.

This is followed by an explanation of the corresponding state estimation problem.

$$\begin{aligned}
\begin{bmatrix} x_{t+1} \\ \gamma_{t+1} \end{bmatrix} &= \mathcal{F}_{r_{t+1}}\Big(\begin{bmatrix} x_t \\ \gamma_t \end{bmatrix}, u_t, \xi_{r_{t+1}} \Big) \\
y_t &= \mathcal{G}_{r_t}\Big(\begin{bmatrix} x_t \\ \gamma_t \end{bmatrix}, n_{r_t} \Big) \\
pr(r_t = j | r_{t-1} = i) &= \pi_{ij} \qquad (19)
\end{aligned}$$

Here $\mathcal{F}$ is implicitly understood to include model structures and parameters from $\{f, \mathbb{A}_r, \mathbb{B}_r, \mathbb{C}_r\}$ and the hidden Markov chain. A similar remark is extended to $\mathcal{G}$. Besides $\mathcal{F}$ and $\mathcal{G}$, the statistics of the noise ξ (a concatenation[4] of (ω, φ, e)) and n (a concatenation of (v, e)) can depend on r.

3.3.1 System identification & practical considerations

Identification of MJS's, an area of interest in its own right, is particularly challenging due to the influence of the unknown Markov states. For example, exact Expectation-Maximization (EM) methods, typically used for parameter estimation in the context of Maximum Likelihood Estimation (MLE), are computationally intractable due to the exponentially growing number of Hidden state trajectories. Murphy [64], Pavlovic et al. [69] and Wong and Lee [105] have employed approximate EM techniques to learn the parameters of Markov Jump Linear systems (MJLS). Vidal et al. [103] proposed a method based on generalized Principal Component Analysis (PCA) for learning MJLS's that are unperturbed by disturbances. The author(s) of [100] focused solely on estimating the transition probability matrix (Π) for Markov jump linear systems.

The case of system identification for nonlinear (jump) systems is still an open area of research. This, as well as the difficulties associated with the identification of Markov jump linear systems (such as convergence to a local optima on the likelihood landscape), necessitates other practical considerations, as described next.

[4]Depending on the structure of the problem, such a concatenation may be an involved process. Nonetheless, this has been done to achieve notational convenience and uniformity.

3.3.1.1 Practical considerations

For the purpose of this chapter, it is assumed that the system, noise and Markov parameters are available, either through identification under strict control and/ or fundamental knowledge.

In ascertaining the impact of the uncertainty vector $\theta \in \mathbb{R}^{n_\theta}$, one would need to determine all disturbance scenarios of interest (for example one may suspect intermittent drifts affecting the input channels, abrupt jumps in the output channels and so on). The disturbance modeling would then proceed as mentioned before in Section 3.2.

Of special note is the fact that learning the exact value of the elements of Π is not required. Indeed, examples have shown results to be robust towards the tuning of Π (see, for example, Section 3.5.1.3).

Nonetheless, simple guiding rules may be derived from a straightforward application of probability [90]. Specifically, the elements of Π may be approximated as such:

$$\pi_{ii} \approx 1 - \frac{1}{l_i}$$

where l_i is the expected duration (in number of sample time units) spent in regime i until a transition occurs. For instance, in the case of a system experiencing abrupt jumps (Section 3.2.3.2), the expected duration in the jump state may be expected to be at most a fraction longer than a single sample time unit, as can be easily ascertained by a control practitioner with sufficient process insights. If more rigorous methods for approximating Π are desired, a Hidden Markov Model may be learnt and its transition probability matrix used in the existing context. However, the output of this Hidden Markov Model needs to be redefined to be the time-varying parameters (obtained through adaptive system identification) of the plant, which is being subject to the pre-supposed disturbance patterns. Readers are referred to [90] for a treatment on this issue.

3.3.2 State estimation

For the Markov jump systems this work is concerned with, the optimal filter involves an exponentially growing number of linear filters due to the dependence of the noise statistics

on r_t. Without knowledge of the sequence $(r_0, r_1, \ldots, r_t)$, the optimal filter needs to average over all possible past trajectories, the number of which scales as J^t, where J is the cardinality of $\mathcal{J}$, the set containing all possible realizations of r. On the other hand, if the trajectory were known, the estimation problem specializes to a sequence of predictor-corrector steps, the cornerstone of linear and non-linear filtering theory.

A popular sub-optimal filtering technique, the second order Generalized Pseudo-Bayesian (GPBn) algorithm developed by [4] is employed throughout this chapter for the reasons mentioned in [79] and [100]. In particular, GPB2 will be used. Details of the 'Interacting Multiple Model' (IMM) algorithm [9, 34], an approach with similar performance as GPB2 but at the (lower) computational cost of GPB1 are omitted since the focus of this work is not on state estimation for jump systems. More involved maximum likelihood-based methods that are not considered in this chapter can be found in [55] and the references therein.

Define η as the concatenated state $[x; \gamma]$, $\mathcal{R}_{t-1}^{t} \triangleq (r_{t-1}, r_t)$ as a sequence of the two most recent Markov-state realizations and Y_0^t, the measurement sequence $(y_0, \ldots, y_t)$. Note from Eq. (19) that r_t is assumed to affect x_t and y_t immediately.

Accordingly, let $\hat{\eta}_{t|t}(\mathcal{R}_{t-1}^{t})$ denote the estimate of η_t that is conditioned on the two most recent latent-state realizations, as well as Y_0^t. Similarly, the corresponding estimation error covariance matrix is represented as $P_{t|t}(\mathcal{R}_{t-1}^{t})$. The main idea is to have these quantities with trajectories whose last two terms differ, be combined (via moment-matching) into a single Gaussian, parameterized by $\{\hat{\eta}_{t|t}, P_{t|t}\}$[5]. A recursive scheme, characterized by two phases: 'branching' (Eq. (20)) and 'merging' (Eqs. (21), (22)) are briefly outlined and illustrated in Figure 2. A simple numerical example can be found in Section 3.3.2.1. Starting with $\left\{\hat{\eta}_{t-1|t-1}(r_{t-1}),\, P_{t-1|t-1}(r_{t-1})\right\}_{r=1}^{J}$, updates are performed to obtain $\left\{\hat{\eta}_{t|t}(\mathcal{R}_{t-1}^{t}),\, P_{t|t}(\mathcal{R}_{t-1}^{t})\right\}$ through so-called 'branching' operations:

Branching:

$$\{\hat{\eta}_{t|t}(\mathcal{R}_{t-1}^{t}), P_{t|t}(\mathcal{R}_{t-1}^{t})\} \;=\; \texttt{filter}\big[\hat{\eta}_{t-1|t-1}(r_{t-1}),\, P_{t-1|t-1}(r_{t-1}),\, y_t\big] \tag{20}$$

[5] In fact, the probability density function more closely resembles a mixture-of-Gaussians

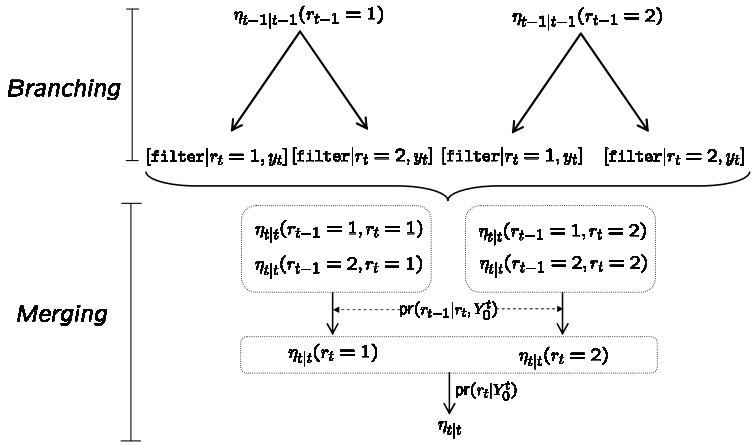


Figure 2: Schematic illustrating the GPB2 algorithm for a two-regime problem.

Here, each application of `filter` refers to prediction and measurement update steps, both of which correspond to the state estimator of choice matched to the appropriate regime. For brevity, the input term, u, has been omitted.

These J^2 pairs of estimates and covariance matrices are then successively merged into terms accounting for shorter sequences whereupon the quantities $\{\hat{\eta}_{t|t}(r_t),\, P_{t|t}(r_t)\}$ are finally obtained. The latter serve to initialize the 'branching' operations at the next time step. The following equations (Eqs. (21), (22)) represent the 'merging' phase:

Merging:

$$\begin{aligned}
\hat{\eta}_{t|t}(r_t) &= \sum_{r_{t-1}\in\mathcal{J}} \hat{\eta}_{t|t}(\mathcal{R}_{t-1}^t)\mathrm{pr}(r_{t-1}|r_t,Y_0^t) \\
P_{t|t}(r_t) &= \sum_{r_{t-1}\in\mathcal{J}} [\left\{\hat{\eta}_{t|t}(\mathcal{R}_{t-1}^t) - \hat{\eta}_{t|t}(r_t)\right\}\{\cdot\}' \\
&\quad + P_{t|t}(\mathcal{R}_{t-1}^t)] \cdot \mathrm{pr}(r_{t-1}|r_t,Y_0^t) \\
\mathrm{pr}(r_{t-1}|r_t,Y_0^t) &= \frac{1}{c_1}\cdot \mathrm{pr}(y_t|\mathcal{R}_{t-1}^t,Y_0^{t-1})\mathrm{pr}(r_t|r_{t-1})\cdot \mathrm{pr}(r_{t-1}|Y_0^{t-1})
\end{aligned} \tag{21}$$

$$\begin{aligned}
\hat{\eta}_{t|t} &= \sum_{r_t\in\mathcal{J}} \hat{\eta}_{t|t}(r_t)\mathrm{pr}(r_t|Y_0^t) \\
P_{t|t} &= \sum_{r_t\in\mathcal{J}} [\left\{\hat{\eta}_{t|t}(r_t) - \hat{\eta}_{t|t}\right\}\{\cdot\}' + P_{t|t}(r_t)]\cdot \mathrm{pr}(r_t|Y_0^t) \\
\mathrm{pr}(r_t|Y_0^t) &= \frac{1}{c_2}\cdot \sum_{r_{t-1}\in\mathcal{J}} \mathrm{pr}(y_t|\mathcal{R}_{t-1}^t,Y_0^{t-1})\mathrm{pr}(r_t|r_{t-1})\cdot \mathrm{pr}(r_{t-1}|Y_0^{t-1})
\end{aligned} \tag{22}$$

Here, c_1 refers to a constant ensuring that $\mathrm{pr}(r_{t-1}|r_t, Y_0^t)$ sums to unity. The first two lines of Eq. (21) are a consequence of the property of the conditional expectation and variance operators[6]. The third line comes directly from an application of Bayes' rule (generating the term $\mathrm{pr}(y_t|\mathcal{R}_{t-1}^t, Y_0^{t-1})$) as well as the Markov property and the definition of conditional probability (hence the terms $\mathrm{pr}(r_t|r_{t-1})$, $\mathrm{pr}(r_{t-1}|Y_0^{t-1})$). $\mathrm{pr}(y_t|\mathcal{R}_{t-1}^t, Y_0^{t-1})$ is related to the innovations term associated with the `filter` operation employed at the branching stage.

To obtain point estimates, a final merging step (Eq. (22)) is required, where c_2 is another normalizing constant. If required, a prediction and/ or filtered estimate of r_{t+1}, can be obtained viz:

$$\begin{aligned} \hat{r}_{t+1|t} &= \arg\max_{r_{t+1}} \left\{ \mathrm{pr}(r_{t+1}|Y_0^t) \triangleq \sum_{r_t} \mathrm{pr}(r_{t+1}|r_t) \cdot \mathrm{pr}(r_t|Y_0^t) \right\} \\ \hat{r}_{t|t} &= \arg\max_{r_t} \left\{ \mathrm{pr}(r_t|Y_0^t) \right\} \end{aligned} \tag{23}$$

A more sophisticated technique that is not considered in this chapter, involves dynamic programming [55] to compute the sequence with the highest probability.

3.3.2.1 Numerical Illustration of GPB2

Consider the following Markov jump linear system used for the purpose of simulation.

$$\begin{aligned} x_t &= x_{t-1} + \varphi_{r_t} \\ y_t &= x_t + v_t \\ \mathbb{E}[\varphi_{r=1}\varphi_{r=1}'] &= 10^{-3} \\ \mathbb{E}[\varphi_{r=2}\varphi_{r=2}'] &= 20 \\ \pi_{11} &= \pi_{22} = 0.99 \end{aligned} \tag{24}$$

During the simulations, measurement noise samples are generated using $R \triangleq \mathbb{E}[v_t v_t']$ set to 5. The latter quantity is available for estimator design. x_0 is set to 110 and available for estimator initialization. With this, regime 1 corresponds to a quiescent phase, and the other regime, a mode with significant velocity (due to changes in φ).

[6]Consider two uni-dimensional random variables V_1, V_2. Then, $\mathbb{E}_{V_1}[V_1] = \mathbb{E}_{V_2}\mathbb{E}_{(V_1|V_2)}[V_1|V_2] = \sum_{V_2} \mathbb{E}_{(V_1|V_2)}[V_1|V_2] \cdot \mathrm{pr}(V_2)$. Similarly, $\mathrm{Var}_{V_1}[V_1] = \mathbb{E}_{V_2}[\mathrm{Var}_{(V_1|V_2)}[V_1|V_2]] + \mathrm{Var}_{V_2}[\mathbb{E}_{(V_1|V_2)}[V_1|V_2]] = \sum_{V_2} \left[\mathrm{Var}_{(V_1|V_2)}[V_1|V_2] + (\mathbb{E}_{(V_1|V_2)}[V_1|V_2] - \mathbb{E}_{V_1}[V_1])^2\right] \cdot \mathrm{pr}(V_2)$.

For time-index $(t-1)$, given $\left\{\hat{x}_{t-1|t-1}(r_{t-1}), P_{t-1|t-1}(r_{t-1})\right\}$, the following are the time (Eq. (25)) and measurement update (Eq. (26)) steps utilized within the 'branching' phase of the GPB2 methodology:

Branching:

$$\begin{aligned}
\hat{x}_{t|t-1}(\mathcal{R}_{t-1}^{t}) &= \hat{x}_{t-1|t-1}(r_{t-1}) \\
P_{t|t-1}(\mathcal{R}_{t-1}^{t}) &= P_{t-1|t-1}(r_{t-1}) + \mathbb{E}[\varphi_{r_t}\varphi_{r_t}'] \qquad (25)
\end{aligned}$$

$$\begin{aligned}
\hat{x}_{t|t}(\mathcal{R}_{t-1}^{t}) &= \hat{x}_{t|t-1}(\mathcal{R}_{t-1}^{t}) + K_t(\mathcal{R}_{t-1}^{t})(y_t - \hat{x}_{t|t-1}(\mathcal{R}_{t-1}^{t})) \\
K_t(\mathcal{R}_{t-1}^{t}) &= P_{t|t-1}(\mathcal{R}_{t-1}^{t})(P_{t|t-1}(\mathcal{R}_{t-1}^{t}) + R)^{-1} \\
P_{t|t}(\mathcal{R}_{t-1}^{t}) &= (I - K_t(\mathcal{R}_{t-1}^{t}))P_{t|t-1}(\mathcal{R}_{t-1}^{t}) \qquad (26)
\end{aligned}$$

With the quantities $\left\{\hat{x}_{t|t}(\mathcal{R}_{t-1}^{t}),\, P_{t|t}(\mathcal{R}_{t-1}^{t})\right\}$, merging is done via Eqs. (21), (22), such that a point estimate is ultimately obtained:

Merging:

$$\begin{aligned}
\hat{x}_{t|t}(r_t) &= \sum_{r_{t-1}} \hat{x}_{t|t}(\mathcal{R}_{t-1}^{t})\mathrm{pr}(r_{t-1}|r_t, Y_0^t) \\
P_{t|t}(r_t) &= \sum_{r_{t-1}} [\left\{\hat{x}_{t|t}(\mathcal{R}_{t-1}^{t}) - \hat{x}_{t|t}(r_t)\right\}\{\cdot\}' + P_{t|t}(\mathcal{R}_{t-1}^{t})] \cdot \mathrm{pr}(r_{t-1}|r_t, Y_0^t) \\
\hat{x}_{t|t} &= \sum_{r_t} \hat{x}_{t|t}(r_t)\mathrm{pr}(r_t|Y_0^t) \\
P_{t|t} &= \sum_{r_t} [\left\{\hat{x}_{t|t}(r_t) - \hat{x}_{t|t}\right\}\{\cdot\}' + P_{t|t}(r_t)] \cdot \mathrm{pr}(r_t|Y_0^t)
\end{aligned}$$

State estimation results for a typical realization are depicted in Figure 3, where the estimates corresponding to two different steady-state Kalman filters are presented for benchmarking purposes. The first low-gain filter is tuned assuming a state noise variance of 10^{-3}. Conversely, the second high-gain filter assumes a state noise variance of 20. Since neither steady-state filter works well across both regimes, significant tuning efforts are required if a stationary Kalman filter is to be employed. Having established the means by which realistic disturbances may be modeled using a HMM framework and the state estimation mechanism

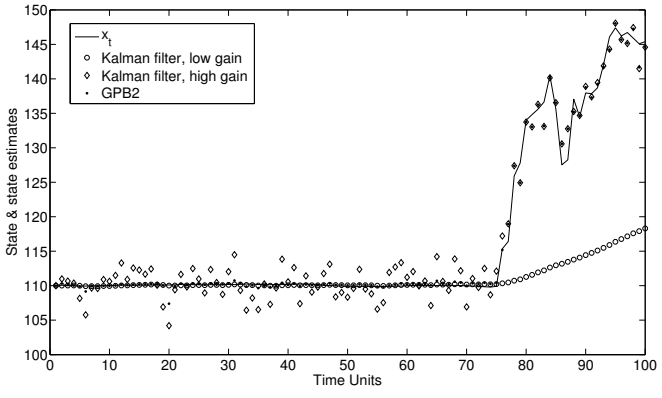


Figure 3: Comparing state estimators reacting to a regime change.

(for the eventual purpose of monitoring and control), the following sections examine the usefulness of the proposed work in a variety of process control-relevant situations.

3.4 Example 1: HMM disturbance framework for offset-free linear model predictive control

Supported by constraint-handling capabilities and a well-established body of theory, Linear Model Predictive Control (LMPC) is the preferred advanced process control method employed in process industries. For offset-free control, augmentation with integrators is common [65]:

$$\begin{pmatrix} x_{t+1} \\ d_{t+1} \\ p_{t+1} \end{pmatrix} = \begin{pmatrix} A & G_d & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} x_t \\ d_t \\ p_t \end{pmatrix} + \begin{pmatrix} B \\ 0 \\ 0 \end{pmatrix} u_t$$

$$y_t = \begin{pmatrix} C & 0 & G_p \end{pmatrix} \begin{pmatrix} x_t \\ d_t \\ p_t \end{pmatrix} \tag{27}$$

$d_t \in \mathbb{R}^{n_d}$ and $p_t \in \mathbb{R}^{n_p}$ are interpreted as state/ input and output disturbances respectively. Furthermore, $n_d + n_p \leq n_y$ and other constraints on G_d and G_p are required for detectability.

The most popular choice used in earlier versions of MPC (*e.g.*, Dynamic Matrix Control [18]) is $G_p = I_{n_y}, G_d = 0$, which assumes that independent integrated white noise

disturbance enters each output channel. Though popular, this choice can give poor performance for systems with slow poles [87] and/ or integrating modes. Another choice is $G_p = 0, G_d = B$, which assumes independent integrated white noise disturbance enters each input channel.

Regardless, the aforementioned approaches may be limiting in the face of a wrong assumption or switching patterns. Consider Figure 1(b), a time-series plot of disturbances entering the input and output channels of an arbitrary dynamical system. This scenario can be thought of as an approximation to the case where there exists significant intermittent non-stationary disturbance patterns in each channel. Such behavior of switching patterns is oftentimes neglected in disturbance modeling and controller design due to the lack of a suitable framework. On the other hand, the failure to address this can be to the detriment of the resulting closed-loop behavior, as will be demonstrated in Section 3.4.1.1.

In response to this, researchers have developed techniques to estimate in an on-line fashion, noise parameters required by the Kalman filter. Nevertheless, this 'design' approach [68] implicitly assumes that step disturbances enter either in the input or output channels ($G_d = B$ or $G_p = I$ is typical). In addition, it could require that significant on-line data be collected before the covariance estimates converge and the controller responds to a new pattern.

In contrast, the proposed formulation will be shown to be robust to a wide variety of disturbance scenarios. First, in order to ensure the detectability of the system in the presence of both input and output disturbances ($G_d = B, G_p = I$), the differenced form used by [72] is adopted. Incorporating the proposed HMM framework, it is assumed that the dynamical system of concern evolves according to the following:

$$\begin{aligned} x_{t+1} &= Ax_t + B\theta^d_{t+1} + Bu_t \\ y_t &= Cx_t + \theta^p_t + v_t \qquad (28) \\ \begin{bmatrix} \theta^d \\ \theta^p \end{bmatrix}_{t+1} &= \begin{bmatrix} \theta^d \\ \theta^p \end{bmatrix}_t + \begin{bmatrix} \varphi^d \\ \varphi^p \end{bmatrix}_{r_{t+1}} \qquad (29) \end{aligned}$$

where as before, θ^d_t and θ^p_t are the input and output disturbances respectively. Eq. (29)

is a specialization of Eq. (16). These integrators are driven by zero-mean, uncorrelated, white Gaussian signals $\varphi_r^d \sim \mathcal{N}(0, Q_r^{\varphi^d})$, and $\varphi_r^p \sim \mathcal{N}(0, Q_r^{\varphi^p})$. $v_t \sim \mathcal{N}(0, R)$ is normally distributed measurement noise.

Having established this, and provided that the original system of concern is detectable, a detectable differenced formulation Eq. (30) that will be used by the receding-horizon regulator as well as the (GPB2) state-estimator is:

$$\begin{aligned}
\begin{pmatrix} \Delta x_{t+1} \\ z_{t+1} \end{pmatrix} &= \underbrace{\begin{pmatrix} A & 0 \\ CA & I \end{pmatrix}}_{\tilde{A}} \begin{pmatrix} \Delta x_t \\ z_t \end{pmatrix} + \underbrace{\begin{pmatrix} B \\ CB \end{pmatrix}}_{\tilde{B}} \left(\Delta u_t + \varphi^d_{r_{t+1}} \right) + \begin{pmatrix} 0 \\ \varphi^p_{r_{t+1}} \end{pmatrix} \\
&= \tilde{A}\eta_t + \tilde{B}\Delta u_t + \underbrace{\tilde{B}\varphi^d_{r_{t+1}} + \begin{pmatrix} 0 \\ I \end{pmatrix} \varphi^p_{r_{t+1}}}_{\xi_{r_{t+1}}} \\
y_t &= \underbrace{\begin{pmatrix} 0 & I \end{pmatrix}}_{\tilde{C}} \begin{pmatrix} \Delta x_t \\ z_t \end{pmatrix} + y^* + v_t
\end{aligned} \tag{30}$$

Here, $z_t \triangleq (y_t - v_t - y^*)$, with y^* denoting the desired output setpoint. Also, Δ represents a time-differencing operator. $\xi_{r_{t+1}}$, the covariance of which depends on the hidden state r, represents the effective state noise. $\varphi^d_{r_{t+1}}$, and $\varphi^p_{r_{t+1}}$ refer to zero-mean white Gaussian signals that drive the input and output integrators, respectively. Since it is assumed that $\varphi^d_{r_{t+1}}$, and $\varphi^p_{r_{t+1}}$ are uncorrelated, then one has Eq. (31).

$$\mathbb{E}[\xi_{r_{t+1}} \xi'_{r_{t+1}}] \;=\; \tilde{B}\mathbb{E}[\varphi^d_{r_{t+1}} \varphi^{d'}_{r_{t+1}}]\tilde{B}' + \begin{pmatrix} 0 & 0 \\ 0 & \mathbb{E}[\varphi^p_{r_{t+1}} \varphi^{p'}_{r_{t+1}}] \end{pmatrix} \tag{31}$$

The benefits of using the differenced formulation is that detectability is ensured under rather mild conditions. This is favorable as compared to approaches based on Eq. (27), where certain choices like $G_d = B$, and $G_p = I$ [65] make the system undetectable. As mentioned earlier, the GPB2 filter will be used for state estimation.

3.4.1 Numerical example

3.4.1.1 System parameters

For simplicity, consider the triple $(A = 0.9, B = 1, C = 1.5)$ parameterizing a nominal Single-Input-Single-Output (SISO) system. Generally speaking, it is not certain in advance if input or output disturbances will dominate. Furthermore, there may exist probabilistic switches between regimes, as postulated in this work. To evaluate the performance of the proposed controller under these possible situations, four simulation scenarios are considered. As particular instances of Eqs. (28), (29), these (and their simulation parameters) correspond to

- I. Levels of input noise << levels of output noise. This can be thought of the case where $\theta_t^d = 0, \forall t$. $\mathbb{E}[\varphi^d \varphi^{d\prime}]$ and $\mathbb{E}[\varphi^p \varphi^{p\prime}]$ correspond to the 'low:high' regime as reported in Table 1.

- II. Levels of input noise >> levels of output noise. Similarly, in this case, $\theta_t^p = 0,\ \forall t$ and the noise statistics correspond to the 'high:low' regime.

- III. Levels of input noise are comparable to levels of output noise. Here, the noise parameters are the same as those in 'high:high' regime of Table 1.

- IV. Relative levels of input and output noise switch in a probabilistic manner (see Figure 1(b)). The transition probability matrix Π, used in the simulated studies, is given in Eq. (32) and reflects the situation where either input or output (but not both) disturbances dominate. In accordance with intuition, it can also be seen that relative to regimes 2 and 3, the system tends to spend less time, on average, in the 'high-high' and 'low-low' regimes. For instance, the expected duration the system spends in regime 1 or 4 is $\frac{1}{1-0.8} = 5$ time units whereas that spent in regime 2 or 3 is $\frac{1}{1-0.97} \approx 33$ units. Furthermore, drastic 'low-low' to 'high-high' transitions (and vice versa) are forbidden. The noise parameters used for simulation can be found in Table

1.

$$\Pi = \begin{pmatrix} 0.800 & 0.100 & 0.100 & 0.000 \\ 0.010 & 0.970 & 0.010 & 0.010 \\ 0.010 & 0.010 & 0.970 & 0.010 \\ 0.000 & 0.100 & 0.100 & 0.800 \end{pmatrix} \tag{32}$$

Table 1: Noise parameters used in simulation studies

Input Noise: Output Noise	Regime $r \in \{1,2,3,4\}$	$Q_r^{\varphi^d}$	$Q_r^{\varphi^p}$
'Low:Low'	$r=1$	10^{-10}	10^{-10}
'Low:High'	$r=2$	10^{-10}	50
'High:Low'	$r=3$	10	10^{-10}
'High:High'	$r=4$	10	50

For simplicity, all simulations were run with negligible measurement noise, that is $R \approx 0$.

3.4.1.2 Parameters Used for Controller/ Estimator Design

In order to investigate the effect of disturbance model vs. plant simulation mismatch, four model-predictive controller/ estimator pairs were constructed, all based on the proposed differenced model form of Eq. (30). These differ only in the estimators employed. Namely, these controller/ estimators assume:

- I. Output disturbance only. In this case, $\mathbb{E}[\xi\xi']$ can be computed using data from Table 1 corresponding to $r = 2$, and Eq. (31). As noted by [66], the resulting steady-state Kalman gain is $\begin{bmatrix} 0 & I \end{bmatrix}$ parameterizing an open-loop observer for Δx and a deadbeat observer for z.

- II. Input disturbance only. Similarly one uses $r = 3$ data from Table 1, for computing $\mathbb{E}[\xi\xi']$. A deadbeat observer is obtained in this case.

- III.Output and input disturbances. Here, one uses $r = 4$ data from Table 1, for computing $\mathbb{E}[\xi\xi']$.

- IV. Switching behavior. For this case, a GPB2 estimator with Π given in Eq. (32), is employed.

Although the velocity form is employed, controllers / estimators I, II and III can be regarded as special cases of Eq. (27), and as such are considered as standard MPC formulations for imparting the integral action. For all the (simulation scenario-controller) pairs considered, the objective function, with $y^* = 0$ is as follows (Eq. (33)):

$$\min \frac{1}{N} \sum_{t=0}^{N} \left[z_t' z_t + 0 \cdot \Delta u_t' \Delta u_t \right]_{N \to \infty} \tag{33}$$

Constraints have been removed for clarity of exposition and the absence of a penalty on excessive actuator movement in the objective function results in a deadbeat controller (Eq. (34)):

$$\Delta u_t = - \begin{pmatrix} 0.90 & 0.67 \end{pmatrix} \begin{pmatrix} \widehat{\Delta x} \\ \hat{z} \end{pmatrix}_{t|t} \tag{34}$$

3.4.1.3 Results & Discussion

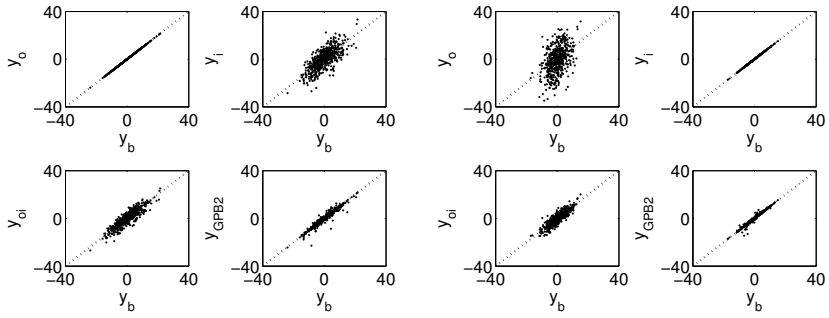
For each simulation scenario I-IV, 500 realizations, each of duration 500 sample time units, were run. Furthermore, for each realization of each simulation scenario, a controller coupled with a time-varying Kalman filter assuming perfect knowledge of the true simulation regime was tested. Although this assumption is impractical, such a controller is tested for benchmarking purposes. For example, in scenario IV corresponding to switching disturbances, such a time-varying Kalman filter would have had access to the actual Markov regime. As mentioned in Section 3.3.2, such ideality is in contrast to the GPB2 estimator, which estimates the regime sequence based on on-line measurements and relies on moment matching to bound the computational cost. The performance index $\mathcal{P}$[7] is the average ratio of squared-tracking error for each controller to that corresponding to the benchmarking controller/ estimator. Closed-loop results are reported in Table 2. Figure 4 contains

[7] $\mathcal{P} \triangleq \frac{1}{S} \sum_{s=1}^{S} \left(\frac{\sum_{t=1}^{T} y_t^2}{\sum_{t=1}^{T} (y_b)_t^2} \right)$; y_b refers to the output corresponding to the benchmark

Table 2: Mean of normalized squared error over 500 realizations

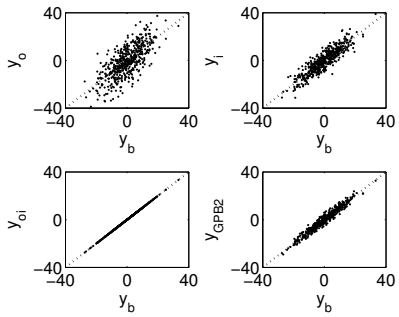
Simulation Scenario	Controller/ Estimator: Output (y_o)	Input (y_i)	Output&Input (y_{oi})	GPB2 (y_{GPB2})
Output (I)	1.00	1.62	1.21	1.03
Input (II)	5.15	1.00	1.33	1.02
Output&Input (III)	1.84	1.24	1.00	1.05
Switching (IV)	3.21	1.52	1.22	1.08

benchmarked plots of y (for a typical realization) corresponding to various combinations of controller/ estimator schemes and simulation scenarios. As can be seen from Table 2 and

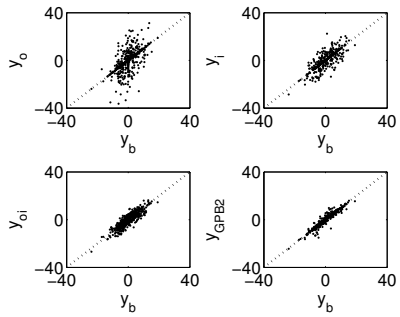


(a) Output disturbance.

(b) Input disturbance.



(c) Output and input disturbance.



(d) Switching disturbances.

Figure 4: Benchmarked output plots for various combinations of employed controller/ estimator schemes and simulation scenarios.

Figs. 4(a), 4(b), 4(c) and 4(d), the proposed model predictive controller designed based on the GPB2 estimator of the HMM model yields the best performance amongst all the controller/ estimators other than that which coincides with the actual simulation scenario. This suggests that the proposed formulation is generally more robust than the standard

controllers employed. Due to the relatively large time constant of the system ($A = 0.9$), the output estimator/ regulator (y_o) expectedly [87] gives the poorest performance for the scenarios it has not been designed for.

Furthermore, using a formulation that includes both input and output disturbances often means using an observer gain that 'averages' the input and output disturbance effects. How this translates to final closed-loop performance is not clear. Using the time-varying GPB2 estimator results in a dynamic observer gain that is a function of on-line measurements. Despite the mismatch in terms of Π, the control performance is still acceptable.

In the next section, the proposed HMM framework, although designed to handle stochastic disturbance signals, is shown to be flexible enough to handle deterministic disturbance scenarios.

3.5 Example 2: Rejecting & detecting deterministic step changes

Deterministic step changes constitute a simple but important class of disturbance signals. In this section, it will be demonstrated that using the proposed noise model results in superior closed loop performance as compared to the popular Integrated White Noise (IWN) assumption, for the purpose of rejecting such perturbations.

Consider the following simulation where a persistent step change of constant magnitude κ, is introduced to the output channel at (an arbitrarily selected) time t^*.

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t + \theta_t^p + v_t \\ \theta_t^p &= \begin{cases} 0, \, t < t^* \\ \kappa, \, t \geq t^* \end{cases} \end{aligned} \tag{35}$$

In order to model this behavior within the context of the HMM framework, Eqs. (28), (29), and (30) are adjusted by ignoring parts related to θ^d. Then, Π is set such that the hidden Markov chain spends the majority of its time in regime 1 and transitions to regime 2 are swiftly followed by switches back to the 1st mode. The variance of $\varphi_{r_t}^p$ for regime 1 is set approximately to the nil matrix, whereas the (non-zero) elements of the covariance matrix for regime 2 may be interpreted as user-defined tuning parameters. These assumptions

suggest a disturbance model that anticipates infrequent jumps (see Figure 1(c)) and is considered a close approximation to an abrupt step change of constant magnitude. In particular, the following disturbance model is obtained:

$$\begin{aligned}
\begin{pmatrix} \Delta x_{t+1} \\ z_{t+1} \end{pmatrix} &= \tilde{A}\eta_t + \tilde{B}\Delta u_t + \begin{pmatrix} 0 \\ I \end{pmatrix} \varphi^p_{r_{t+1}} \\
y_t &= \tilde{C} \begin{pmatrix} \Delta x_t \\ z_t \end{pmatrix} + y^* + v_t \\
\pi_{11} &= \pi_{21} >> \pi_{12} = \pi_{22} \\
\mathbb{E}[\varphi^p_{r=1}\varphi^p_{r=1}] &= diag(q_{1,1}, \ldots, q_{1,n_y}),\ q_{1,l} << 1,\ 1 \le l \le n_y \\
\mathbb{E}[\varphi^p_{r=2}\varphi^p_{r=2}] &= diag(q_{2,1}, \ldots, q_{2,n_y}),\ q_{2,l} >> 1,\ 1 \le l \le n_y \\
\mathbb{E}[vv'] &= diag(r_1, \ldots, r_{n_y})
\end{aligned} \tag{36}$$

The definitions of $\tilde{A}, \tilde{B}, \tilde{C}$, z and y^* are found in Section 3.4. The estimator corresponding to the HMM model of (Eq. (36)) is expected to be able to detect the jump events in the plant (Eq. (35)), thereby allowing the regulator to reject the step changes without over-reacting to measurement noises.

3.5.1 Comparing closed-loop performance: different disturbance models

As an illustration, consider a SISO system with $A = 0.9$, and $B = 1$, $C = 3$. For the simulations, T, the length of each run, is set to 200 time steps, κ to 20, t^* to 175, and the variance of the measurement noise, R to 5. Like the previous example, the (unconstrained) control objective is the minimization of the following infinite-horizon sum, with a set-point at the origin:

$$\min \left\{ \sum_{t=0}^{\infty} z_t' z_t + 0.1 \Delta u_t' \Delta u_t \right\} \tag{37}$$

The matrix parameterizing the resulting linear quadratic regulator is obtained by solving a discrete-time Riccati equation so that one gets Eq. (38)

$$\Delta u_t = - \begin{pmatrix} 0.89 & 0.32 \end{pmatrix} \begin{pmatrix} \widehat{\Delta x} \\ \hat{z} \end{pmatrix}_{t|t} \tag{38}$$

3.5.1.1 Output channel Integrated-White-Noise-type disturbance model

IWN disturbance models are commonly employed in process industries. In this context, a disturbance model that assumes IWN on the output channel serves as an appropriate yardstick for comparing closed-loop performance.

The resulting IWN process and disturbance model is represented by adjusting Eq. (36) to remove its dependence on the Markov state, r. Since the plant state x is unperturbed, the resulting (steady-state) Kalman gain (K) has the structure: $K = [0, f]'$, where $f \in [0, 1)$. f depends on the relative ratio of $Q_{\text{iwn}} \triangleq \mathbb{E}[\varphi\varphi']$ to $R \triangleq \mathbb{E}[vv']$. As may be expected, f approaches 0, signifying the absence of output feedback, when either Q_{iwn} is extremely small or the measurement noise covariance exceedingly large, that is, as $Q_{\text{iwn}}/R \to 0$. Conversely, $f \to 1$ as $Q_{\text{iwn}}/R \to \infty$.

Generally speaking, Q_{iwn}[8] is a tuning parameter that indicative of a signal-to-noise tradeoff. Since this is difficult to ascertain *a-priori*, two natural candidates are:

$$Q_{\text{iwn},1} = 0.013 \Leftrightarrow f_1 = 0.05 \tag{39}$$

$$Q_{\text{iwn},2} = 90.25 \Leftrightarrow f_2 = 0.95 \tag{40}$$

For benchmarking purposes, $f^* = 0.5$ corresponds to a well-tuned IWN disturbance model. For ease of references, the state estimators corresponding to these candidates are denoted as IWN_1, IWN_2 and IWN^* respectively.

3.5.1.2 HMM disturbance model: parameters employed

Eq. (36) is specialized as such. Both rows of Π are set to [0.999, 0.001]. Furthermore, $Q^{\varphi}_{r_t=1} \triangleq \mathbb{E}[\varphi_{r=1}\varphi'_{r=1}]$, and $Q^{\varphi}_{r_t=2} \triangleq \mathbb{E}[\varphi_{r=2}\varphi'_{r=2}]$ are set according to Eq. (39) and Eq. (40) respectively.

3.5.1.3 Results & discussion

Closed loop results corresponding to 100 experiments, each of duration $T = 200$ are shown in Table 3. The second column corresponds to the unattainable case where full state feedback is available. Over the course of the closed-loop experiments, it is observed that the

[8] R is assumed to be known

Table 3: Tracking error corresponding to various disturbance models, in response to the introduction of a step disturbance at $t = t^* = 175$. The operator, $< \cdot >$, denotes sample average over 100 realizations.

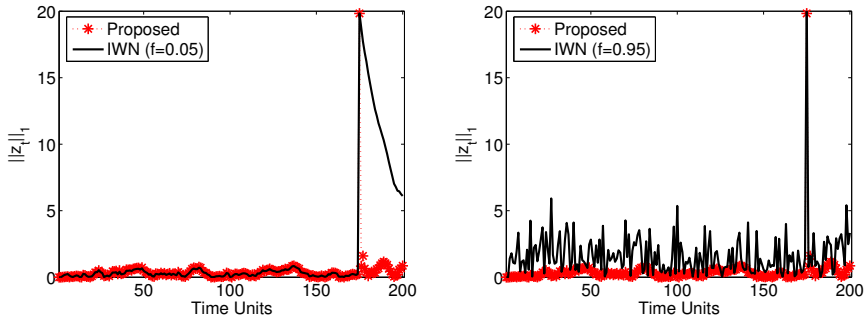
Tracking error for different time phases	Full state feedback	Proposed HMM model	IWN* (tuned)	IWN$_1$ (low gain)	IWN$_2$ (high gain)
$\left\langle \left[\sum_{t=0}^{t^*-1} z_t' z_t \right]^{0.5} \right\rangle$	0.0	4.7	16.8	4.5	27.1
$\left\langle \left[\sum_{t=t^*}^{T} z_t' z_t \right]^{0.5} \right\rangle$	20.0	20.5	24.3	61.7	23.0
$\left\langle \left[\sum_{t=0}^{T} z_t' z_t \right]^{0.5} \right\rangle$	20.0	21.1	29.5	61.8	35.6

proposed framework, giving an increase of less than 6% in the error when compared to the (idealistically assumed) full feedback case, yields better performance than a well-tuned IWN-based estimator, and indeed either IWN$_1$ or IWN$_2$. Being insensitive to output noise signals, the controller based on the low-gain IWN estimator generally performs well before the introduction of the step (first row of Table 3) but not immediately after as it is quite sluggish (Figure 5(a)) in rejecting the step disturbance. The reverse is true for high-gain IWN estimators (see second row of Table 3 and Figure 5(b)) which are comparatively more sensitive to the noise in the feedback signal. A well-tuned IWN estimator, though striking a nice compromise between these two extremes, is nonetheless outperformed significantly by the proposed method, as can be seen from row three of Table 3 and Figure 5(c). The GPB2 estimator that results from the HMM framework is able to detect the absence/ presence of a step change, thereby serving effectively as a time-varying filter. The average, time-varying GPB2 gain mapping y_t to $[\widehat{\Delta x}_{t|t},\, \hat{z}_{t|t}]'$ in the measurement update step may be represented by:

$$\sum_{r_t, r_{t-1}} p(r_t|Y_0^t) p(r_{t-1}|r_t, Y_0^t) K_t(\mathcal{R}_{t-1}^t)$$

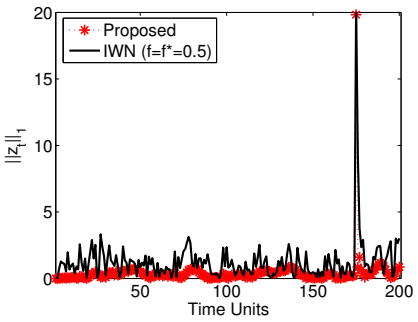
where $K_t(\mathcal{R}_{t-1}^t)$ is the observer gain generated within the GPB2 framework (see Section 3.3.2). Figure 6 shows the trajectory of the only non-trivial, *i.e.,* (2,1), element of the averaged GPB2 gain responding effectively to the output disturbance.

It is noted that tuning the GPB2 estimator is relatively easier than an IWN-based estimator. For handling infrequent steps, π_{11} and π_{12} need only to be close to unity. Furthermore, the values of $Q_{r_t}^{\varphi}$ for $r = 1$ and $r = 2$ are easily set by selecting variances



(a) Proposed model vs. IWN_1.

(b) Proposed model vs. IWN_2.



(c) Proposed model vs. IWN^*.

Figure 5: Comparing closed loop performance of various controller/ estimator pairs: plots of $||z_t||_1$ vs. time for a typical stochastic realization.

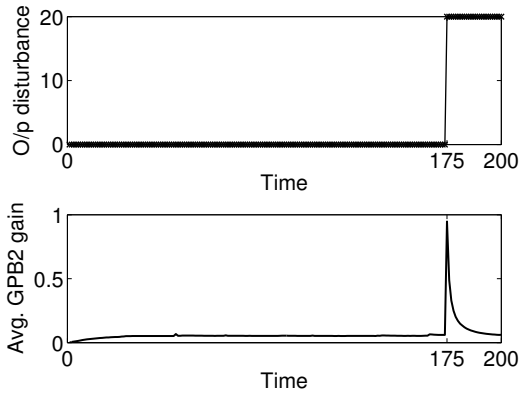


Figure 6: Estimating presence of step disturbance.

Table 4: Re-running experiments in Table 3 with a de-tuned GPB2 estimator.

Tracking error	Full state feedback	HMM model (detuned)	IWN* (tuned)
$\left\langle \left[\sum_{t=0}^{T} z_t' z_t \right]^{0.5} \right\rangle$	20.0	24.0	29.5

that correspond to low and high values of f in the IWN case. To examine robustness, the experiments were repeated with $\pi_{11} = \pi_{12} = 0.90$. From Table 4, it is seen that the proposed model still outperforms a well-tuned IWN estimator.

3.6 Conclusions, limitations & future work

The main contribution of this work is in proposing a Hidden Markov Model framework general enough to describe a large number of realistic but hitherto unquantifiable disturbance scenarios as well as forms of disturbances used in the standard control literature. Examples, ranging from offset-free control in the presence of intermittent input and output disturbances to rejecting deterministic disturbances to control in the presence of highly-varying feed conditions, were used to demonstrate the applicability and relevance of the proposed disturbance model within the framework of popular model-based control solutions.

Admittedly, optimal control policies exist for certain classes of MJS's [17] but are expected to be too limiting or difficult to implement for the purpose of process control. Instead, the authors believe that the benefits of an HMM-based disturbance framework are most immediately reaped by incorporating them within existing control methodologies (such as MPC). This is particularly true when one considers that the significant performance improvement can be achieved at relatively small cost of computation (mostly that associated with the state estimation).

Development of systematic model development / identification methods and more robust control algorithms for Markovian Jump Systems is the focus of our future research.

CHAPTER IV

HYBRID CYBERNETIC MODEL-BASED SIMULATION OF CONTINUOUS PRODUCTION OF LIGNOCELLULOSIC ETHANOL: REJECTING ABRUPTLY CHANGING FEED CONDITIONS

4.1 Introduction

Feed conditions, representing the most immediate avenue via which a plant's performance is affected, are crucial [20]. Issues pertaining to consistency and availability of feed plague a wide spectrum of industries. The pulp and paper, petroleum, fiberglass manufacturing and alternative-fuels industries are such examples. Although details vary from industry to industry, key factors exist that make dealing with feed conditions a strategic priority. From a production standpoint, it is highly desirable to mitigate the impact of highly-varying feed conditions via feedback and/ or feed-forward control, especially for situations where little can be done with regards to the incoming variability. A typical scenario is a downstream unit that is fed by an upstream one, for which the reaction chemistry is poorly understood/ controlled. An interesting and important case in point is ethanol production, the leading substitute for fossil fuels.

Interest in ethanol has grown substantially over the past several decades. As reported by [28] and the references therein, ethanol has already been introduced on a large scale in Brazil, the United States as well as several European countries. Consequently, its role in helping countries attain the goals laid out in long-term, sustainable energy initiatives cannot be overemphasized [102].

At present, (bio)ethanol is predominantly produced from starches (from corn crops) or sugar sources (such as sugar cane). Due to competing agricultural needs for the same resources, such as arable land, as well as the negative downstream greenhouse effects of pursuing such 'first-generation' ethanol solutions, attention has turned to 'second-generation' bioethanol. In particular, producing ethanol from lignocellulosic biomass sources has been

deemed to be the most promising alternative for several reasons. First, lignocellulosic biomass comes predominantly from low cost [26, 57], readily-available and otherwise-discarded agricultural and industrial wastes (such as corn stovers, and wood residues (wood chips, soft woods)). Second, these raw materials do not present a conflict between food and energy production. Furthermore, biomass, being a carbon-neutral source, is environmentally more benign [52].

Converting lignocellulose to ethanol necessitates pre-treatment and then hydrolysis through enzymatic or chemical means (see Figure 4.1). This de-polymerization phase generates six-carbon hexose sugars (*e.g.*, glucose) and five-carbon pentose sugars (such as the comparatively abundant xylose) to be fed to a fermentor (seeded with genetically modified yeast capable of utilizing both substrates), before further post-processing steps are carried out[1]. Although further research and development in all these steps is required (*e.g.*, organ-

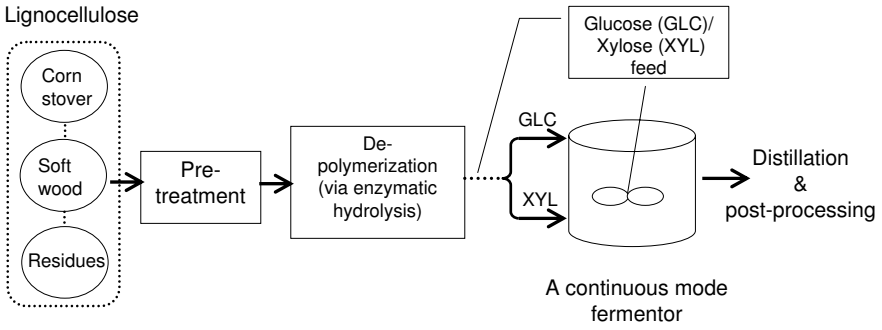


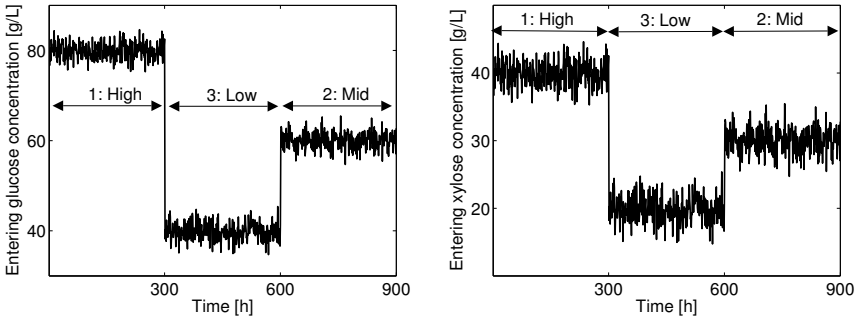
Figure 7: Continuous mode ethanol production (adapted from [57]). Input glucose (GLC) and xylose (XYL) concentrations may experience severe fluctuations due to the diversity of raw materials and/ or changes in intermediate processing steps.

isms capable of more effectively and efficiently converting both (typically preferred) hexoses and pentoses need to be engineered [107]) before lignocellulosic ethanol production is embraced ubiquitously, it is noted that industrial efforts at the plant scale are underway [28]. Furthermore, xylose-fermenting strains of yeast are now reaching levels of performance that approach economically feasible lignocellulosic ethanol production [29]. As such, large scale production of lignocellulosic ethanol will very likely materialize in the near future.

[1]Simultaneous Saccharification and Fermentation (SSF) is not considered in this work.

In anticipation of this, this chapter explores (albeit necessarily through *in-silico* experiments) means of systematically describing and mitigating, through feedback control, the negative impact of highly-varying feed (*i.e.,* glucose and xylose) signals (to the chemostat). The objective is to ensure high throughput whilst respecting, due to economics, high conversion requirements. Given the expectedly differing sources of potential feed sources (which renders lignocellulosic ethanol production attractive in the first place), and the lack of maturity (though not viability) of pre-treatment and hydrolysis technologies, cost-effective mass production of lignocellulosic ethanol may be greatly facilitated by such a control system.

In this work, 'highly-varying feed conditions' are distinguished from the typical (but oftentimes inadequate) description of variations as white noise or filtered white noise. The archetypal example considered, as an approximation to reality, is the case where glucose and xylose compositions jump abruptly between several distinct levels in a probabilistic fashion, as shown in Figure 8. For example, glucose and xylose concentration levels, initially in the 'high' regime, with levels fluctuating about a mean of 80 g/L and 40 g/L respectively, may switch probabilistically (depicted to occur at time = 300 hrs) to the 'low' regime (see Table 5 for the feed conditions postulated in this work). Another random switch occurs at a later time, so that the system transitions to the 'mid' regime. In general, the feed signal remains as an unmeasured disturbance. Estimating this signal is complicated by the presence of the abrupt jumps. Classical disturbance models [53], although widely used in



(a) Glucose concentration vs. time. (b) Xylose concentration vs. time.

Figure 8: Postulated highly varying feed conditions (manifested in entering glucose and xylose concentrations).

process industries, are incapable of simultaneously describing the discrete (due to the regime

Table 5: Entering mean feed concentration for various regimes

	$x_{GLC,in}$ (g/L)	$x_{XYL,in}$(g/L)
1 (High)	80	40
2 (Mid)	60	30
3 (Low)	40	20

jumps associated with feed changes) and continuous dynamics. This is to the detriment of subsequent closed-loop performance. Employing a 'Hidden Markov Model' (HMM) capable of describing such postulated disturbance signals, as recently proposed by [106], to be used within the context of state estimation and closed-loop control, represent an effective way around this issue.

The main contributions are:

1. Taking advantage of a recently developed Hybrid Cybernetic Model (HCM) that can effectively describe glucose and xylose co-fermentation for a modified strain of yeast [42] that shows promise for industrial application [29].

2. Investigating the applicability of an appropriate control algorithm (namely, successive linearization-based Model Predictive Control (slMPC [45]), a computationally attractive alternative to non-linear Model Predictive Control), on the aforesaid HCM model, which includes non-differentiable elements. Doing so successfully mitigates the impact of abruptly changing feed conditions.

3. The selection of an appropriate controlled variable and the attendant optimal control formulation, in the context of slMPC, for the maximization of productivity, subject to conversion requirements. Specifically, the use of conversion as an effective and convenient controlled variable-proxy is established.

4. Demonstrating that the proposed HMM approach results in superior closed-loop performance as compared to employing typical integrated white noise-type descriptions of disturbances.

Section 4.2 gives details of the proposed HMM disturbance model. Since complete state feedback as well as knowledge of the underlying regime trajectory is absent, the general

state estimation strategy is described in Section 4.2.2. Highlights of the high-fidelity hybrid cybernetic chemostat model are presented in Section 4.3. slMPC, the control strategy of choice is briefly presented in general terms in Section 4.4. Due requirements on both productivity and conversion despite having only one degree-of-freedom for control, the selection of an appropriate controlled-variable is explored in Section 4.5. The effectiveness of the proposed HMM disturbance model is demonstrated in Section 4.6. The article concludes in Section 4.7.

4.2 Modeling abruptly changing feed disturbances via HMMs

For the purpose of control, changes in biomass sources and/ or variations in pre-treatment and hydrolysis steps are modeled as probabilistic switches in the concentrations of the glucose and xylose streams entering the chemostat. The Hidden Markov Model (HMM) framework, proposed by [106] for the purpose of modeling the effects of such jump dynamics on general dynamical systems, will be employed.

Indeed, HMMs have found widespread applications in science and engineering - ranging from speech recognition [74] to econometrics [38] - since the 1960s. The main idea is the introduction of a discrete set of latent states, each of which affects the behavior of the dynamical system of concern in a unique way. Specifically, a finite-state Markov chain whose realization at discrete time index t, denoted by $r_t \in \mathcal{J} \triangleq \{1, 2, \ldots, J\}, J \in \mathbb{Z}^+$, is assumed to govern the dynamics of the disturbance. The term 'Hidden' indicates that r_t is not known with probability one and must be inferred from available measurements.

The Markovian jumps[2] are dictated by a transition probability matrix:

$$\Pi = \Big(\mathrm{pr}(r_t = j | r_{t-1} = i) \triangleq p_{ij}\Big)$$

such that the each row of Π sums to one. All Markov chains under consideration are ergodic. For simplicity, the Markov chain is assumed to be at steady state, satisfying $\pi = \Pi'\pi$, where π is a column vector containing the unconditional probabilities of each regime. In other words, π gives the initial probability distribution of r. With this, it is postulated that the

[2] *i.e.*, the transition probabilities depend only on the immediate past.

dynamical system of concern (that is, the chemostat) evolves as follows: [3]:

$$\begin{aligned} x_{t+1} &= f_{\hbar}(x_t, u_t, \theta_t) \triangleq x_t + \int_{t\hbar}^{(t+1)\hbar} f(x_t, u_t, \theta_t) \\ y_t &= g(x_t) + v_t \end{aligned} \tag{41}$$

$$\begin{aligned} \theta_{t+1} &= \mathbb{A}_{r_{t+1}} \theta_t + \omega_{r_{t+1}} \\ pr(r_t = j | r_{t-1} = i) &= \pi_{ij} \end{aligned} \tag{42}$$

Here $x \in \mathbb{R}^{n_x}$ is the continuous system state, $u \in \mathbb{R}^{n_u}$, the manipulated input(s), and v, zero-mean, white Gaussian measurement noise with covariance R^v. Functions f and g capture the continuous-time state-transition and measurement dynamics (of the chemostat) respectively; details are given in Section 4.3. θ is a stochastic, disturbance signal referring to the concentrations of the entering glucose and xylose feeds. In particular, the dynamics of θ are governed by Eq. (42). Having matrix $\mathbb{A}$ as well as the statistics of (the zero-mean, Guassian signal) ω vary according to an underlying Markov chain affords a great deal of flexibility [106] to describe realistic disturbance patterns, including that depicted in Figure 8. The notation $\mathbb{A}_{r_{t+1}}$ signifies that between time interval $[t\hbar, (t+1)\hbar]$, with $\hbar$ denoting the sampling interval, $\mathbb{A}$ may take values from J possible candidates according to the realization of the latent state. With some abuse of notation, the dependence of the statistics of ω on r is denoted by ω_r, per Eq. (42).

For digital controller design, signals (u, θ) are oftentimes assumed to be constant between sampling instants. Consequently, $f_{\hbar}(\cdot)$ yields x_{t+1}, the state vector at the next sampling instant, through numerical integration of f with initial conditions x_t and values of (u, θ) constant over one sample interval, $\hbar$.

Eq. (42) will be specialized depending on the amount of information available for process control and decision making. For example, intimate process knowledge may afford practitioners much insight into the possible mean (though not actual) values of entering glucose and xylose concentrations. In this case, each Markov state may correspond to a specific set of mean values for the glucose and xylose concentrations.

[3] For ease of exposition, discrete-time systems are considered.

Section 4.2.1, the focus of this work, treats the more general case where abrupt jumps (in concentration) of unknown magnitudes need to be described and accounted for, within the HMM framework.

4.2.1 Disturbance modeling via HMMs: unknown mean values of feed concentrations

In the (general) case where the mean values are not known, a slight modification to the notion of the hidden state is required. In particular, the first regime (say) corresponds to a relatively quiescent state, where the concentration of the entering feed hovers around a certain mean level. Regime 2 corresponds to the much rarer event of a large jump to a different mean level. These notions are reflected in Eq. (43).

$$\begin{aligned} \theta_{t+1} &= \theta_t + \omega_{r_{t+1}} \\ \mathbb{E}(\omega_{r_{t+1}=1}\omega'_{r_{t+1}=1}) &\approx 0 \\ \mathbb{E}(\omega_{r_{t+1}=2}\omega'_{r_{t+1}=2}) &>> 1 \\ \pi_{11} = \pi_{21} &\approx 1 \end{aligned} \tag{43}$$

Here, ω is a zero-mean white Gaussian noise signal whose second moment surges whenever a jump occurs. The second regime lasts only briefly, as reflected by the fact that $\pi_{11} = \pi_{21} \approx 1$. Due to the infrequent nature of such jumps, θ tends to track its previous value as the system returns to the quiescent phase.

For both disturbance models, state estimation using the concatenation of Eqs. (41)-(43) is necessary for any feedback control strategy to work, by virtue of the certainty equivalence principle. As such, state estimation for these thus termed 'Markov jump systems', will be briefly discussed in the following section.

4.2.2 State estimation for systems with Markovian jump parameters

Concatenating Eqs. (41) and (42), one gets:

$$\begin{aligned}
\begin{bmatrix} x_{t+1} \\ \theta_{t+1} \end{bmatrix} &= \mathcal{F}_{r_{t+1}}\Big(\begin{bmatrix} x_t \\ \theta_t \end{bmatrix}, u_t, \omega_{r_{t+1}} \Big) \\
y_t &= \mathcal{G}\Big(\begin{bmatrix} x_t \\ \theta_t \end{bmatrix} \Big) + v_t \\
pr(r_t = j | r_{t-1} = i) &= \pi_{ij} \qquad (44)
\end{aligned}$$

Here $\mathcal{F}$ is implicitly understood to include model structures and parameters from $\{f_\hbar, \mathbb{A}_r\}$ and the hidden Markov chain. The statistics of the noise term ω, can also depend on r. That is to say, at during time interval $[t\hbar, (t+1)\hbar]$, ω is sampled from a Gaussian distribution whose parameters vary according to r: $\omega_{t+1} \sim \mathcal{N}(0, R^\omega_{r_{t+1}})$. $\mathcal{G}(\cdot)$, subsuming $g(\cdot)$ is the (concatenated) state-to-output map and $v \sim \mathcal{N}(0, R^v)$, is as before, zero-mean, white Gaussian measurement noise.

For the Markov jump systems this chapter is concerned with, the optimal filter involves an exponentially growing number of linear filters due to the dependence of the noise statistics on r. Without knowledge of the sequence $(r_1, r_2, \ldots, r_t)$, the optimal filter needs to average over all possible past trajectories, the number of which scales as J^t, where J is the cardinality of $\mathcal{J}$, the set containing all possible realizations of r. Therefore, a popular sub-optimal filtering technique, the n-th order Generalized Pseudo-Bayesian (GPBn) algorithm developed by [4] is employed throughout this chapter. For simplicity, the case of $n = 2$ is expanded upon in the following paragraphs.

Define η as the concatenated state $[x', \theta']'$, $\mathcal{R}_t^{t+1} \triangleq (r_t, r_{t+1})$ as a sequence of the two most recent Markov-state realizations and Y_0^t, the measurement sequence $(y_0, \ldots, y_t)$. Accordingly, let $\eta_{t+1|t+1}(\mathcal{R}_t^{t+1})$ denote the estimate of η_{t+1} that accounts for the two most recent Markov-state realizations; the corresponding estimation error covariance matrix is represented as $P_{t+1|t+1}(\mathcal{R}_t^{t+1})$.

If the exact trajectory were known, a time-varying filter would suffice. Since this is not so, a single filter is associated with each possible sequence of changes (of finite length 2).

As a result, the quantities $\{\eta_{t+1|t+1}(\mathcal{R}_t^{t+1}), P_{t+1|t+1}(\mathcal{R}_t^{t+1})\}$, obtained from an application of the Extended Kalman Filter (EKF), initialized with $\{\eta_{t|t}(r_t), P_{t|t}(r_t)\}$, differ in terms of the trajectories considered. The main idea behind GPB2 is to linearly combine these (using *a priori* and *a posteriori* knowledge) to form $\{\eta_{t+1|t+1}(r_{t+1}), P_{t+1|t+1}(r_{t+1})\}$ and finally, a single Gaussian, parameterized by $\{\eta_{t+1|t+1}, P_{t+1|t+1}\}$. Details of a recursive scheme, characterized by two phases: 'branching' and 'merging' have been outlined in the previous chapter (see Chapter 3.3.2).

4.3 A hybrid cybernetic model-based description of the continuous fermentor

Even for organisms much simpler than yeast, metabolic modeling, having to account for numerous reactions and cellular regulatory mechanisms, is a complicated undertaking. Various tractable frameworks have been proposed to alleviate this problem. Specifically, the development of cybernetic models by Ramkrishna and his co-workers has been successful in terms of efficient and high-fidelity modeling of bioreactors [77] for the purpose of process optimization and process control-relevant simulations [25, 36].

Overlaying sets of 'cybernetic' variables on top of (typically-employed) Monod-type kinetics enables the resultant model to account for metabolic regulatory activities. These cybernetic variables, used to modulate the (effective) metabolic fluxes and enzymatic production rates, reflect the fact that micro-organisms allocate resources optimally in order to promote cellular well-being [76]. Cybernetic models with increasing degrees of sophistication with regards to the strategic behavior of the micro-organism of concern have been developed [40, 75, 109, 108].

Recently, [92] developed a version of Hybrid Cybernetic Models (HCMs) (first introduced by [39]) appropriate for large-scale metabolic networks. In particular, the method was applied to an engineered strain of *Saccharomyces* yeast (1400(pLNH33)) capable of co-fermenting glucose and xylose, the main fermentable sugars resulting from the pre-processing of lignocellulosic biomass. This strain was first engineered through the introduction of genetic material from *Pichia stipitis* and *Saccharomyces cerevisiae*, to impart the ability of xylose-utilization into the host strain, *Saccharomyces* yeast 1400 [42].

Recent literature has noted the promise of transformed versions of *Saccharomyces* yeast 1400 (in particular, transformant 1400(pLHN32) with a cofermentation effectiveness similar to that of 1400(pLHN33) [31]), for industrial-scale lignocellulosic ethanol production. As such, the HCM model for 1400(pLNH33), suitably adjusted for continuous-mode ethanol production, will be presented and expanded upon in the following sub-section and used for numerical simulations in the sequel.

4.3.1 Overview of a hybrid cybernetic chemostat model seeded with Saccharomyces yeast 1400(pLNH33)

The large number of (intra-and-extra-cellular) reactions, representative of a complex metabolic network, such as that of *Saccharomyces* yeast 1400(pLNH33) necessitates the concept of Elementary Flux Modes (EFMs) [85]. Approximately speaking, the EFMs span the steady-state flux (*i.e.*, reaction rate) distributions of any biochemical network and form a set of nondecomposable [84] pathways consisting of a minimal set of reactions that function at steady state. In the context of this article, the EFMs are derived from the stoichiometric matrix corresponding to the intra-celluar reactions, coincident with the assumption that the latter are at (pseudo)-steady state.

Due to the large number (201) of EFMs within the 1400(pLNH33) network, [92] employed a two-step EFM-reduction technique (termed 'yield analysis') to reduce the number significantly to three representative clusters of 'active EFMS': i) glucose uptake (4 EFMs), ii) xylose uptake (3 EFMs) and iii) simultaneous glucose and xylose uptake (5 EFMs). Each of these biologically meaningful active modes, determined from *a priori* knowledge and experimental data, converts a source of carbon to either biomass or energy.

With this, the continuous-time ordinary differential equations for the chemostat are given by Eqs. (45)-(46):

$$\frac{d\tilde{x}}{dt} = [S_{\tilde{x}}Z] \cdot \mathrm{diag}([\tilde{\nu}_1, \ldots, \tilde{\nu}_{12}]) \cdot \tilde{r} \cdot x_{BIOM} + D(\tilde{x}_{in} - \tilde{x}) \tag{45}$$

Here, $\tilde{x} \in \mathbb{R}^{9\times 1}$ is the concentration vector of extra-cellular species:

$$\tilde{x} \triangleq [x_{GLC}, x_{XYL}, x_{ACTx}, x_{BIOM}, x_{CO_2}, x_{ETH}, x_{GOLx}, x_{MAINT}, x_{XOLx}]'$$

Table 6: Description of subscripts used in Eq. (45)

Metabolite abbreviation	Description
GLC	glucose
XYL	xylose
ACTx	acetate (external)
BIOM	biomass
CO_2	carbon dioxide
ETH	ethanol
GOL_x	glycerol (external)
MAINT	excess energy (consumed for maintenance)
XOLx	xylitol (external)

The definitions of the subscripts are given in Table 6; focus is given to the first, second, fourth and sixth elements. These represent glucose, xylose, ethanol and biomass concentration levels (in milli-molarity ($10^{-3}\times$ mole/L) or (g/L) [4]) respectively. The term $S_{\tilde{x}}Z$ represents the product of the extra-celluar stoichiometric matrix ($S_{\tilde{x}}$) and the concatenation of the twelve active elementary flux modes (Z). $\tilde{r} \in \mathbb{R}^{12\times 1}$ constitutes the flux vector. Each reaction $\tilde{r}_i$, (dependent on Monod-type kinetics and enzymatic concentrations (x_E)) corresponds to a unique EFM and is modulated by an accompanying cybernetic variable $\tilde{\nu}_i \in [0, 1]$, lying in the unit interval. It reflects a relative 'return-rate' of investing cellular resources on the i^{th} EFM. In this case, $\tilde{\nu}_i$ is a direct function of the extent of the carbon uptake rate and may be interpreted as the effectiveness of the i^{th} EFM in utilizing the (respective) sugar feeds (for the purpose of energy or biomass production). D (1/h) refers to the dilution rate and $\tilde{x}_{in}$ is the concentration of the extra-celluar species in the entering feed stream. In general, the feed stream is assumed to contain only glucose and xylose and will be free of other metabolites such as acetate, xylitol, etc. The specific forms of $\tilde{r}$ and $\tilde{\nu}$, and values of the kinetic parameters to be employed during the simulation are contained in Appendix A. Each EFM is assumed to be catalyzed by an enzyme. The equations for enzymatic synthesis are represented by Eq. (46).

$$\frac{dx_E}{dt} = \alpha + \text{diag}([\tilde{u}_1, \ldots, \tilde{u}_{12}]) \cdot \tilde{r}_E - \text{diag}([\beta_1, \ldots, \beta_{12}]) \cdot x_E - \mu \cdot I x_E \tag{46}$$

[4]Per convention, feed and ethanol concentrations are generally presented in the text and figures with units of (g/L) in this work. All simulations, with the exception of biomass (g/L), are however, run on a milli-molarity basis. Units conversion, by multiplication with molecular weight or its inverse, is therefore necessary where appropriate.

Here, $x_E \in \mathbb{R}^{12\times 1}$ refers to the concentration of enzymes. The terms on the right hand side of Eq. (46) represent constitutive synthesis, inducible synthesis, death and dilution (by growth) rates respectively. Vector $\tilde{r}_E$ represents the rates of enzymatic synthesis, which are regulated by another set of cybernetic variables, $\tilde{u} \in [0, 1]$. The mathematical definitions of $(\tilde{u}, \tilde{\nu})$ are as such:

$$\tilde{u}_i = \frac{\mathbf{r}_i}{\sum_j \max\{\mathbf{r}_j, 0\}},\ i = 1, 2, \ldots, 12 \tag{47}$$

$$\tilde{\nu}_i = \frac{\mathbf{r}_i}{\max_j\{\mathbf{r}_j\}},\ i = 1, 2, \ldots, 12 \tag{48}$$

Here, $\mathbf{r}_i$, $i = 1, 2, \ldots, 12$, refers to the time-varying return-on-investment (*i.e.*, energy or biomass production, the definitions of which are given in Table 7) corresponding to each of the twelve elementary flux modes. $\mathbf{r}_i$, $i = 1,]\ldots, 12$ are essentially scaled versions of $\tilde{r}_i$. Note that the cybernetic variables $(\tilde{u}, \tilde{\nu})$ are not differentiable, posing an additional level of complexity when linearizing ordinary differential equations (Eqs. (45)-(46)) for the design of the EKF and linearization-based controller (Section 4.4).

μ refers to the specific growth rate (of biomass) and is given by: $\mu \triangleq \frac{1}{x_{BIOM}} \frac{dx_{BIOM}}{dt}$.

Together, Eqs. (45)-(46) define the continuous-time differential equations governing the chemostat (that is, f in Eq. (41)). The system's (continuous) state vector is therefore defined as: $x \triangleq [\tilde{x}', x_E']' \in \mathbb{R}^{21\times 1}$, with the assumption that the chemostat is at a constant volume of 1 (L).

Table 7: Rate-of-investment for each elementary flux mode.

Return on investment	Definition
$\mathbf{r}_i$	$6\tilde{r}_i,\ i = 1, 2, 3$
$\mathbf{r}_i$	$5\tilde{r}_i,\ i = 4, 5, 6, 7$
$\mathbf{r}_8$	$26\tilde{r}_8$
$\mathbf{r}_9$	$26\tilde{r}_9$
$\mathbf{r}_{10}$	$60.4\tilde{r}_{10}$
$\mathbf{r}_{11}$	$7.88\tilde{r}_{11}$
$\mathbf{r}_{12}$	$6.06\tilde{r}_{12}$

4.4 Control strategy: successive linearization-based Model Predictive Control (slMPC)

Model Predictive Control (MPC) is a popular advanced process control solution used in process industries. The main idea is to solve, at each sample time, an open-loop optimization problem minimizing the deviation of the predicted plant trajectory (over a horizon of p steps) from the desired one. This yields a finite sequence of controller actions. The first action is implemented, one unit-sample of time elapses and the process repeats in a rolling-horizon fashion.

The hybrid cybernetic models employed are highly non-linear due to the cybernetic regulation variables, as well as the Monod-type kinetics. Coupled with the existence of multiple feed conditions, controllers based on one linear model are not expected to perform well. A full nonlinear MPC implementation is difficult to solve reliably in a reasonable period of time due to the large number of model equations and generally long prediction horizon required. As such, an MPC scheme based on successive linearization is preferred.

In particular, slMPC [45] is a computationally attractive alternative to a full nonlinear MPC implementation. The key lies in successively obtaining a linear approximation to the nonlinear plant, and employing it for the purpose of multi-step ahead prediction and control. Given appropriately chosen constraints (*i.e.*, linear equality and inequality ones only), the resulting open-loop optimization problem is a Quadratic Program (QP), one for which exist efficient optimization solvers based on interior point, active-set and/ or conjugate gradient methods [11]. The steps in implementing the slMPC algorithm are summarized below.

- At each sample time t, obtain state and disturbance estimates $\{x_{t|t}, \theta_{t|t}\}$ using the appropriate chemostat and disturbance model. These are used to estimate the values of the flux distribution, $\tilde{r}_{t|t}$, the rate-of-return vector, $\mathbf{r_{t|t}}$ (see Table 7) and the approximate analytical forms of $(\tilde{u}, \tilde{\nu})$, (see Eqs. (49)-(50)), to be used for linearization.

$$\tilde{u}_{t|t} = \frac{\mathbf{r}_{t|t,i}}{\sum_{l\in\mathcal{L}} \mathbf{r}_{t|t,l}}, \text{ where } \mathcal{L} \triangleq \{l : \mathbf{r}_{t|t,l} \geq 0\} \tag{49}$$

$$\tilde{\nu}_{t|t,i} = \frac{\mathbf{r}_{t|t,i}}{\mathbf{r}_{t|t,i^*}}, \text{ where } i^* = \arg\max_j \{\mathbf{r}_{t|t,j}\} \tag{50}$$

  Note that the original functional forms of the cybernetic variables are non-differentiable.

- Then, with the approximate analytical forms of the cybernetic variables, linearize the model equations around the state and disturbance estimate and previous control move $(\eta_{t|t} \triangleq [x'_{t|t}, \theta'_{t|t}]', u_{t-1})$.

- Obtain the p-step-ahead model predictions for the controlled variable(s) of interest, $y^c_{t+j|t}$, $j = 0, 1, \ldots, p$, for a given choice of $m(\leq p)$ manipulated variables. In general, y^c_t may be a function of x_t and u_t. See Section 4.4.1 for details.

- Using the model predictions, formulate and solve a quadratic program with meaningful constraints. Section 4.4.2. Implement the control move u_t whilst discarding the others and repeat the process.

4.4.1 p-step ahead prediction

At each sampling time, the continuous time model is linearized and discretized yielding the multi-step ahead predictor, Eqs. (51)- (52). A detailed derivation is deferred to [45].

$$x_{t+i|t} = f_{i\hbar}(x_{t|t}, u_{t-1}, \theta_{t|t}) + [\sum_{j=0}^{i-1} (A^{sl}_t)^j B^{sl}_t | \ldots | B^{sl}_t] \Delta \mathcal{U}^{t+i-1}_t, \; i = 1, ..., p \tag{51}$$

$$y^c_{t+i|t} = g^c(x_{t|t}, u_{t-1}) + C^{sl}_t (x_{t+i|t} - x_{t|t}), \; + [D^{sl}_t | \ldots | D^{sl}_t] \Delta \mathcal{U}^{t+i-1}_t, \; i = 0, 1, \ldots, p \tag{52}$$

Here, $\Delta \mathcal{U}^{t+i-1}_t \triangleq [\Delta u_t, \ldots, \Delta u_{t+i-1}]'$, is a vector of future control moves. $m\,(\leq p)$ denotes the control horizon, beyond which it is assumed that the controller action is constant. That is, $\Delta u_{t+m} = \ldots = \Delta u_{t+p-1} \triangleq 0$. In Eq. (51), the first term is from integrating the nonlinear model directly whereas the second term is an approximation based on the linearized model. Future predictions of the controlled variable, $y^c_{t+i|t}, i = 1, 2, \ldots, p$, in Eq. (52), is based on a Taylor-series approximation about $(x_{t|t}, u_{t-1})$.

A^{sl}_t is obtained through differentiation of the most probable realization of $\mathcal{F}$ (*i.e.*, $\mathcal{F}_{r_{t|t-1}}$ is determined by applying Eq. (23)) with respect to x and evaluating the derivative at $(\eta_{t|t}, u_{t-1})$. Note that the linearization is around the previous input, u_{t-1}. B^{sl}_t is obtained in a similar fashion. Matrices (C^{sl}_t, D^{sl}_t) are obtained by differentiating $g^c(\cdot)$, the function that maps the state variables to the controlled variable(s), with respect to x and u respectively.

The use of linear approximations for the last terms of Eqs. (51)-(52) circumvents the need to solve a nonlinear optimization problem in determining the control action u_t. Implicit in Eqs. (51)-(52) is the assumption that the (infrequently switching) Markov state and matrices $(A_t^{sl}, B_t^{sl}, C_t^{sl})$ remain constant throughout the prediction horizon.

4.4.2 Control move implementation

A sequence of optimal moves that minimizes the expected future error are computed based on Eq. (52). The desired prediction equation can be written as:

$$\mathcal{Y}_t^c \quad = \quad \mathcal{S}_1 + \mathcal{S}_2 \Delta \mathcal{U}_t^{t+m-1} \tag{53}$$

where $\mathcal{Y}_t \triangleq [y_{t|t}^c, y_{t+1|t}^c, \ldots, y_{t+p|t}^c]'$ is the vector of future predictions of the Controlled Variable (c.v.). $\mathcal{S}_1$ is the vector of predicted future c.v.'s assuming the controller action stays at u_{t-1}:

$$\mathcal{S}_1 \quad = \quad \begin{bmatrix} g^c(x_{t|t}, u_{t-1}) \\ g^c(x_{t|t}, u_{t-1}) - C_t^{sl} x_{t|t} \\ g^c(x_{t|t}, u_{t-1}) - C_t^{sl} x_{t|t} \\ \vdots \\ g^c(x_{t|t}, u_{t-1}) - C_t^{sl} x_{t|t} \end{bmatrix} + \begin{bmatrix} 0 \\ C_t^{sl} f_{\hbar}(x_{t|t}, u_{t-1}, \theta_{t|t}) \\ C_t^{sl} f_{2\hbar}(x_{t|t}, u_{t-1}, \theta_{t|t}) \\ \vdots \\ C_t^{sl} f_{p\hbar}(x_{t|t}, u_{t-1}, \theta_{t|t}) \end{bmatrix}$$

$\mathcal{S}_2$, a compensatory term, is given by:

$$\mathcal{S}_2 \quad = \quad \begin{bmatrix} D_t^{sl} & 0 & \ldots & 0 \\ C_t^{sl} B_t^{sl} + D_t^{sl} & D_t^{sl} & \ldots & 0 \\ C_t^{sl}(A_t^{sl} B_t^{sl} + B_t^{sl}) + D_t^{sl} & C_t^{sl} B_t^{sl} + D_t^{sl} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_t^{sl} \sum_{i=0}^{p-1} (A_t^{sl})^i B_t^{sl} + D_t^{sl} & \ldots & \ldots & C_t^{sl} \sum_{i=0}^{p-m} (A_t^{sl})^i B_t^{sl} + D_t^{sl} \end{bmatrix}$$

The following quadratic minimization is solved to obtain the desired sequence of control moves within the horizon.

$$\begin{aligned} \min_{\Delta \mathcal{U}_t^{t+m-1}} \quad & ||\mathcal{Y}_t^c - \mathcal{Y}_t^*||_{\Lambda^{\mathcal{Y}}}^2 + ||\Delta \mathcal{U}_t^{t+m-1}||_{\Lambda^{\mathcal{U}}}^2 \\ & \Phi \cdot \Delta \mathcal{U}_t^{t+m-1} \leq \phi \\ & \Sigma \cdot \Delta \mathcal{U}_t^{t+m-1} = \sigma \end{aligned} \tag{54}$$

$\mathcal{Y}_t^*$ is a vector denoting the desired c.v. trajectory; $\Lambda^{\mathcal{Y}}, \Lambda^{\mathcal{U}}$, are quantities that induce 2-norms, reflecting the tradeoff between set point tracking and excessive actuator movement. Eq. 54 represents a quadratic program that can be solved efficiently (by MATLAB's$^{\text{TM}}$ `quadprog` function, for instance).

4.5 Selection of an appropriate controlled variable: simulation studies

As mentioned, a key quantity is ethanol productivity (defined in Eq. (55)), obtained from the product of dilution rate (D) and the concentration of ethanol (the desired product) in the outlet stream. It is expected that dilution rate (D) is the only manipulated variable. Since increasing D has the effect of monotonically (and strictly) decreasing the product concentration, a unique maximal value for productivity must exist for each set of feed conditions at equilibrium, as shown in Figure 9(a) [91].

However, economics typically dictate a sufficiently high conversion rate (of 95%, say, or greater). This implies that the maximal allowable productivity will occur at values of D corresponding to the lowest tolerable conversion[5] (defined in Eq. (56)).

$$\mathcal{P}_t = x_{ETH,t} \cdot u_t \text{ (g/L.h)} \tag{55}$$

$$\mathcal{X}_t = 1 - \frac{x_{GLC,t} + x_{XYL,t}}{x_{GLC,in,t} + x_{XYL,in,t}} \tag{56}$$

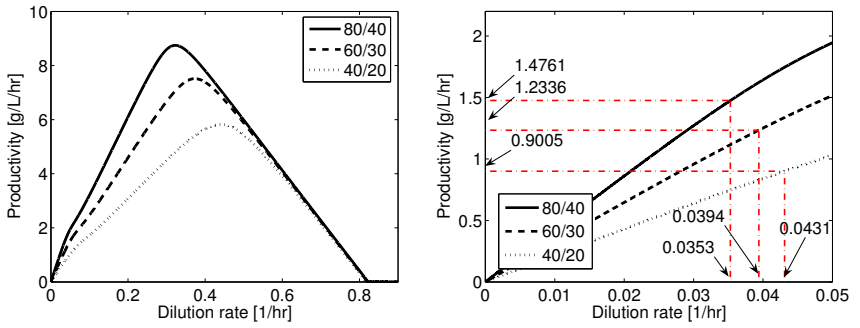
These insights are depicted in Figure 9 revealing the steady-state profiles that correspond to the various feed conditions. For example, when the system switches from regime 1 (80/40) to regime 2 (60/30), due to the conversion constraints (of 95% or greater), the maximal attainable productivity will likewise shift from 1.4761 to a lower value of 1.2336 (g/L/h).

Given that there is only one degree of freedom to meet the dual (and conflicting) requirements of high productivity and conversion, a systematic study with regards to the choice of a suitable controlled variable is needed, which is the focus of the following sub-sections.

4.5.1 Productivity as the CV

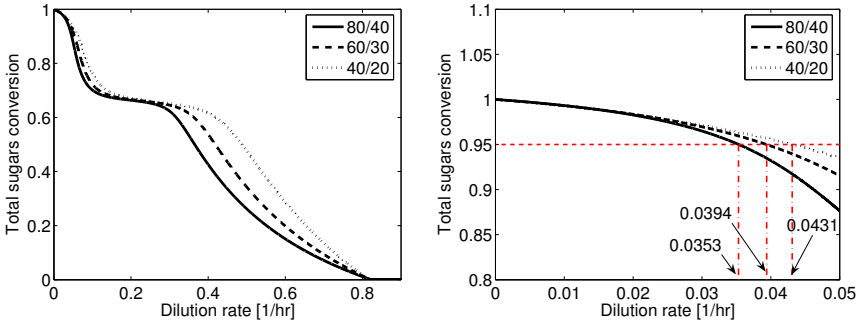
The most direct choice of productivity, $\mathcal{P}$, as the controlled variable, with a hard constraint on the conversion typically results in infeasibility in the response to changes in entering

[5] where concentrations are reported in (g/L)



(a) Productivity vs. Dilution rate

(b) Productivity vs. Dilution rate (zoomed-in)



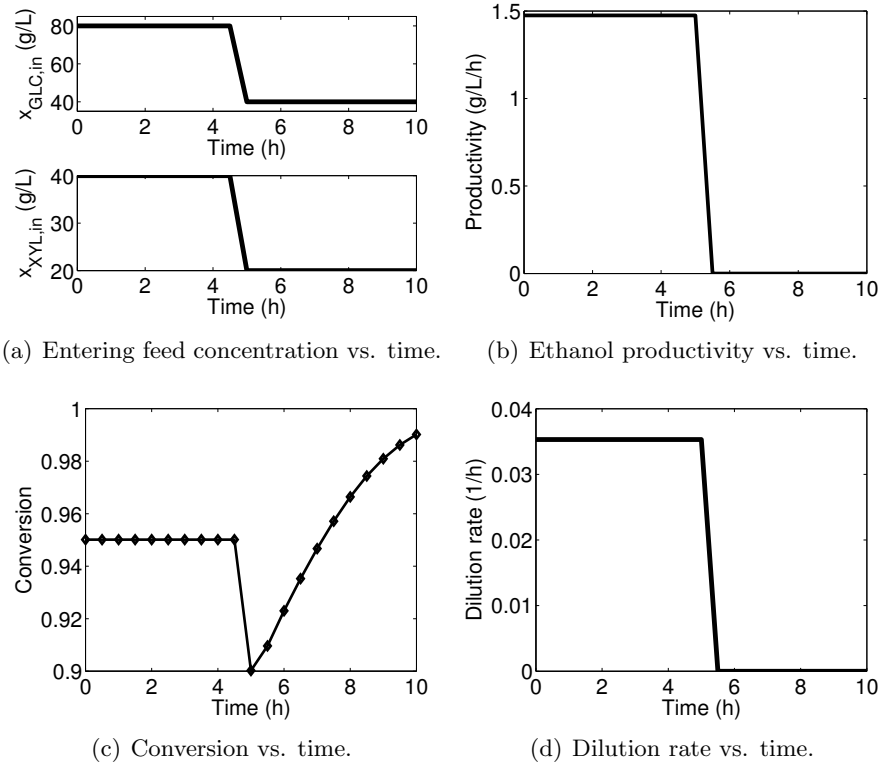
(c) Conversion vs. Dilution rate

(d) Conversion vs. Dilution rate (zoomed-in)

Figure 9: Steady state productivity and conversion profiles for various entering glucose/ xylose (g/L) concentrations. The faint (red) lines correspond to the 95% conversion limit.

feed conditions. Consider a feed switch at $t = 5$ (h), from initial regime 1 to regime 3 (see Table 5), as depicted in Figure 10(a). Such a sudden shit from a regime with a higher feed concentration to one with lower concentration will cause the denominator in Eq. (56) to decrease and the conversion, $\mathcal{X}$, to likewise drop temporarily below the 95% threshold.

In order to compensate for this decline in a manner such that the 95% bound is not violated during subsequent steps, the (non-negative) dilution rate needs to be decreased (see Figure 9(d)). However, even with the most drastic of reductions, as shown in Figure 10(d), it still takes approximately two hours for sugars-conversion to re-enter appropriate bounds. Figure 10(b) reveals the effect of decreasing the flow rate to zero in order to increase the conversion. These observations suggest that any control algorithm with the productivity as the controlled variable of choice as well as the imposition of hard constraints on the conversion will most likely run into infeasibility issues.



(a) Entering feed concentration vs. time.

(b) Ethanol productivity vs. time.

(c) Conversion vs. time.

(d) Dilution rate vs. time.

Figure 10: Plots of productivity, conversion and dilution rate in response to a sudden decrease (from regime 1 (80/40) to regime 3 (60/30)) in feed conditions at $t = 5$. Note the constraint violation despite decreasing dilution rate to 0 (1/h).

4.5.2 Investigating the incorporation of a soft constraint on conversion

In light of the previous discussion, constraint-violations are inevitable. As such, a soft constraint on the conversion, reflecting a trade-off between the desire to have high productivity and imposing a penalty on excessively low levels of conversion, may be imposed. Specifically, consider the following specialization of Eq. (54)

$$\min \sum_{i=0}^{p} \mathcal{Q}_{\mathcal{P}} \left(\mathcal{P}_{t+i|t} - \mathcal{P}_t^{sp}\right)^2 + \sum_{i=1}^{p} \mathcal{Q}_{\mathcal{X}} \left(\epsilon_{t+i|t}^2\right) + \sum_{i'=0}^{m-1} \mathcal{R}(\Delta D_{t+i'})^2 \tag{57}$$

$$0 \leq \epsilon_{t+i|t},\, i = 0, 1, \ldots, p \tag{58}$$

$$\mathcal{X}_{t+i|t} \geq \mathcal{X}^{min} - \epsilon_{t+i|t},\, i = 0, 1, \ldots, p \tag{59}$$

$$0 \leq D_{t+i},\, i = 0, 1, \ldots, (m-1) \tag{60}$$

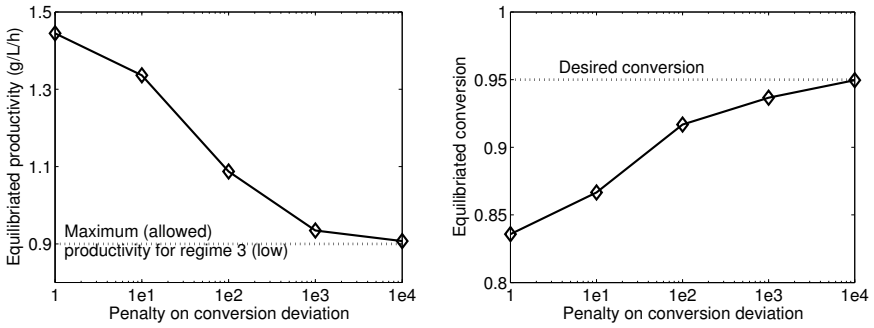
The lower bound (Eq. (60)) ensures physically meaningful control actions. $\mathcal{P}^{sp}$ and $\mathcal{X}^{min} \triangleq 95\%$ refer to the (potentially time-varying) targets for productivity and conversion respectively. Since $D_{t+i} = D_{t-1} + \sum_{i=0}^{t} \Delta D_{t+i}$, Eq. (60) can be easily re-written as a linear inequality constraint involving the previous control action and ΔD_{t+i}, $i = 0, \ldots, (m-1)$.

Intuitively, allowing $\mathcal{P}^{sp}$ to change with the underlying regime (see Figure 9(b)) whilst imposing appropriate weights on $\mathcal{Q}^{\mathcal{P}}$ and $\mathcal{Q}^{\mathcal{X}}$ would give good closed-loop performance. However, due to difficulties in determining the actual regime the system is in, and a lack of intimate knowledge on steady-state profiles (as represented in Figure 9), adjusting the set-point according to the feed condition may not be practicable. In this case, the following assignments are made

$$\begin{aligned} \mathcal{P}^{sp} &= 1.4761 \text{ (g/L/h)} \\ \mathcal{X}^{min} &= 0.95 \end{aligned}$$

$\mathcal{P}^{sp}$ represents the highest possible productivity across all three regimes. However, for regimes 2 and 3, attaining this setpoint is only possible at the expense of lower (but economically unacceptable) conversions. In this case, an appropriate choice of $\mathcal{Q}_{\mathcal{P}}$ and $\mathcal{Q}_{\mathcal{X}}$ needs to be enforced, although one may expect a larger weight needs to be imposed on the conversion than on the productivity, especially when the system transitions to regime 3.

In order to test the above idea, the following study is made. For simplicity, complete state feedback is assumed to be present and the entering feed signals are also measured. $\mathcal{R}$ is set to a negligible value. Figure 11 shows the steady-state values of the productivity and conversion for various ratios of $\mathcal{Q}_{\mathcal{X}}$ to $\mathcal{Q}_{\mathcal{P}}$ in response to a change in feed conditions from regime 1 to 3. The latter represents the most severe change that may occur. It is noted that



(a) Productivity vs $\frac{\mathcal{Q}_{\mathcal{X}}}{\mathcal{Q}_{\mathcal{P}}}$ (b) Conversion vs $\frac{\mathcal{Q}_{\mathcal{X}}}{\mathcal{Q}_{\mathcal{P}}}$

Figure 11: Investigating closed loop performance as a function of penalty weights in response to a shift in feed conditions from regime 1 to regime 3.

given the requirement of a fixed set-point, a very high value of $\frac{\mathcal{Q}_{\mathcal{X}}}{\mathcal{Q}_{\mathcal{P}}}$ (about 10^4) is required if the conversion requirements are to be satisfied (at 95%).

4.5.3 Conversion as the controlled variable of choice

In view of Figure 11, and for the given control requirements, a convenient proxy for the productivity is the conversion, $\mathcal{X}$. The reasons are as follows:

1. A single set-point (of 95%) is sufficient.

2. For relatively high conversion levels, operating at the lowest acceptable conversion (for any feed regime) corresponds to maximal productivity.

3. There exists a one-to-one (inversely proportional) mapping between dilution rate and conversion. In the case of productivity, due to the existence of a unique maximum, a given productivity may be achieved at a low dilution rate (or high conversion) or at high dilution rate (or low conversion). Such non-linearity may present control difficulties especially in the event of plant-model mismatch.

Subsequently, g^c (see Eq. (52)), the state-to-c.v. map is given by Eq. (56). The quadratic control objective, to be used in the remaining discussions, is summarized by Eqs. (61)-(62):

$$\min_{\Delta D_t^{t+m-1}} \sum_{i=1}^{p} \mathcal{Q}_{\mathcal{X}} \left(\mathcal{X}_{t+i|t} - 0.95\right)^2 + \sum_{i'=0}^{m-1} \mathcal{R} \left(\Delta D_{t+i'}\right)^2 \tag{61}$$

$$0 \leq D_{t+i},\, i = 0, 1, \ldots, (m-1) \tag{62}$$

4.6 Rejecting unmeasured feed disturbances

Previous sections have established a systematic framework for disturbance modeling (Section 4.2.1 - 4.2.1), a computationally feasible Model Predictive Control algorithm (Section 4.4), and a state estimation algorithm (suitable for the overall, concatenated Markov jump system) to be used in conjunction with the aforesaid control strategy (Section 4.2.2). Also, the conversion has been ascertained to be a suitable and convenient controlled variable (Section 4.5.3).

In this section, the effectiveness of slMPC for rejecting unmeasured, entering feed disturbances that exhibit abrupt but significant changes is explored by means of numerical experiments. The only manipulated variable is the entering feed flow rate, or equivalently, the dilution rate (D) since the volume of the chemostat is assumed to be constant (at a nominal value of 1 (L)). As postulated in Section 4.1, entering glucose and xylose concentrations ($x_{GLC,in}, x_{XYL,in}$) switch between regimes at randomly selected times, albeit infrequently. A representative scenario is described as such:

$$\begin{bmatrix} x_{GLC,in} \\ x_{XYL,in} \end{bmatrix}_t = \begin{cases} \mathbb{A}_1 + \omega_t, t \in [0, 100) \\ \mathbb{A}_3 + \omega_t, t \in [100, 200) \\ \mathbb{A}_2 + \omega_t, t \in [200, 300) \end{cases} \tag{63}$$

where $\mathbb{A}_i$ refers to the mean entering feed concentration (in (g/L)) for regime i (see Table 5). The choice of mean dwelling time regime is 100 (h), well beyond the settling time of the system. This selection is therefore consistent with the assumption of infrequent jumps. Noise vector $\mathbb{E}(\omega\omega')$ is set to diag([1.8, 1.5]) (g^2/L^2) for all three regimes

The measured variables are assumed to be the concentrations of glucose, xylose and the desired product, ethanol: $[x_{GLC}, x_{XYL}, x_{ETH}]$. The state to observation vector map is

therefore given by Eq. (64):

$$y_t^o = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix} x_t + v_t \quad (64)$$

For all simulations, the covariance of the measurement noise ($R \triangleq \mathbb{E}(vv')$) is set to diag($[0.018, 0.75, 2.3]$) ($g^2/L^2$) and made available to all controller/ estimator schemes, to be described as follows.

4.6.1 Various disturbance models: comparing closed-loop performance

The rejection of these unmeasured, abruptly changing feed conditions, is facilitated through a disturbance model. An appropriate choice (of the latter) allows a state-estimator to be built so that the disturbance signals may be accurately reconstructed, thereby facilitating feedback control through slMPC.

As such, two candidate disturbance models are studied for the purpose of describing the disturbance patterns delineated in the opening paragraphs of this Section. In all cases, the regime that the system is in is not known. The following models are considered:

- Eq. (65) is the commonly employed, non-switching, Integrated White Noise (IWN) description (of the entering feed signal).

$$\theta_{t+1} = \theta_t + \omega_{t+1} \quad (65)$$

  Here, ω is a zero-mean, white Gaussian noise signal with a time-invariant second moment, $\mathbb{E}(\omega\omega')$. In general, how this parameter may be tuned is not clear. Small values of $\mathbb{E}(\omega\omega')$ relative to R^v, the measurement noise covariance, typically lead to sluggish responses to output feedback, whereas the reverse situation results in over-sensitivity to measurement noise. These two disturbance models are referred to as IWN(lo) and IWN(hi), respectively, with the following covariance matrices:

$$\mathbb{E}(\omega\omega')_{IWN,lo} = \text{diag}([1.8 \times 10^{-5},\, 1.5 \times 10^{-5}])\,(g^2/L^2)$$
$$\mathbb{E}(\omega\omega')_{IWN,hi} = \text{diag}([18,\, 15])\,(g^2/L^2)$$

  These values represent two extreme choices.

- Eq. (43) represents switching disturbances with unknown mean values (*i.e.*, an HMM-based disturbance model). Here, the following transition probability matrix and (regime-dependent) noise covariance matrices are used:

$$\begin{aligned} \Pi &= \begin{bmatrix} 0.99 & 0.01 \\ 0.99 & 0.01 \end{bmatrix} \\ \mathbb{E}(\omega_{r=1}\omega'_{r=1}) &= \text{diag}([1.8 \times 10^{-5},\, 1.5 \times 10^{-5}])\,(\text{g}^2/\text{L}^2) \\ \mathbb{E}(\omega_{r=2}\omega'_{r=2}) &= \text{diag}([18,\, 15])\,(\text{g}^2/\text{L}^2) \end{aligned}$$

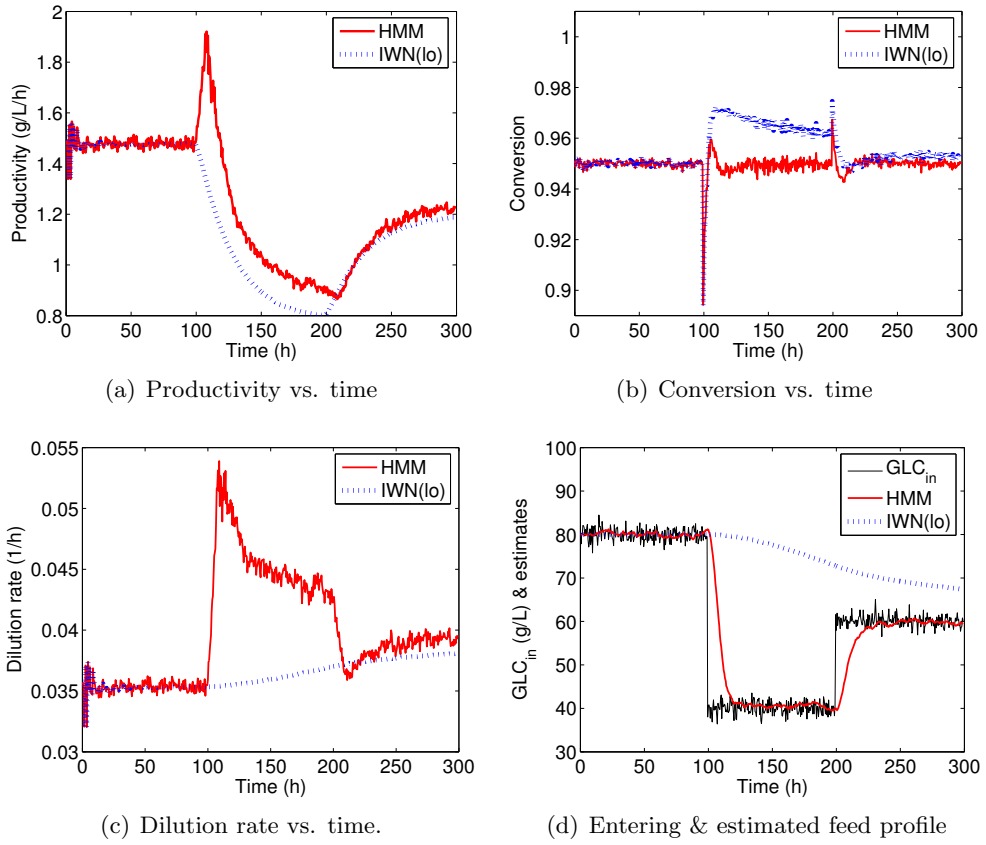
  As mentioned in Section 4.2, $r = 1$ corresponds to the quiescent phase, whereas $r = 2$, indicates the occurrence of a rare jump event. It is noted that $\mathbb{E}(\omega_r\omega'_r)$ have been chosen to correspond to the two extreme cases for the IWN-model.

The (HMM-based) disturbance model (HMM) necessitate the use of a GPB2 state estimator (Section 4.2.2), whereas a (non-switching) extended Kalman filter suffices for the first case. In all simulations, the penalty on excessive actuator movement, $\mathcal{R}$, is set to a negligible value of 0.1% that of $\mathcal{Q}_{\mathcal{X}}$. Figure 12 and Figure 13 are characteristic of a typical stochastic realization, with the initial regime 1 switching to regime 3 and finally to regime 2.

From Figs. 12(c), 13(c) and Figs. 12(d), 13(d) , it can be observed that, in tracking the unmeasured feed signal (only the glucose profile is shown), disturbance model IWN(lo) suffers from low sensitivity to output feedback, whereas, model IWN(hi), from over-sensitivity to measurement noise. State estimation resulting from the proposed HMM-based disturbance model indicates appropriate switching between insensitivity to measurement noise (during the quiescent phase) and sensitivity to large feed jumps. The reason is that the GPB2 estimator corresponding to this HMM framework serves effectively as a time-varying filter. It is noted that tuning the GPB2 estimator (for the HMM case) is relatively easier than doing so for an IWN-based estimator [106]. For handling infrequent, abrupt steps, π_{11} and π_{12} need only to be close to unity. Furthermore, the values of $\mathbb{E}(\omega_r\omega'_r)$ for $r = 1$ and $r = 2$ are easily set by selecting variances that of low and high values.

Looking at the productivity profiles, it is noted that IWN(lo) generated the lowest productivity levels due to the excessive sluggishness of the state estimator. This is also

reflected in the levels of conversion larger than 95% for the IWN(lo) case. Conversion profiles for IWN(hi) and the proposed HMM case are similar, since noise that directly enters the feed channel is incorporated into the conversion computations (Eq. (56)). Nevertheless, productivity profiles can be thought of as the most critical benchmark. Although the mean productivity profile for IWN(hi) is similar to that of the proposed HMM approach, the former suffers from a larger variance, to the detriment of effective, predictable production. It is noted that for each feed regime, the controller is able to bring the system to the appropriate productivity values (Figure 9).



(a) Productivity vs. time

(b) Conversion vs. time

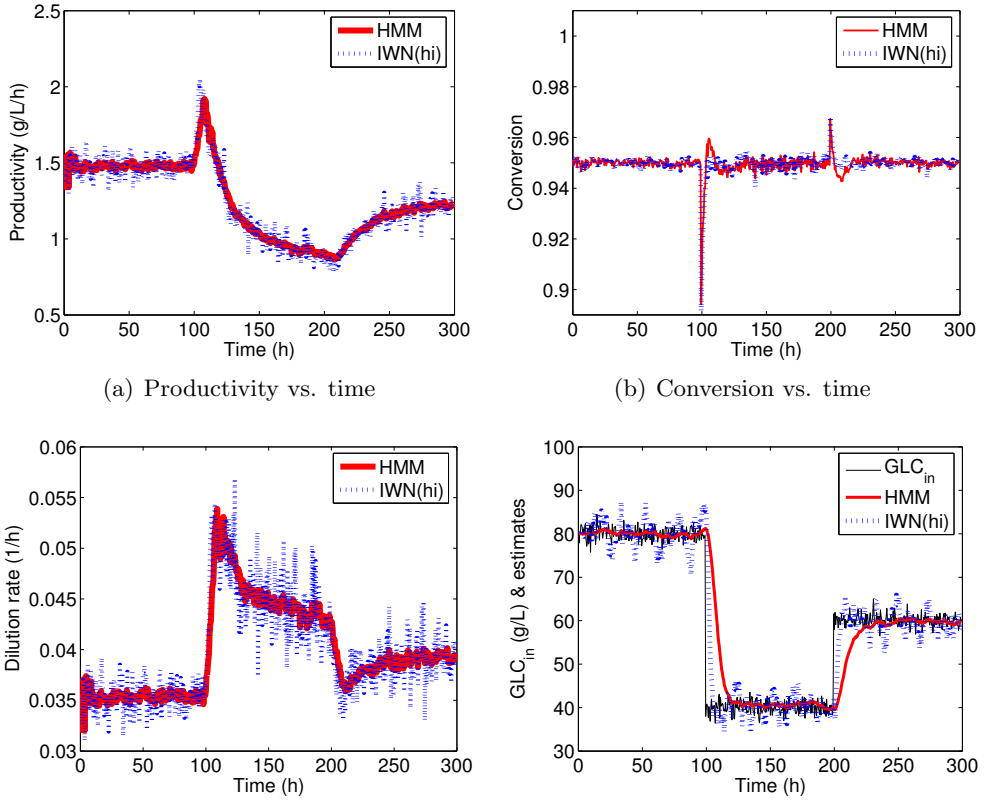
(c) Dilution rate vs. time.

(d) Entering & estimated feed profile

Figure 12: State profiles corresponding to various disturbance models: Proposed HMM method vs. IWN(lo)

4.7 Conclusions

In order for ethanol to play a greater role as an alternative source of energy, control strategies accounting for highly varying but abruptly changing feed conditions (as considered in



(a) Productivity vs. time

(b) Conversion vs. time

(c) Dilution rate vs. time.

(d) Entering & estimated feed profile

Figure 13: State profiles corresponding to various disturbance models: Proposed HMM moethod vs. IWN(hi)

this article) are needed. This work studies the applicability of slMPC in the context of a chemostat bioreactor (seeded with yeast capable of co-fermentation) for producing ethanol. The development of a special version of hybrid cybernetic models enables accurate *in-silico* experiments to be conducted. Also, the use of HMM-based disturbance models enables a more effective state estimation strategy with consequent benefits on closed-loop control. The conversion, rather than the productivity, turns out to be the most effective choice of c.v.

Future work calls for the development and application of cybernetic models for other promising yeast strains [29], with the inclusion of temperature and ionic effects.

CHAPTER V

FAULT DETECTION IN PROCESS SYSTEMS USING HIDDEN MARKOV DISTURBANCE MODELS

5.1 Introduction

Tracking the closed-loop performance and health of process systems, although intuitively important, is oftentimes overlooked during the design of control solutions. Maintenance, required to mitigate the effects of system faults, typically necessitates expert personnel not found within normal plant situations ([37]). For this reason, multiple process monitoring algorithms have been developed so that such faults may be automatically detected, diagnosed and eventually removed.

Process monitoring methods may be further classified as i) data-driven ii) analytical and/ or iii) knowledge-based ([13]). The first involves statistical treatment of large quantities of process data and are typified by data-mining and machine learning techniques (such as principal and independent component analysis), statistical control charts and so on. Knowledge-based methods employ qualitative reasoning and are oftentimes rules-based with a strong logic underpinning. A thorough overview of all three classes is presented by [13] and the references therein.

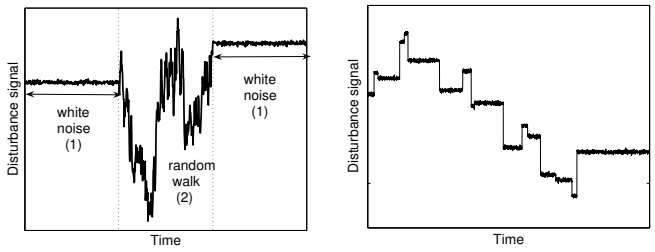
This work, relying on dynamical models of the process for fault detection, is a particular type of analytical approach. Consequently, a necessary standing assumption is the availability of a mathematical model derived from first principles or otherwise. Given the wide-spread popularity of model-based control (such as Model Predictive Control), the controller's model can be readily ported over for the purpose of fault-detection. A model structure, such as in Eqs. (66), (67), is therefore relevant in subsequent developments.

$$\begin{aligned} x_t &= f(x_{t-1}, \theta_{t-1}, u_{t-1}, \omega_t) \\ y_t &= g(x_t, \theta_t, v_t) \end{aligned} \tag{66}$$

$$\begin{aligned} \gamma_t &= \mathbb{A}_t \gamma_{t-1} + \mathbb{B}_t \varphi_t \\ \theta_t &= \mathbb{C}_t \gamma_t + e_t \end{aligned} \tag{67}$$

Here, $x_t \in \mathbb{R}^{n_x}$ represents the state at discrete time index t, $u_t \in \mathbb{R}^{n_u}$, the control input, and $y_t \in \mathbb{R}^{n_y}$, a noise-corrupted measurement signal. $\theta_t \in \mathbb{R}^{n_\theta}$ represents a fault vector with potentially time-varying dynamics governed by matrices $(\mathbb{A}_t, \mathbb{B}_t, \mathbb{C}_t)$ and noise vectors (φ_t, e_t)[1]. ω_t and v_t are process and measurement noise signals respectively. $f(\cdot)$, which may represent an integration of the continuous-time model over a unit sample-time, is the state transition map. Similarly, $g(\cdot)$ represents the state-to-output map.

Faults are typically manifested ([37]) as i) process parameter changes, and/ or ii) disturbance parameter changes, as well as iii) actuator and sensor problems – all captured by θ. Depending on circumstances, these may be sudden jumps (*e.g.* due to an abrupt introduction of significant sensor bias), or slow drifts or random walk-type changes (*e.g.* as a result of catalyst fouling) or even a mixture of both (Figure 14). Such failure modes, which cannot be directly observed, and need to be estimated, are conveniently incorporated into the fault model Eq. (67) by adding the notion of latent states (denoted by r), each of which modifies the fault model (see Eq. (67)) differently. This work explores the use of a Hidden



(a) Intermittent drifts: white noise probabilistically interspersed with integrated white noise.

(b) Abrupt jumps.

Figure 14: Possible disturbance signals (θ).

Markov chain, used previously to model realistic disturbances in the context of process control ([105]), to describe the temporal, probabilistic transitions between the latent states.

[1]In practice, the user would model θ according to disturbance scenarios of interest.

Furthermore, this work can be interpreted as a generalization of the popular approach of assuming statistical independence, from one time period to the next, between hidden states. For example, at each time step t, [104] and [37] allowed the statistics of φ_t and e_t to be described by a Mixture-Of-Gaussians (MOG)[2]. This captures the situation where faults that do occur happen infrequently but with significantly larger magnitudes. Persistent faults like drifts, which are easily described by the proposed Hidden Markov Model (HMM) approach, are captured in the MOG context by introducing additional states or non-linearities in the model.

The main contribution is to show that the aforementioned faults (abrupt jumps/ biases and drifts) can be better modeled and detected by the proposed method. Another novel application is in the context of detecting valve stiction, where it is demonstrated that the output of the valve (which is not normally measured) can be effectively tracked using the same proposed framework.

Section 5.2 provides the details behind an HMM, its subsequent use for fault detection and relevance to prior work. Section 5.3 demonstrates the effectiveness of the proposed method in the context of a heat exchanger. Section 5.4 explores the valve stiction issue before concluding remarks regarding future research are presented in Section 5.5.

5.2 Fault modeling using Hidden Markov Models

HMMs represent a useful class of statistical models where a latent state, taking values from an alphabet $\mathcal{J} \in \{1, 2, \ldots, J \in \mathbb{Z}^+\}$ of cardinality J, transitions probabilistically in a Markovian[3] fashion from one sampling time to the next. Mathematically, a finite-state Markov chain is a sequence of random variables $(r_0, r_1, ..., r_t, ...)$, where the transition probability matrix $\Pi = (\pi_{ij}) = (pr(r_t = j | r_{t-1} = i), i, j \in J) : \sum_{j=1}^{J} \pi_{ij} = 1, \forall i \in \mathcal{J}$, governs the probabilistic temporal transitions. The term 'Hidden' signifies that the actual regime label is usually not known with complete certainty and must be inferred from available noisy measurements of itself or other related states. In the simplest case, each latent state has a

[2] *i.e.,* at each time step, a member from a set of Gaussians, from which the noise signal is to be sampled, is selected with some time-invariant probability.

[3] transitions depend only upon the immediate past.

probability distribution over a finite set of possible output symbols. All Markov chains under consideration are ergodic. For simplicity, the Markov chain is assumed to be at steady state, satisfying $\pi = \Pi'\pi$, where π is a column vector containing the unconditional and initial probabilities of each regime. HMMs have found widespread applications in science and engineering - ranging from speech recognition ([74]) to bioinformatics and diverse fields such as econometrics.

HMMs and their generalizations have been used in fault detection, with significant differences to our proposed approach. [90], for example, did not consider an explicit fault model (*i.e.,* Eq. (67)). Instead, the process parameters are continuously estimated (in batch mode) and treated as output of an underlying Markov chain. This necessitates linking the process parameter vector to fault modes, which is not always possible. A recursive maximum a posteriori filter is then used for fault-mode detection. [33] suggested a similar (see Section 5.2.1) HMM approach to sensor problem diagnosis but limited considerations to faults in the output channels and input signals taking values from a finite, discrete set. [1] learned an HMM corresponding to each operating condition and, unlike the approach proposed in this work, does not make use of the process model. There, fault detection is achieved by a classification scheme that chooses the HMM that maximizes the probability of a given sequence of observations.

5.2.1 Proposed fault model: intermittent drifts & abrupt jumps

Following the successes in other fields, a generalization of Eq. (67) is considered by allowing the statistics of (φ_t, e_t) (and potentially the fault model parameters $(\mathbb{A}, \mathbb{B}, \mathbb{C})$) to vary according to a hidden Markov chain.

Intermittent Drifts. In the case of one-dimensional intermittent drifts (Figure (14a)),

one has:

$$\begin{aligned} \gamma_{t+1} &= \gamma_t + \varphi_{r_{t+1}} \\ \theta_t &= \gamma_t + e_t \\ r_t &\in 1, 2 \\ \pi_{11} &\approx 1,\ \pi_{11} < 1 \\ \pi_{22} &\approx 1,\ \pi_{22} < 1 \end{aligned} \tag{68}$$

Here, φ_{r_t} and e_t are uncorrelated, zero-mean Gaussian signals with covariances (that may depend on r_t) of $Q^{\varphi}_{r_t}$ and Q^{e}_{t}. The abuse of notation on the subscript of φ emphasizes the dependence of the covariance of the noise signal on the underlying Markov chain. When $r_t = 1$ (*i.e.,*, the white-noise regime), $Q^{\varphi}_{r_t=1} \approx 0$. Random-walk type behavior occurs when the hidden state switches to $r_t = 2$, where $Q^{\varphi}_{r_t=2} >> 0$; Q^{e}_{t} is invariant to the hidden regime and of appropriate magnitude. Since it is common that there is low probability of switching once the system enters a particular regime, a diagonally-dominant Π is employed.

Abrupt Jumps. In the case of modeling abrupt jumps, Eq. (68) is adjusted such that $\pi_{11} = \pi_{12} = p \approx 1$, $p < 1$, so that $\Pi = [p,\ 1-p; p,\ 1-p]$. This ensures that the jump state (the second one, in this case) is infrequently accessed and when it is, a significant step-change occurs.

In this latter case, since it is assumed that the Markov chain is at steady state, this form of the transition matrix implies that the probability of entering a particular regime is independent of the current mode. It is thus clear that the HMM framework subsumes an MOG description.

Fault detection and diagnosis is performed via state estimation (in particular to track θ) without the knowledge of the latent state trajectory. Hence, a brief mention of state estimation, based on a model resulting from the concatenation of Eq. (66), and Eq. (67) is necessary.

5.2.2 Fault detection via state estimation of jump Markov systems

Equations (66) and (67) can be merged to yield:

$$\begin{aligned}
\begin{bmatrix} x_{t+1} \\ \gamma_{t+1} \end{bmatrix} &= \mathcal{F}_{r_{t+1}}\Big(\begin{bmatrix} x_t \\ \gamma_t \end{bmatrix}, u_t, \xi_{r_{t+1}} \Big) \\
y_t &= \mathcal{G}_{r_t}\Big(\begin{bmatrix} x_t \\ \gamma_t \end{bmatrix}, n_{r_t} \Big) \\
pr(r_t = j | r_{t-1} = i) &= \pi_{ij}
\end{aligned} \tag{69}$$

Here, $\mathcal{F}$ is implicitly understood to include model structures and parameters from $\{f, \mathbb{A}, \mathbb{B}, \mathbb{C}\}$ and the hidden Markov chain. A similar remark is extended to $\mathcal{G}$. Besides $\mathcal{F}$ and $\mathcal{G}$, the statistics of the noise ξ (a concatenation of (ω, φ, e)) and n (a concatenation of (v, e)) can depend on r. The system represented by Eq. (69) is also termed a Markov jump system. Without knowledge of the sequence $(r_0, \ldots, r_t)$, the optimal filter involves averaging over an exponentially growing number of linear filters. The number of filters scales as J^t, where J is the cardinality of the set containing all possible realizations of r.

The following paragraphs outline the Generalized Pseudo Bayesian estimation algorithm of order 2 (GPB2), a popular sub-optimal method, developed by [4]. The main idea to have trajectories whose last 2 terms differ be merged (via moment-matching) into a single

Gaussian. Using the law of total probability and Bayes' Rule, it can be shown that:

$$
\begin{aligned}
x_{t+1|t+1} &= \sum_{r_{t+1}} p(r_{t+1}|t+1)x_{t+1|(t+1,r_{t+1})} \\
x_{t+1|(t+1,r_{t+1})} &\triangleq \sum_{r_t} x_{t+1|(t+1,r_{t+1},r_t)}p(r_t|r_{t+1},t+1) \\
P_{t+1|t+1} &= \sum_{r_{t+1}}\{(x_{t+1|t+1}-x_{t+1|(t+1,r_{t+1})})(\cdot)' \\
&\quad +P_{t+1|t+1,r_{t+1}}\}p(r_{t+1}|t+1) \\
P_{t+1|t+1,r_{t+1}} &= \sum_{r_t}\{(x_{t+1|t+1,r_{t+1}}-x_{t+1|(t+1,r_{t+1},r_t)})(\cdot)' \\
&\quad +P_{t+1|t+1,r_{t+1},r_t}\}p(r_t|r_{t+1},t+1) \\
p(r_t|r_{t+1},t+1) &= \frac{1}{c_1}p(y_{t+1}|t,r_{t+1},r_t)p(r_t|r_t)p(r_t|t)
\end{aligned}
$$

$$
p(r_{t+1}|t+1) = \frac{1}{c_2}\sum_{r_t} p(y_{t+1}|t,r_{t+1},r_t)p(r_{t+1}|r_t)p(r_t|t)
$$

The term $p(y_{t+1}|t,r_{t+1},r_t)$ refers to the probability density of the corresponding one-step ahead output prediction. $x_{t+1|(t+1,r_{t+1})}$ refers to the estimate of x_{t+1} given output measurements $\{y_0,\ldots,y_{t+1}\}$ and a certain realization of r_{t+1}; $P_{t+1|(t+1,r_{t+1})}$ denotes the corresponding error covariance matrix. The pair $(x_{t+1|(t+1,r_{t+1},r_t)}, P_{t+1|(t+1,r_{t+1},r_t)})$ are similarly defined. It is noted that starting from $(x_{t|(t,r_t)}, P_{t|(t,r_t)})$, a single application of the time and measurement update steps of the (extended) Kalman filter yields these latter quantities. c_1 and c_2 are normalizing constants such that the merging probabilities $p(r_t|r_{t+1},t+1)$ and $p(r_{t+1}|,t+1)$ sum to unity.

5.2.3 A-posteriori regime estimation

If required, a prediction and/ or filtered estimate of the hidden regime can be obtained viz:

$$
\begin{aligned}
\hat{r}_{t+1|t} &= \arg\max_{r_{t+1}}\left\{p(r_{t+1}|t) \triangleq \sum_{r_t} pr(r_{t+1}|r_t)\cdot pr(r_t|t)\right\} \\
\hat{r}_{t|t} &= \arg\max_{r_t}\{p(r_t|t)\} \qquad (70)
\end{aligned}
$$

5.3 Example 1: Fault tracking in a shell & tube heat exchanger

In this example, the usefulness of the proposed method in detecting faults is studied in the context of a shell and tube heat exchanger Eq. (71) also considered by [37]. In particular, we contrast the proposed HMM approach against an MOG method ([37]) in modeling the latent states that govern the fault signals (see Section 5.3.1 for simulation details). The main difference is that the latter framework assumes that each latent state occurs with a (time-invariant) probability that is independent of the previous realization. The governing non-linear ordinary differential equations used for simulation but not estimator design, are:

$$\begin{aligned} \frac{dT_c}{dt} &= \frac{q_c}{V_c}(T_{ci} - T_c) + \frac{\alpha_c}{V_c}(T_h - T_c) \\ \frac{dT_h}{dt} &= \frac{q_h}{V_h}(T_{hi} - T_h) - \frac{\alpha_h}{V_h}(T_h - T_c) \\ y &= \begin{pmatrix} T_c \\ T_h \end{pmatrix} + \mu_v + v \end{aligned} \tag{71}$$

Here, the measured state variables are the temperatures of the hot and cold streams respectively: $[T_c; T_h]$. $[T_{ci}; T_{hi}]$ are the temperatures of the incoming cold and hot streams respectively. $[\alpha_c; \alpha_h]$ are system parameters reflecting the heat transfer coefficient, heat transfer area, density, specific heat capacity of the cold and hot streams respectively. Similarly, $[q_h; q_c]$ are the flow rates of the hot and cold streams and represent the degrees of freedom available to a controller. $[V_c; V_h]$ are the volumes of the cold and hot sides. Steady-state values are reported in Table 8. v refers to zero-mean measurement noise of covariance $R \triangleq \mathbb{E}[vv']$. μ_v is nominally a null vector but might be subject to changes due to disturbances.

5.3.1 Simulation conditions

Although a variety of fault types may be considered (*e.g.* those affecting the various input and output channels and/ or changes in parameters (α_c, α_h), as discussed in Section 5.1), for clarity of exposition, only two different fault types are assumed. Furthermore, these affect only the cold side. Given initially quiescent conditions (see Table 8), one considers:

1. An abrupt step that is normally distributed with zero mean [L/min] and variance q_u^{hi}

$[\mathrm{L}^2/\mathrm{min}^2]$ affecting the input channel on the cold side (q_c) at some unknown time t_u. This may be thought of as a sudden bias developing in the input channel:

$$q_{ct} \quad = \quad q_{ct-1} + \varphi_t^u \cdot \delta(t, t_u),\ \varphi_t^u \sim \mathcal{N}(0, q_u^{hi}) \tag{72}$$

$\delta(\cdot,\cdot)$ is the Dirac delta function. q_u^{hi} has a value of 2 in the following experiments.

2. A sudden drift (see Figure 14a) affecting the sensor relaying T_c (*i.e.,* y_1) measurements between an unknown time span: $\mathcal{T} \triangleq [t_{y,1}, t_{y,2}]$. Namely, one has:

$$\mu_{v,1_t} \quad = \quad \mu_{v,1_{t-1}} + \varphi_t^y \tag{73}$$

where $\mathbb{E}[\varphi_t \varphi_t'] = q_y^{hi} = 0.5$ if $t \in \mathcal{T}$ and $\mathbb{E}[\varphi_t \varphi_t'] = q_y^{lo} = 10^{-10} \approx 0$, for other time periods. $\mu_{v,2}$ remains at the origin for all time.

The above non-linear model is not available for state estimation. Instead, a version linearized about the nominal operating conditions is available. With a sampling time of 0.5 min, $A = [0.91,\ 0.03; 0.03, 0.91]$, $B = [\text{-}0.12, 0.002; \text{-}0.002, 0.12]$, $C = diag([1,\ 1])$. Measurement covariance, R, is set to $diag([0.5,\ 0.5])$ and known. Since estimation is the focus of this example, the system is run in the absence of feedback control.

Table 8: Nominal steady state operating conditions

Variable	Value	Units
$q_c^* = q_h^*$	10	L/min
T_{ci}^*	25	^{o}C
T_{hi}^*	100	^{o}C
T_c^*	43.75	^{o}C
T_h^*	81.25	^{o}C
α_c^*	5	m^3/min
α_h^*	5	m^3/min
$V_c^* = V_h^*$	75	L

5.3.2 Proposed HMM method to handle abrupt jumps & intermittent drifts

The following Markov jump linear model, a specialization of Eq. (69), is employed:

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + b\theta^u_t + \omega_{t+1} \\ \theta^u_{t+1} &= \theta^u_t + \varphi^u_{r_{t+1}} \\ \theta^y_{t+1} &= \theta^y_t + \varphi^y_{r_{t+1}} \\ y_t &= Cx_t + \theta^y_t + v_t \end{aligned} \tag{74}$$

where x_t, the state variable at discrete time index t are deviations from $[T^*_c; T^*_h]$. Similarly, the vector $u_t \in \mathbb{R}^2$ represents deviations from $[q^*_c; q^*_h]$. b represents the first column of matrix B, consistent with the fact that disturbances enter the q_c channel. $[\theta^u; \theta^y]$ are input and output disturbance state variables respectively. Both θ^u and θ^y are modeled as integrators but distinguished by the effects of the hidden Markov regime on the second moments of φ^u and φ^y. Consistent with the assumption of an abrupt jump, the covariance of φ^u is assumed to be large with a small probability, and vice versa. θ^y is naturally modeled as an intermittent drift (see Eq. (68)). Details are given in the following paragraphs.

A four-regime Markov chain is considered. These regimes represent the following scenarios:

1. No disturbance in input channel, No disturbance in output channel ('LO-LO')
2. No disturbance in input channel, Drifting disturbance in output channel ('LO-HI')
3. Abrupt disturbance in input channel, No disturbance in output channel ('HI-LO')
4. Abrupt disturbance in input channel, Drifting disturbance in output channel ('HI-HI')

Accordingly, a simple method for determining the values of the transition probability matrix (Π) is proposed. Per the earlier discussion (Section 5.2.1), two (sub) transition probability matrices are appropriate for the input (Π^u) and output channels (Π^y) respectively, the first state being the 'normal' regime in both cases.

$$\Pi^u = \begin{pmatrix} 0.99 & 0.01 \\ 0.99 & 0.01 \end{pmatrix};\ \Pi^y = \begin{pmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{pmatrix} \tag{75}$$

An overall transition probability matrix (Π) accounting for the four scenarios can be obtained by assuming statistical independence between the input and output channels. For example in computing π_{23}, one has transitions between the 'normal' to 'abnormal' state for the input channel and the opposite transitions for the output channels so that

$$\pi_{23} = \pi_{21}^u \pi_{12}^y \tag{76}$$

The overall Π^4 is:

$$\begin{pmatrix} 0.98 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.98 & 0.01 & 0.01 \\ 0.98 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.98 & 0.01 & 0.01 \end{pmatrix}$$

In accordance to the noise statistics of the possible fault scenarios, the covariance of the overall noise vector $\xi_t \triangleq [\omega_t,\, \varphi_t^u,\, \varphi_t^y]$ for the 4 regimes are:

1. 'LO-LO': $\mathbb{E}[\xi_t \xi_t'] = diag([10^{-10},\, 10^{-10},\, 10^{-10},\, q_y^{lo}])$
2. 'LO-HI': $\mathbb{E}[\xi_t \xi_t'] = diag([10^{-10},\, 10^{-10},\, 10^{-10},\, q_y^{hi}]$
3. 'HI-LO': $\mathbb{E}[\xi_t \xi_t'] = diag([10^{-10},\, 10^{-10},\, q_u^{hi},\, q_y^{lo}]$
4. 'HI-HI': $\mathbb{E}[\xi_t \xi_t'] = diag([10^{-10},\, 10^{-10},\, q_u^{hi},\, q_y^{hi}]$

Process noise ω is negligible compared to θ^u and will be assumed to be absent for simplicity.

5.3.3 Alternative MOG description

If one were to be restricted to an MOG description of the latent regime, then an additional state (θ^β) is required:

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + b\theta_t^u + \omega_{t+1} \\ \theta_{t+1}^u &= \theta_t^u + \varphi_{t+1}^u \\ \theta_{t+1}^\beta &= \theta_t^\beta + \varphi_{t+1}^\beta \\ \theta_{t+1}^y &= \theta_t^y + \theta_t^\beta \\ y_t &= Cx_t + \theta_t^y + v_t \end{aligned} \tag{77}$$

[4] the rows do not sum to unity due to rounding errors

Similar to Eq. (74), θ^u refers to the input channel disturbance and is modeled as an abrupt jump. However, the output disturbance (θ^y) is now modeled as a double integrator (driven by θ^β). θ^β itself may be interpreted as a velocity term and is driven by φ^β which is set to have a small covariance (10^{-10}) with large probability and a large covariance (of q_y^{hi}) with small probability. This captures the (rare) event of a velocity change when the output disturbance transitions from the white-noise regime to the random-walk mode and vice versa (see Figure 14(a)). In this case, the sub transition matrices for the input and output channels are:

$$\Pi^u = \Pi^y = \begin{pmatrix} 0.99 & 0.01 \\ 0.99 & 0.01 \end{pmatrix}$$

The overall transition matrix may be obtained as before, per Eq. (76). The covariance of the overall noise vector $\xi_t \triangleq [\omega_t, \varphi_t^u, \varphi^\beta, \varphi_t^y]$ for the 4 regimes are:

1. $\mathbb{E}[\xi_t \xi_t'] = diag([10^{-10}, 10^{-10}, 10^{-10}, q_y^{lo}, 10^{-10}])$
2. $\mathbb{E}[\xi_t \xi_t'] = diag([10^{-10}, 10^{-10}, 10^{-10}, q_y^{hi}, 10^{-10}]$
3. $\mathbb{E}[\xi_t \xi_t'] = diag([10^{-10}, 10^{-10}, q_u^{hi}, q_y^{lo}, 10^{-10}]$
4. $\mathbb{E}[\xi_t \xi_t'] = diag([10^{-10}, 10^{-10}, q_u^{hi}, q_y^{hi}, 10^{-10}]$

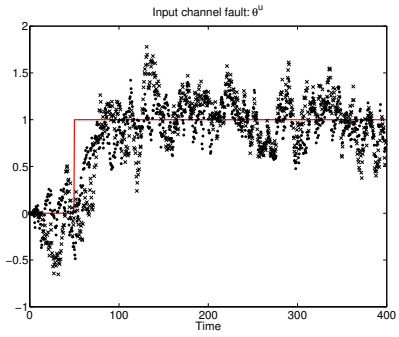
5.3.4 Example 1: Results

Table 9 presents a summary (average over 100 realizations) of the state-estimation error for both the input and output channel. A typical realization is depicted in Figure 15.

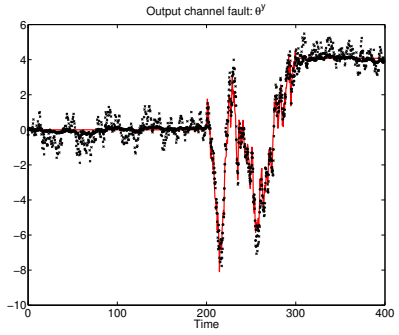
Table 9: 2-norm of state-estimation error (Average of 100 realizations)

Channel	Proposed see Eq. (74)	MOG approach see Eq. (77)
Input	11.4	12.9
Output	13.3	19.7

Due to the similarities in modeling the abrupt jump in the output channel, it can be seen from Figure 15(a) and the first line of Table 9 that the performance of the state estimator corresponding to both approaches yield similar performances. However, the MOG approach



(a) Tracking θ^u



(b) Tracking θ^y

Figure 15: Tracking θ^u and θ^y. Comparing the proposed HMM vs. MOG approaches. Legend: solid line - actual fault signal; Dots ($\cdot$) - HMM; Crosses (x) - MOG

fares significantly worse than the proposed HMM approach in tracking the fault signal (which is an intermittent drift) corresponding to the output channel (see Figure 15(b) and the second row of Table 9).

5.4 Example 2: Valve stiction

Valve stiction is a common problem in control valves, the latter being widely used in process industries ([15]). Due to the effects of friction, the output (u^x) of the control valve does not track its input (u^c) (*i.e.,* the control signal prescribed by the controller) instantaneously. Instead, u^x has been observed to demonstrate a delayed and sluggish response to u^c, where the valve 'sticks' to its current position if changes in the control signal (and/ or the absolute magnitude itself) are insufficiently large to overcome friction effects. This is usually to the detriment of closed-loop performance.

It is assumed that the plant is linear and therefore parameterized by matrices (A, B, C), where A is the state-transition map, B, the input-to-state map and C, the state-to-output map. Technical definitions, first-principles and empirical models of stiction can be found in the articles by [15, 14] and the references therein. For simplicity, an efficient single-parameter model employed by [94] and [93] for stiction detection is used for simulations in the sequel:

$$u_t^x = \begin{cases} u_{t-1}^x, \text{ if } |u_t^c - u_{t-1}^x| \leq d \\ u_t^c, \text{ otherwise} \end{cases} \tag{78}$$

where d represents the valve stiction band. The larger the value of d, the more severe the stiction problem.

The detection, diagnosis and compensation-for valve stiction has received much attention in academia and industry. Based on Eq. (78), [94] proposed a suitable model for detecting stiction:

$$u_t^x = \tilde{\delta}_t \cdot u_{t-1}^x + (1 - \tilde{\delta}_t) \cdot u_t^c$$

where $\tilde{\delta}_t$ is a binary (0/1) mode parameter occurring with a certain (i.i.d) probability.

For the same purpose of stiction detection and estimating the typically unmeasured

u^x_t, we allow $\tilde{\delta}_t$ to have statistics governed by an underlying Markov chain so that observations reflecting persistent 'stickiness' can be more effectively modeled. Also, instead of identifying the segmentation sequence $\{\tilde{\delta}_1, \ldots, \tilde{\delta}_t\}$ that maximizes the posterior quantity $pr(\tilde{\delta}_1, \ldots, \tilde{\delta}_t | y_1, \ldots, y_t)$ through dynamic programming, we propose a novel Markov jump linear description that is consistent with Eq. (78) to be used by a GPB2 state-estimator:

$$\begin{pmatrix} x_t \\ u^x_{t-1} \end{pmatrix} = \begin{pmatrix} A & B^x_{r_{t-1}} \\ 0 & \tilde{\delta}_{r_{t-1}} \end{pmatrix} \begin{pmatrix} x_{t-1} \\ u^x_{t-2} \end{pmatrix} + \begin{pmatrix} B^c_{r_{t-1}} \\ 1 - \tilde{\delta}_{r_{t-1}} \end{pmatrix} u^c_{t-1}$$

$$y_t = \begin{pmatrix} C & 0 \end{pmatrix} \begin{pmatrix} x_t \\ u^x_{t-1} \end{pmatrix} + v_t \tag{79}$$

When $r = 1$, stiction is absent, $\tilde{\delta} = 0$, $B^x = 0$, $B^c = B$. Conversely, when $r = 2$, stiction is present, $\tilde{\delta} = 1$, $B^x = B$, $B^c = 0$.

5.4.1 Simulation studies: mixing tank

For simulation studies, we consider a simple isothermal mixing-tank (of cross-sectional area $\mathcal{A}$) with an outlet stream whose flow-rate is controlled by a valve (with resistance $\mathcal{R}$):

$$\frac{dm}{dt} = \frac{1}{\mathcal{A}}(q_1 + q_2 - \frac{m}{\mathcal{R}}) \tag{80}$$

The controlled (and also measured) variable is the liquid level (m). The flow-rate of the first stream, q_1, is a measured disturbance whereas that of the other stream (q_2) represents the manipulated variable. A PI controller (with gain K_c, and integral time constant τ_I) is given by:

$$u^c_t = u^c_{t-1} + K_c[e_t - e_{t-1} + \frac{h}{\tau_I} e_t],\ e_t \triangleq l - y_t$$

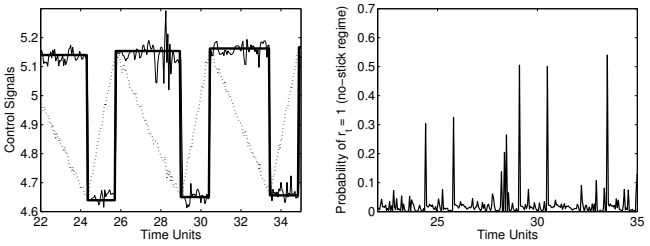
Here l is the set-point, nominally calibrated to a value of 6. For ease, $\mathcal{A}$, $\mathcal{R}$, K_c, τ_I and the measured disturbance signal q_1, are all set to nominal values of 1. A relatively large value for the stiction band is employed: $d = 0.5$. A sampling time of $h = 0.05$ is employed, resulting in the following parametrization to be used by the state estimator: $A = 0.951$, $B = 0.0488$ and $C = 1$. Measurement noise is set to have a known covariance of $R \triangleq \mathbb{E}[v_t v_t'] = 10^{-4}$.

To reflect the high degree of stiction, the transition probability matrix Π is:

$$\begin{pmatrix} 0.01 & 0.99 \\ 0.01 & 0.99 \end{pmatrix}$$

5.4.2 Results: estimating valve output & detecting valve stiction

Tracking results for a typical closed-loop realization are shown in Figure 16. The existence of the cycles in u^c and u^x (Figure 16(a)) is due to the presence of integral action as well as the valve stiction phenomenon. From Figure 16(a), it can be seen that the proposed methodology is able to estimate u^x. Observing the (a-posteriori) probability (see Eq. (70) and Figure 16(b)) of the first mode (or equivalently the second) via reveals the time instances where a switch occurs (by means of the probability peaks). Doing so represents an effective way for detecting stiction.



(a) Time series plot of valve input and output and estimate of valve output. Legend: dotted: u^c; bold line: u^x; line: $\hat{u}^x$

(b) Probability of mode 1

Figure 16: Tracking unmeasured valve output in mixing-tank example.

5.5 Conclusions & future work

The main contribution of this work is to show that the common faults (abrupt jumps/ biases and drifts) can be better modeled and detected by the proposed HMM-based method. Another novel application is in the context of detecting valve stiction, where it is demonstrated that the output of the valve (which is not normally measured) can be effectively estimated. Future work involves extending the problem to large scale systems (*e.g.* a network of unit operations) of industrial interest.

CHAPTER VI

CONTROLLING JUMP STOCHASTIC SYSTEMS VIA POST-DECISION-STATE-BASED APPROXIMATE DYNAMIC PROGRAMMING

6.1 Introduction

Classical Advanced Process Control (APC) methods do not handle uncertainty in a systematic, closed-loop optimal fashion. This is typified by the approach adopted by Model Predictive Control (MPC), the defacto APC solution. MPC is now considered to be a mature technology owing to the plethora of research and industrial experiences during the past three decades. It is a technique in which the current control action is obtained by minimizing online, a cost criterion defined on a finite time interval. Nominal, deterministic trajectories of future disturbance signals and uncertainties are necessarily assumed in order to obtain an optimization problem amenable to on-line solution via math programming. The solution generates a control sequence from which the first element is extracted and implemented. The procedure is repeated at the next time instant. MPC's ability to handle constrained, multi-variable control problems in an open-loop optimal manner is responsible for its popularity.

MPC's open-loop optimal control formulation used to find the control moves at each sample time means the fact that information about future uncertainty will be revealed, this being generally beneficial for control performance, is not considered. A-priori, deterministic assumptions about future uncertainty results in poor or unpredictable performance in a number of situations. For example, the authors of [81] have demonstrated that the optimal stationary point for the minimal energy control of an inverted pendulum subject to zero-mean stochastic forcings is not the inverted, upright position, as one might conclude in the deterministic case.

It is often cited that the optimal operating point of a unit operation lies close to/ on

the constraint boundaries [51, 2]. Since noise is not accounted for explicitly, MPC tends to place the system trajectory next to a constraint boundary [81]. As a result, excursions into the infeasible region occur in the event of disturbances. Adjustments to the set-point so as to back-off [56, 54] from the constraint boundaries are therefore common. Another noteworthy scenario is the case of unstable systems, such as those with integrators driven by white noise. It is noted that the latter are particularly common in chemical processes. In the case of open-loop control (*e.g.*, in MPC), it can be easily demonstrated that the variance of the prediction error is unbounded with prediction horizon length.

Yet another situation where uncertainty has a vital role to play is that of dual control [21, 22, 23, 24]. In the context of dual control, control actions significantly affect the propagation of future uncertainty. In this case, the control action serves the antagonistic functions of probing the system to obtain information regarding unknown system parameters and set-point tracking. Deterministic formulations do not achieve such "active learning". In a similar fashion, due consideration needs to be given to the interplay between control and state estimation [32]. The latter is oftentimes necessary when full state feedback is unavailable. Although a state estimator (such as the Extended Kalman Filter) generally returns a distribution, most control solutions employ only a point estimate. The implicit assumption of the manipulated variable not affecting the quality of the state estimates might result in poor closed-loop performance. Similar to the dual control case, the optimal policy involves probing so as to reduce the error covariance. The authors of [32] proposed a modification of the optimization criterion used in MPC for the control of nonlinear systems, where separation does not usually hold.

Most of the past attempts at ameliorating the impact of uncertainty have been reflected in robust MPCs formulations based on the objective of minimizing the worst-case scenarios [86] at the expense of overly conservative policies and the assumption that the disturbance signals be bounded. Multi-scenario formulations [43] have also been developed but the number of scenarios is limited and they do not give closed-loop optimal policies in general. Stochastic programming based methodologies [70] allow for recourse actions at the computational expense of enumerating an exponentially growing number of scenarios.

6.1.1 Dynamic programming for solving multi-stage stochastic control problems

In light of the above, Dynamic Programming (DP) [7] represents a unified framework for solving multi-stage, stochastic control problems. Underlying DP is the "cost-to-go" function that maps a state to a real value that quantifies the state's desirability. Intuitively, at any state (x), the optimal policy would endeavor to minimize the amount of effort in bringing the system to a next state that is as desirable as possible. It is noted that DP involves solving a single-stage problem online as compared to MPC which involves in computationally exorbitant multi-stage optimizations. Unfortunately, traditional DP suffers from a computational "curse of dimensionality", the idea being that obtaining the value function for every single state of the (potentially continuous) state space involves computational requirements that scale exponentially with the size of the problem. To circumvent this, an approach called approximate dynamic programming (ADP), a technique that urfaced from the research on reinforcement learning in the Artificial Intelligence (AI) community [95, 8]), has been successfully used. The main ideas revolve around the intelligent sampling of the state space through simulations and an appropriately designed function approximator. ADP involves iterative off-line computations to yield the cost-to-go; these values are then used on-line for control. Both off-line and on-line calculations involve single-stage optimizations. The authors of [47, 48, 44, 46, 49, 50] have specialized ADP methods for solving process control problems. However, the focus has primarily been on deterministic control problems. A reason is that when applied to stochastic control problems, the on-line and off-line optimizations within the DP/ ADP framework involve a minimization (or maximization) over an expected value, the computation of which is generally cumbersome.

In this work, ADP based on an intermediate post-decision state variable, first introduced by [80] and employed extensively by [71] in solving operations research problems, is used to allow for more computationally effective strategies. The main benefit from the introduction of the post-decision state allows the generally non-commutative optimization (e.g. min) and expectation ($\mathbb{E}$) operators to be interchanged. The main advantage is that the off-line ADP computations may then be run in parallel using off-the-shelf solvers. It is noted that

the latter have been the cornerstone of MPC technology.

The rest of the chapter is organized as follows. Section 6.2 presents the idea of Dynamic programming based on the post-and-pre decision state variables. Section 6.3 outlines the proposed *Approximate* DP framework based on the post-decision state. A more thorough explanation of the proposed ADP framework as applied to stochastic jump systems is then given in Section 6.4. Examples highlighting the importance of accounting for uncertainty systematically, in the context of general stochastic systems, are shown in Section 6.5.

6.2 Pre and post-decision-state-based dynamic programming

Consider the optimal control of the following discrete-time stochastic system:

$$x_{t+1} \quad = \quad f(x_t, u_t, \omega_t) \tag{81}$$

where $x_t \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ refers to the system state at discrete time index t, $u_t \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ a control or action vector, and ω_t an exogenous, unmeasured, stochastic signal. x may contain physically meaningful states as well as measured disturbances, and parameters subject to uncertainty. f refers to the single-stage transition function. For problems where the system's dynamics are represented by ordinary differential equations, f is then the result of numerical integration across a single sample-time, with vectors u and ω held constant. Throughout this chapter, it is assumed that full state feedback is available. In the event that only output feedback is available, x is interpreted as an information vector that contains the sufficient statistics of the state estimate's probability density function. Such lifting is possible as the information vector is governed by another related set of equations (*i.e.*, the filter dynamics).

Let $\mu \in \Gamma$ be a 'state-feedback policy' that maps the state vector to the action vector, where Γ represents the set of all admissible (stationary) such policies. $J^{\mu}(x)$ will be used to denote the 'cost-to-go' function, which is defined as the infinite horizon, discounted sum of the stage-wise costs under the policy μ starting from an arbitrary state x:

$$J^{\mu}(x) \quad = \quad \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \phi(x_k, u_k = \mu(x_k)) | x_0 = x\right] \tag{82}$$

where ϕ represents a pre-specified stage-wise cost (*e.g.* $\phi(x,u) := ||x||_Q^2 + ||x||_R^2$) and $\gamma \in [0, 1)$ is a discount factor. The goal then is to find the optimal (stationary) policy

$\mu^* : \mathcal{X} \rightarrow \mathcal{U}$, that yields the minimum cost-to-go function as below:

$$J^{\mu^*}(x) = \min_{\mu \in \Gamma} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \phi(x_{t+k}, u_{t+k} = \mu(x_{t+k}))|x_t = x\right] \tag{83}$$

$J^{\mu^*} : \mathcal{X} \rightarrow \mathbb{R}^{0+}$ is the *optimal* 'cost-to-go' function and is an indication of the attractiveness of a given state in terms of future rewards. By definition, $J^{\mu^*}(x) \leq J^{\mu}(x), \forall x$ and $\forall \mu \in \Gamma$.

Based on the principle of optimality [7], one is able to re-write Eq. (83), thereby obtaining Bellman's optimality equations:

$$\begin{aligned} J^{\mu^*}(x) &= \min_{u \in \mathcal{U}} \left\{\phi(x,u) + \gamma \mathbb{E}_{(\omega|x)}[J^{\mu^*}(f(x,u,\omega))]\right\} \\ &= \left(TJ^{\mu^*}\right)(x) \end{aligned} \tag{84}$$

T above represents the single-pass DP operator represented by the minimization operation. The optimal policy is implicitly obtained through the solution of the associated single-stage optimization:

$$\mu^*(x) = \arg\min_{u \in \mathcal{U}} \left\{\phi(x,u) + \gamma \mathbb{E}_{(\omega|x)}[J^{\mu^*}(f(x,u,\omega))]\right\} \tag{85}$$

In principle, the optimal control problem is solved once J^{μ^*} is known. The repeated application of T on an arbitrarily initialized cost-to-go leads to convergence and underpins the idea behind Value Iteration (VI).

$$J^{\mu^*}(x) = TJ^{\mu^*}(x) = \lim_{i \rightarrow \infty} (T)^i J^{\mu}(x), \ \forall \mu, x \tag{86}$$

Note that Eqs. (84)-(85) involve a computationally cumbersome minimization over an expected quantity whose probability distribution function is generally unknown. The post-decision state, $x^p \in \mathcal{X}^p \subseteq \mathbb{R}^{n_x}$, is introduced to alleviate this problem. x^p refers to the system state immediately after the control vector is introduced to the system but before the uncertainty is realized. As a result, f may be decomposed into the following sub-transitions:

$$x_t^p = f_1(x_t, u_t) \tag{87}$$

$$x_{t+1} = f_2(x_t^p, \omega_t) \tag{88}$$

where the composition of f_1 and f_2 is equivalent, in effect, to f, in Eq. (81). Note that f_1 describes a deterministic transition between the pre-decision state variable (x) and x^p. f_2 involves the transition due to uncertainty after the control action is implemented. Consequently, the value function of x^p, $J^{\mu,p}(x^p)$, may be expressed in terms of the value function of x, as such:

$$J^{\mu,p}(x_t^p) \quad = \quad \mathbb{E}_{(\omega|x_t^p)}\left[J^{\mu}(x_{t+1})\right], \forall \mu \tag{89}$$

By considering the optimal policy μ^*, and substituting Eq. (84) into Eq. (89), the min and $\mathbb{E}$ operators are interchanged, yielding:

$$J^{\mu^*,p}(x_t^p) \quad = \quad \mathbb{E}_{(\omega|x_t^p)}\left[\min_{u_{t+1}\in\mathcal{U}}\left\{\phi(x_{t+1},u_{t+1})+\gamma J^{\mu^*,p}(x_{t+1}^p)\right\}\right] \tag{90}$$

The single-stage on-line optimization is also streamlined:

$$\mu^*(x) \quad = \quad \arg\min_{u\in\mathcal{U}}\left\{\phi(x,u)+\gamma J^{\mu^*,p}(f_1(x,u))\right\} \tag{91}$$

The post-decision analogue for the cost-to-go value function of an arbitrary policy is also readily written:

$$J^{\mu,p}(x_t^p) \quad = \quad \mathbb{E}_{(\omega|x_t^p)}\left[\phi(x_{t+1},\mu(x_{t+1}))+\gamma J^{\mu,p}(x_{t+1}^p)\right] \tag{92}$$

As with the situation with the pre-decision state, the optimal control problem is solved once $J^{\mu^*,p}$ is known. The post-decision state DP operator, T^p, may be similarly defined so that VI (based on x^p) converges to $J^{\mu^*,p}$, as such:

$$J^{\mu^*,p}(x^p) = T^p J^{\mu^*,p}(x^p) = \lim_{i\to\infty}(T^p)^i J^{\mu,p}(x^p),\ \forall \mu, x^p \tag{93}$$

Convergence (see Eq. (86) and Eq. (93)) to a unique point is guaranteed due to the fact that both T and T^p are γ-contraction maps (see Section 6.3.1). The introduction of the post-decision state allows the generally non-commutative min and $\mathbb{E}$ operators to be interchanged. Eq. (90), used off-line during value iteration, consists of an independent collection of deterministic optimization problems, which may be run in parallel using off-the shelf solvers.

In process control problems, due to the continuous nature of the state and action spaces which must be discretized, numerical solutions become quickly bottle-necked as the problem dimensions grow. In fact, the growth would be exponential as the number of discretized points grows with the dimension as such. Hence, a naive application of VI in this case is computationally prohibitive and the 'curse-of-dimensionality' is even more apparent in continuous problems. For problems with continuous state and action space, one needs to resort to approximations that involve an intelligent state-sampling/ discretization scheme and/ or an efficient representation of the cost-to-go [44, 71]. We specialize discussions to the case involving the post-decision state (x^p) in the following section.

6.3 Approximate dynamic programming based on the post-decision state

Value iteration can work with only finite state space. For systems with continuous state and action space, one must then work with discretized state space, either through gridding, or preferably, sampling. It is often the case that only a small portion of $\mathcal{X}$, $\mathcal{X}^p$ and $\mathcal{U}$ will ever be visited under optimal and/ or high-quality sub-optimal policies. Let us denote the subset of the pre-decision and post-decision state spaces that are 'relevant', *i.e.*, visited with non-trivial probability under the optimal control, as $\mathcal{X}^*_{REL}$ and $\mathcal{X}^{*,p}_{REL}$ respectively. Such sets would be continuous but much smaller-sized than $\mathcal{X}$ and $\mathcal{X}^p$ in general. The key notion is that if one could identify $\mathcal{X}^*_{REL}$ and $\mathcal{X}^{*,p}_{REL}$ or a parsimonious superset of them, one can sample the sets with sufficient density to perform the dynamic programming at significantly reduced computation. Of course, the difficulty is that it is not easy to obtain such a set ahead of time without knowing the optimal controller itself.

We expand the ADP approach proposed by the authors of [47, 44, 98] for the purpose of process control applications to a post-decision-state based Approximate Dynamic Programming scheme. The main idea is to employ carefully designed simulation schemes for the sampling of the state space and function approximation (for the purpose of cost-to-go interpolation) to this end. We have the following off-line computations:

1. Identify a finite-sized, 'relevant' pre-decision state-space, $X_{sam} \subset \mathcal{X}$ and its corresponding post-decision counterpart, X^p_{sam}. Note that each (x, x^p) tuple are related

through Eq. (87), where x^p is defined to be the state variable obtained immediately after an action is taken but before the realization of any uncertainties. Also, the number of elements in X_{sam} and X^p_{sam} are equal, that is, $|X_{sam}| = |X^p_{sam}| = N$. Identification of these finite sets is achieved, for instance, by simulating all possible combinations of an initial sub-optimal policy, $\mu_{[0]}$, and operating conditions. The latter are defined as all starting states of interest (for servo problems) as well as potential values of measured disturbance values. Dithering may also be introduced for the purpose of exploration.

2. Assign a cost-to-go for all elements of X^p_{sam}. The initial, finite-sized 'cost-to-go' table, denoted by $\mathcal{T}^p_{[0]} \triangleq \{x^p, \hat{J}^{\mu^*,p}_{[0]}(x^p) \,|\, x^p \in X^p_{sam}\}$, is thus obtained. The symbol $\hat{(\cdot)}$ is used to emphasize the approximate nature of the cost-to-go sequence, even in the limit of the iterative process. Exact initialization is not critical per se since the fixed point derived from the following step is unique. In the sequel, initialization proceeds as follows:

$$\hat{J}^{\mu^*,p}_{[0]}(x^p) \quad = \quad \mathbb{E}_{(\omega|x^p)}[\hat{J}^{\mu^*}_{[0]}(f_2(x^p,\omega))] \tag{94}$$

where $\hat{J}^{\mu^*}_{[0]}(x) = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^t \phi(x_k, \mu_{[0]}(x_k)) \big| x_0 = x]$. For practical implementation, an exact computation of the infinite sum may be approximated by a summation up to a finite length ($\tilde{t}$) that corresponds to $\gamma^{\tilde{t}} \leq \tau$, where τ is a sufficiently small number. When there are a number of candidate initial, sub-optimal policies, the most promising one may be used for initialization of the value function.

3. Obtain converged cost-to-go values for X^p_{sam} through VI, yielding the sequence of value tables $\{\mathcal{T}^p_{[0]}, \mathcal{T}^p_{[1]}, \ldots\}$. Since the VI requires the evaluation of the cost-to-go function for post-decision states (x^p) not necessarily in X^p_{sam}, a well-designed function approximator is needed to interpolate among the stored values (see discussion in Section 6.3.1). A certain choice of function approximator ensures that each pass of the iteration is a *contraction-map* with a unique fixed point (see Section 6.3.1). In

other words, each step of the modified VI involves:

$$\hat{J}^{\mu^*,p}_{[i+1]}(x^p) = \left(T^p F(\hat{J}^{\mu^*,p}_{[i]})\right)(x^p),\ \forall x^p \in X^p_{sam} \tag{95}$$

Here $F(\hat{J}^{\mu^*,p}_{[i]})$ denotes the cost-to-go function approximator based on the stored values from $\mathcal{T}_{[i]}$. Termination occurs when $\|\hat{J}^{\mu^*,p}_{[i+1]} - \hat{J}^{\mu^*,p}_{[i]}\|_\infty$ is less than a pre-defined tolerance. Alternatively, the relative measure $\|\frac{1}{\hat{J}^{\mu^*,p}_{[i]}}(\hat{J}^{\mu^*,p}_{[i+1]} - \hat{J}^{\mu^*,p}_{[i]})\|_\infty$ may be used for cases where the cost-to-go values are large.

4. Return to step 1, since the relevant domain of the state-space may not be properly ascertained a-priori. If this is the case, an initial policy for the next round of value iteration would be that instructed by the current cost-to-go table. Otherwise, use the converged values for online control.

Figure 17 is a schematic of the proposed algorithm.

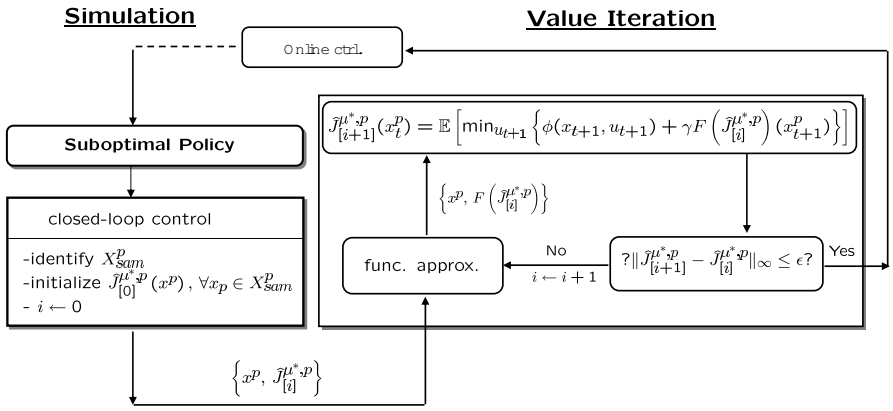


Figure 17: Proposed ADP algorithm based on the post-decision state.

The authors of [60] used an approximate policy iteration scheme where $J^{\mu,p}(x^p), \forall x^p \in \mathcal{X}^p$ is assumed to be linear in a set of basis functions (known or otherwise assumed to be orthogonal polynomials of sufficiently large degree). The coefficients are learnt through a least-squares procedure once the system of interest is allowed to evolve according to the current policy, which is similar to step (1) where relevant states are collected. The limitation is that suitable basis functions are difficult to ascertain in general.

6.3.1 Function approximation & stable learning

The need for function approximation for the purpose of generalization has been discussed. Given a training set (that is, an existing value table, where the VI index will be dropped in this sub-section for notational compactness) $\mathcal{T}^p \triangleq \{x^p(i), \hat{J}^{\mu^*,p}(x^p(i))\}_{i=1}^N$, composed of a finite number (N) of input ($x^p(i) \in X^p_{sam}$) and target values ($\hat{J}^{\mu^*,p}(x_i)$), a function approximator, F, whose domain is $\mathcal{X}^p$, maps an arbitrary query point $x^p_q \in \mathcal{X}^p$ to (a subset of) the real line.

The dominant and natural choice for function approximators has typically involved parametric global approximators such as neural networks or the use of basis functions such as high order orthogonal polynomials or Fourier series [99, 41]. While this approach has met with some success in certain applications (*e.g.* in Backgammon [96], it is not immune from divergent behavior [47, 44] when employed in the context of ADP. In certain cases, the off-line iteration would fail to converge, with the cost-to-go approximation showing non-monotonic behavior or instability with respect to iterations. The author(s) of [97] were the first to attribute the failure with function approximation to an 'over-estimation' effect. [83] demonstrated that sub-optimality can be severe when a global approximator with a linear combination of basis functions is employed. The author(s) of [10] provide insightful illustrations showing the failure of popular function approximators during off-line learning.

There are considerably fewer papers that address function approximation schemes for problems with continuous state and action spaces [60], [46]. The problem of linear quadratic regulation, for which the value function is known to be quadratic in structure, is a noted exception [12]. The authors of [67] proposed a kernel-based approach for problems with continuous states but finite actions and demonstrate convergence to the optimal cost-to-go value function with an increasing number of samples and decreasing kernel bandwidth under a model-free scheme. The authors of [60] proposed a provably convergent approximate policy iteration under the assumption of known basis functions and other technical conditions.

Stable learning during the off-linear value iteration step of the proposed ADP strategy is highly desirable as it can be frustrating to run a large number of iterations only to have the result "blow up" all of sudden due to some complicated coupling between the function

approximation error and value iteration. To have provable convergence of the approximate value iteration (not necessarily to the optimal value function, however), one needs to use a function approximator with a certain property called "non-expansion" . The author of [27] discussed the viability of using such a class of function approximators. With such a choice, the overall operator composed of value-iteration and then function approximation results can be shown to be a contraction map therefore ensuring convergence.

Definition A γ-contraction mapping m (with respect to the infinity norm) defined on a normed vector space (mapping elements from this space, $\mathcal{V}$, to itself) is defined as such:

$$\forall v_1, v_2 \in \mathcal{V},\ \|m(v_1) - m(v_2)\|_\infty \leq \gamma \|v_1 - v_2\|_\infty,\ \gamma \in [0, 1)$$

where v_1, v_2 are arbitrarily chosen elements of $\mathcal{V}$.

Definition When $\gamma = 1$, $m : \mathcal{V} \rightarrow \mathcal{V}$ is termed a non-expansion [27].

From Banach's fixed-point theorem, it can be easily shown that the iterated sequence $\{v, m(v), m^2(v), \ldots\}$ converges to a unique fixed point. As explained earlier, the proposed ADP method starts with initial estimates $\hat{J}^{\mu^*,p}_{[0]}(x^p)$, $\forall x^p \in X^p_{sam}$. This is followed by function approximation (recall that this mapping is denoted by F), and an application of the DP operator, T^p to yield $\hat{J}^{\mu^*,p}_{[1]}(x^p)$, $\forall x^p \in X^p_{sam}$. The process is repeated again. A sufficient condition for convergence is to demonstrate that the overall operator T^p composed with function approximator F is a also contraction map. This, in turn, holds true if F is a non-expansion map.

Proposition 6.3.1 *T^pF is a γ-contraction map if F is non-expansive.*

Proof Given arbitrary vectors $\hat{J}^{\mu^*,p}_1, \hat{J}^{\mu^*,p}_2 \in \mathbb{R}^N$, both of which correspond to the same X^p_{sam},

$$\begin{aligned} \|T^pF(\hat{J}^{\mu^*,p}_1) - T^pF(\hat{J}^{\mu^*,p}_2)\|_\infty &\leq \gamma ||F(\hat{J}^{\mu^*,p}_1) - F(\hat{J}^{\mu^*,p}_2)||_\infty & (96) \\ &\leq \gamma ||\hat{J}^{\mu^*,p}_1 - \hat{J}^{\mu^*,p}_2||_\infty & (97) \end{aligned}$$

The first line is true since T^p is a γ-contraction map defined on the space of value functions. The second inequality follows if one employs a function approximator with a non-expansion property. A more complete proof is given in Appendix B.

Function approximators that employ averaging, as defined below, can be shown to possesses a non-expansion property.

Definition F is an averager if every fitted valued is the weighted average of target values, potentially with the addition of a bias term. Specifically,

$$F(\hat{J}^{\mu^*,p})(x_q^p) \quad = \quad \beta_0\left(x_q^p, \{x^p(j)\}_{j=1}^N\right) + \sum_{i=1}^N \beta_i\left(x_q^p, \{x^p(j)\}_{j=1}^N\right) \hat{J}^{\mu^*,p}(x^p(i)) \tag{98}$$

Here, $\{\beta_i\}_{i=0}^N \geq 0$, and $\sum_{i=1}^N \beta_i \leq 1$. Note that the weights β are allowed to depend on the query point (x_q^p) and input values ($\{x^p(i)\}_{i=1}^N$) but not the target values. That such an averager is a non-expansion (*i.e.* Eq. (97) is true) is easily demonstrated. Again, consider arbitrary vectors $\hat{J}_1^{\mu^*,p}$, $\hat{J}_2^{\mu^*,p} \in \mathbb{R}^N$, both of which correspond to the same X_{sam}^p. Taking an arbitrary query point, x_q^p, the following may be written:

$$\begin{aligned}
\left|F\left(\hat{J}_1^{\mu^*,p}\right)(x_q^p) - F\left(\hat{J}_2^{\mu^*,p}\right)(x_q^p)\right| &= \left|\sum_{i=1}^N \beta_i(x_q^p, \{x^p(j)\}_{j=1}^N)\left(\hat{J}_1^{\mu^*,p}(x^p(i)) - \hat{J}_2^{\mu^*,p}(x^p(i))\right)\right| \\
&\leq \left|\sum_{i=1}^N \beta_i(x_q^p, \{x^p(j)\}_{j=1}^N)\left\|\hat{J}_1^{\mu^*,p} - \hat{J}_2^{\mu^*,p}\right\|_\infty\right| \\
&\leq \left\|\hat{J}_1^{\mu^*,p} - \hat{J}_2^{\mu^*,p}\right\|_\infty
\end{aligned} \tag{99}$$

The first line is true by definition. The second is true since all the weights (β_i) are non-negative and the third is true since the weights (excluding the bias term) sum up to less than unity. Since Eq. (99) is true for arbitrary x_q^p, one immediately concludes that:

$$\left\|F\left(\hat{J}_1^{\mu^*,p}\right) - F\left(\hat{J}_2^{\mu^*,p}\right)\right\|_\infty \quad \leq \quad \left\|\hat{J}_1^{\mu^*,p} - \hat{J}_2^{\mu^*,p}\right\|_\infty \tag{100}$$

6.3.2 Kernel regression

In order to use off-the-shelf optimization solvers, a preferred choice of F is one that is differentiable. Nadaraya-Watson Kernel regression [30] represents a popular class of smooth

averagers used to approximate the conditional expectation of any random variable in a non-parametric manner. Consider approximating the conditional expectation of a random variable $J \in \mathbb{R}$ given the independent variable ($x \in \mathbb{R}^n$). Through simple probability, one has:

$$\mathbb{E}\left[J|x\right] \quad = \quad \int_{dJ} J \frac{p(x,J)}{p(x)} \tag{101}$$

Using Kernel Density Estimation (KDE)[1], a popular non-parametric smooth density estimator, one may approximate $p(x)$ as such:

$$\hat{p}(x) \quad = \quad \frac{1}{N} \sum_{i=1}^{N} K\left(\frac{\|x-x(i)\|_2}{\sigma}\right) \tag{102}$$

where x is an arbitrary query point and $\{x(i)\}_{i=1}^{N}$, a training set. $K(\cdot)$ is a Kernel function that decays exponentially with the squared distance between a query point and a particular point in the training set. A popular choice of $K(\cdot)$ is the Gaussian Kernel (with a diagonal covariance matrix $\Sigma_{n\times n} := \text{diag}(\sigma^2, \ldots, \sigma^2)$). It is used in the sequel and is given by:

$$K\left(\frac{\|x-x(i)\|_2}{\sigma}\right) \quad = \quad \frac{1}{(2\pi\sigma^2)^{0.5n}} \exp\left(-0.5\sigma^{-2}\|x-x(i)\|_2^2\right)$$

The KDE is essentially composed of a mixture of N Gaussians, each of which is centered at a particular training point, $x(i)$. The bandwidth parameter, σ, controls the trade-off between bias and variance errors and may be ascertained through cross-validation [30]. With this, the function approximator for $\hat{J}^{\mu^*,p}$ is given by:

$$F\left(\hat{J}^{\mu^*,p}\right)(x_q^p) = \\ \frac{1}{\sum_{j=1} K\left(\frac{\|x_q^p - x^p(j)\|_2}{\sigma}\right)} \cdot \sum_{i=1}^{N} K\left(\frac{\|x_q^p - x^p(i)\|_2}{\sigma}\right) \hat{J}^{\mu^*,p}(x^p(i)) \tag{103}$$

From Eq. (103), it is immediately apparent that the value function of an query point is given by the weighted average of value functions of neighboring points, with a larger contribution ascribed to closer neighbors. Another interpretation of Nadaraya-Watson kernel regression is that it gives a local constant fit to the data [30]. Small values of σ imply that the values of a relatively fewer number of neighbors (of the query point) are used in the averaging

[1] also known as the Parzen window method

process for function approximation, and vice versa. It is noted that previous process control-relevant ADP work [46] based on the pre-decision state, used a k-Nearest Neighbor (k-NN) approach for function approximation. Whilst that yields a local, non-expansive averager, it also corresponds to one that is non-smooth.

6.3.3 Cautious learning: preventing over-extrapolations

It has been demonstrated [89, 46] that simply using a local averager (that is, with $\beta_0 = 0$), though guaranteeing convergence, does not necessarily give a converged function leading to a stable closed-loop behavior. This is because function approximation error can be significant, particularly when the training data is insufficient. Safeguards against 'over-extrapolation' during value iteration are often needed for the successful implementation of ADP schemes using local, non-expansive averagers, as has been shown in [46]. For a query point located in regions with little data present, distance-weighted averaging may fail to provide meaningful generalizations of the cost-to-go. As suggested by [46], the prevention of taking such a query point may be achieved by including in the cost-to-go term $\beta_0 \geq 0$, a penalty that is imposed whenever the minimization step encounters a query point (x_q^p) located in a region with data sparsity [46]:

$$\beta_0(x_q^p) = A.U\left(\frac{1}{\hat{p}(x_q)} - \rho\right) \cdot \left(\frac{\frac{1}{\hat{p}(x_q)} - \rho}{\rho}\right)^2 \tag{104}$$

Here, ρ is a data-sparsity (defined as the inverse of data-density) threshold, A a scaling parameter, and U, the step function that returns a zero value whenever its argument is non-positive and unity, otherwise (that is, when there is inadequate data density). $\hat{p}(x_q)$ is a measure of data density as ascertained by fitting a Kernel density estimator over the independent variables of the training set.

Using the tuning rules suggested in [46], ρ is given a value as such:

$$\rho \triangleq \left[\frac{1}{N}\sum_{i=1}^{N} K\left(1\right)\right] = K\left(1\right)^{-1} \tag{105}$$

In other words, the threshold data density is that density obtained by assuming that an arbitrary query point is 1 standard deviation (σ) away from every other point in the training set. σ is obtained through cross-validating the initial value table obtained, $\mathcal{T}_{[0]}^p$. A is assigned

a value such that the penalty term corresponds to 10 times the maximum estimate of the initial value function (that is, $\hat{J}^{\mu^*,p}_{max} := 10\|\hat{J}^{\mu^*,p}_{[0]}\|_\infty$) when a query point is assumed to be 3 standard deviations away from any other point in the training set. Namely, one gets:

$$A \triangleq \frac{1}{\left(K(3)^{-1}-K(1)^{-1}\right)^2}\hat{J}^{\mu^*,p}_{max} \tag{106}$$

To ensure boundedness of β_0, a maximum value is imposed on the penalty term:

$$\beta_0\left(x^p_q\right) \leftarrow \min\left(\beta_0\left(x^p_q\right), \hat{J}^{\mu^*,p}_{max}\right) \tag{107}$$

The bias term is non-smooth due to the presence of the step function in Eq. (104) and the upperbound on β_0 (Eq. (107)). To circumvent this, the following computations are performed whenever a query point is encountered and its local data density, $\hat{p}(x^p_q)$, calculated:

$$\hat{p}_1(x^p_q) \leftarrow \widetilde{\max}\Big(\hat{p}(x^p_q), K\left(3\right)\Big) \tag{108}$$

$$\hat{p}_2(x^p_q) \leftarrow \widetilde{\min}\Big(\hat{p}_1(x^p_q), K\left(1\right)\Big) \tag{109}$$

$$\widetilde{\max}(a,b) \triangleq \tfrac{b+a}{2} + \tfrac{\sqrt{(b-a)^2+\delta}}{2} \tag{110}$$

$$\widetilde{\min}(a,b) \triangleq a + \tfrac{b-a}{2} - \tfrac{\sqrt{(b-a)^2+\delta}}{2} \tag{111}$$

where $\widetilde{\max}$ and $\widetilde{\min}$ are smooth approximations of the max and min operators respectively (see Eqs. (110)-(111)), and δ is a small positive real number (set to 0.1% for the rest of this chapter). The effect of Eqs. (108)-(109) is to constrain the data density (at any query point) to the interval $[K\left(3\right), K\left(1\right)]$. Whenever the data density exceeds the upper bound of this said interval, it is assigned a value of $K(1)$, such that no penalty is assigned. Whenever the data density goes below the lower bound of the said interval, it is assigned a value of $K(3)$, such that the maximum penalty is given.

With this, the function approximator has the following form:

$$F\left(\hat{J}^{\mu^*,p}\right)(x^p_q) = A\left(\frac{\frac{1}{\hat{p}_2(x^p_q)}-\rho}{\rho}\right)^2 + \frac{1}{\sum_{j=1} K\left(\frac{\|x^p_q - x^p(j)\|_2}{\sigma}\right)} \cdot \sum_{i=1}^{N} K\left(\frac{\|x^p_q - x^p(i)\|_2}{\sigma}\right)\hat{J}^{\mu^*,p}(x^p(i))$$

with A given by Eq. (106), and ρ by Eq. (105).

6.4 Post-decision-state-based Bellman equations for jump stochastic systems

Bellman's optimality equations may be specialized for the control of jump stochastic systems. Consider the following discrete time equations governing the probabilistic temporal transition of a general Markov jump system:

$$x_{t+1}^c = f(x_t^c, r_t, u_t, \omega_t) \tag{112}$$

$$\text{pr}(r_{t+1} = j | r_t = i) = p_{ij} \tag{113}$$

where r_t refers to a discrete state whose dynamics are governed by a transition probability matrix (see Eq. (113)). x_t^c refers to the continuous state at time index t, with the superscript serving to emphasize the difference from (the discrete) r_t. Note that this emphasis will be dropped outside of this Section, for the sake of notational compactness. Eqs. (112)-(113) describe the transition between the pre-decision states, defined in Eq. (114), between time instances.

$$x_t \triangleq (x_t^c, r_t) \tag{114}$$

As before, f may be decomposed into the following sub-transitions:

$$r_t^p = r_t \tag{115}$$

$$x_t^{c,p} = f_1(x_t^c, r_t, u_t) \tag{116}$$

$$x_{t+1}^c = f_2(x_t^{c,p}, r_t^p, \omega_t) \tag{117}$$

Eq. (115) is true since the effect of the control action does not impact the probabilistic transition of the Markov state. From Eqs. (116)-(117), it is clear that the post-decision state may be defined as such:

$$x_t^p \triangleq (x_t^{c,p}, r_t^p) \tag{118}$$

with the definitions of the continuous and discrete post-decision state variables given in Eqs. (115)-(116).

With this, Bellman's equation used during off-line VI for the purpose of controlling Markov jump stochastic systems are given as:

$$\begin{aligned} J^{\mu^*,p}(x_t^{c,p}, r_t^p) &= \mathbb{E}_{(x_{t+1}|x_t^p)}\left[\min_{u_{t+1}}\left\{\phi(x_{t+1}^c, r_{t+1}, u_{t+1}) + \gamma J^{\mu^*,p}(x_{t+1}^p)\right\}\right] \\ &= \mathbb{E}_{(r_{t+1}|r_t)}\mathbb{E}_{(x_{t+1}|x_t^p, r_{t+1})}\left[\min_{u_{t+1}}\left\{\phi(x_{t+1}^c, r_{t+1}, u_{t+1}) + \gamma J^{\mu^*,p}(x_{t+1}^p)\right\}\right] \\ &= \sum \mathrm{pr}(r_{t+1}|r_t)\mathbb{E}_{(x_{t+1}|x_t^p, r_{t+1})}\left[\min_{u_{t+1}}\left\{\phi(x_{t+1}^c, r_{t+1}, u_{t+1}) + \gamma J^{\mu^*,p}(x_{t+1}^p)\right\}\right] \end{aligned} \tag{119}$$

The first line follows from Eq. (90) and the second from the property of the joint expectation of two random variables. Namely, for random variables v_1, v_2, $\mathbb{E}[v_1, v_2] = \mathbb{E}_{v_1}\mathbb{E}_{(v_2|v_1)}[v_2|v_1]$. Since the transition probability matrix is assumed to be known, only the inner expectation on the third line requires sampling from the distribution of ω_r. Furthermore, we have a (post-decision) value table for each of the discrete Markov states.

For online control, we have the following:

$$\mu^*(x^c, r) = \min_u \left\{\phi(x^c, r, u) + \gamma J^{\mu^*,p}((f_1(x^c, r, u), r))\right\}$$

We demonstrate the above on the control of a constrained Markov Jump Linear System in the next subsection . For unconstrained Markov Jump Linear Systems, it is well known that the optimal policy is given by linear mappings (one for each Markov state), where the solution is obtained through coupled Riccati equations [17].

6.4.1 Example 1: Control of a constrained Markov Jump Linear System

Here, we consider the control of a constrained Markov Jump Linear System given by Eq. (120).

$$\begin{aligned} x_{t+1}^c &= 0.9x_t^c + u_t + \omega_{r_t} \\ \mathbb{E}[\omega_{r=1}\omega_{r=1}'] &= 0.01;\ \mathbb{E}[\omega_{r=2}\omega_{r=2}'] = 1 \\ p_{ii} &= 0.9,\ i \in 1, 2 \end{aligned} \tag{120}$$

where t refers to the discrete-time index, $x_t^c \in \mathbb{R}$, the state variable, and $u_t \in \mathbb{R}$, the manipulated variable. For each regime, the noise term ω, is distributed as a white, zero-mean, Gaussian random variable. It is noted that the covariance of the ω depends on the

realization of the Markov state, which is known (either through direct state feedback or provided by a state-estimator) at each time instant. $r = 1$ corresponds to a a regime with a small covariance and $r = 2$, a regime with large covariance for the noise term. The control objective is to regulate the system at the origin with the following constraint:

$$x_t^c \geq 0.7$$

We compare the performance of Linear Model Predictive Control (LMPC) (with prediction and control horizon set to a sufficiently long duration of $p = 10$ time units) against the proposed ADP approach based on the post-decision state variable. Since ω is an unbounded disturbance signal, a soft constraint approach is employed. Namely, for LMPC, we solve at each sample time:

$$\min_{\{\nu_k\}_{k=0}^{p}} \sum_{k=0}^{p} \|\tilde{x}_{k+1}^c\|_2^2 + 0.1\|\nu_k\|_2^2 + 500||\epsilon_k||_2^2 \tag{121}$$

where $\tilde{x}_0^c$ is initialized as x_t^c, and ϵ_k, $k = 0, 1, \ldots, p$ are non-negative auxiliary decision variables representing the least amount of slack required to make the LMPC problem feasible. That is, $\tilde{x}^c + k + \epsilon_k \geq 0.7$. For the proposed ADP approach, we set the discount factor to a value of 0.98 (which is close to unity) and modify the stage-wise cost to penalize deviations from the state constraints. Namely, the stage-wise cost used during VI and on-line control is given by:

$$\phi(x_t^c, r_t, u_t) \quad = \quad \|x_t\|_2^2 + 0.1\|u_t\|_2^2 + 500 \max\Big(0, 0.7 - x_t\Big)^2$$

To construct X_{sam}^p, we employed the aforementioned LMPC controller and conducted closed-loop simulations for the regulation problem with 20 different realizations of the noise signal. To increase the number of training points, we paired each post-decision continuous state ($x^{c,p}$) to each possible realization of the Markov state. A total of 450 training points were obtained, whose cost-to-go values were initialized by computing the cost for LMPC over a sufficiently long horizon of 120. For the purpose of function approximation, we employed kernel regression (for each of the Markov states) with the bandwidth set to 0.075 for value tables corresponding to both Markov states. To avoid over-extrapolation,

we set $A = 10^3$ and $\rho = 0.95$. Value iteration converged within 30 iterations with a relative termination termination set to $\left|\left|\frac{\hat{J}^{\mu^*,p}_{[i+1]} - \hat{J}^{\mu^*,p}_{[i]}}{\hat{J}^{\mu^*,p}_{[i]}}\right|\right|_\infty \leq 0.1$ for each Markov state.

Results for a typical realization is shown in Figure 18, where the Markov state has switched from $r = 1$ to $r = 2$ at $t = 50$. It is evident that LMPC suffers from constraint violation due to its open-loop optimal formulation whereas ADP provides automatic backing off for both regimes.

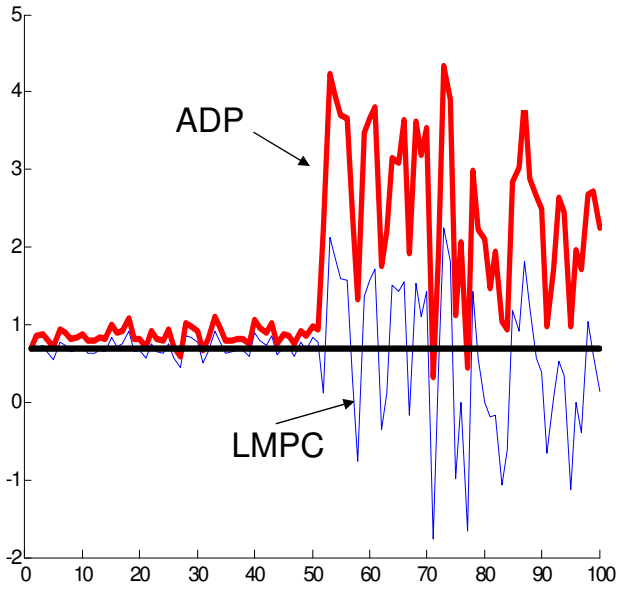


Figure 18: Comparing post-decision state based ADP vs. LMPC. The Markov state switched from $r = 1$ to $r = 2$ midway through the simulation.

6.5 Control of Stochastic Systems: other examples

The proposed ADP approach is not limited to jump systems. Here, we demonstrate the proposed post-decision-state-based ADP algorithm on a variety of stochastic optimal control problems (without jumps) for which popular control solutions (such as MPC) yield unsatisfactory performance due to a poor treatment of the impact of uncertainty.

6.5.1 Example 2: Interaction between state estimation and control

Most of the work on MPC has assumed full state feedback or readily available point state estimates, the latter being the likelier scenario. The implicit assumption of the separation principle between controller and estimator design, whilst accurate for linear systems, does

not necessarily hold for nonlinear ones. In the general case, one would expect the quality of the state estimate to significantly affect the control action and vice versa. In other words, for non-linear stochastic control problems, there exists significant interactions between state estimation and control such that popular open-loop optimal control formulations (such as MPC) might give poor performance. In this example, the proposed ADP approach is applied to a system which is weakly unobservable at the reference state. The same problem was studied by the authors of [32] who modified the optimization problem within MPC to account of the quality of the state estimates.

Consider the following dynamical system:

$$\begin{aligned} x_{t+1} &= x_t + 0.1 x_t u_t + u_t + \omega_t \quad (122) \\ y_t &= x_t^3 + v_t \quad (123) \end{aligned}$$

where $x_t \in \mathbb{R}^{n_x}$ refers to the system state at discrete time index t, $u_t \in \mathbb{R}^{n_u}$, the manipulated variable, and $y_t \in \mathbb{R}^{n_y}$, the observed variable. ω_t (the state excitation noise) and v_t (the measurement noise) are uncorrelated, zero-mean, white Gaussian noise signals with covariance $Q = 0.01$ and $R = 0.1$ respectively. Since full state feedback is assumed to be unavailable, state estimation is performed via an Extended Kalman Filter (EKF). The time and measurement update equations are given by Eq. (124) and Eq. (125) respectively.

$$\begin{aligned} x_{t|t-1} &= x_{t-1|t-1} + 0.1 x_{t-1|t-1} u_{t-1} + u_{t-1} \\ P_{t|t-1} &= (1 + 0.1 u_{t-1})^2 P_{t-1|t-1} + Q \quad (124) \end{aligned}$$

$$\begin{aligned} \mathcal{R}_{\epsilon_{t|t-1}} &= (3x_{t|t-1}^2)^2 P_{t|t-1} + R \\ K_t &= P_{t|t-1}(3x_{t|t-1}^2)\mathcal{R}_{\epsilon_{t|t-1}}^{-1} \\ x_{t|t} &= x_{t|t-1} + K_t(y_t - x_{t|t-1}^3) \\ P_{t|t} &= P_{t|t-1} - K_t(3x_{t|t-1}^2)P_{t|t-1} \quad (125) \end{aligned}$$

The control objective is to bring the system from an arbitrary initial state to the origin, whilst minimizing actuator movement. The performance of MPC is compared against the proposed ADP approach. For MPC, the following open-loop optimization math program is

solved at each time instant t:

$$\min_{\{\nu_k\}_{k=0}^{p}} \sum_{k=0}^{p} \left\{2\tilde{x}_k^2\right\} + 10\tilde{x}_{p+1}^2 \tag{126}$$

where $\tilde{x}_0$ is initialized at $x_{t|t}$, the prediction horizon (p) is set to 9, and the relationship between $\tilde{x}_{k+1}$ and $\tilde{x}_k$ governed by Eq. (122), with ω assumed to be at its mean value of zero across the prediction horizon. It is noted that increasing prediction horizon length has little effect on altering closed-loop performance.

For the proposed ADP approach, the discount factor, γ, is set to 0.98 to ensure the convergence of VI. The pre-decision hyper-state at time t, contains the same information as the pair ($x_{t|t}$, $P_{t|t}$). That is to say, $\mathcal{I}_t = (x_{t|t},\, P_{t|t})'$. With this, the stage-wise cost used within the context of ADP is given by:

$$\begin{aligned} \phi(\mathcal{I}_t, u_t) &= \mathbb{E}_{(\cdot|\mathcal{I}_t)}\left[2x_t^2\right] \\ &= 2x_{t|t}^2 + 2P_{t|t} \end{aligned}$$

From Eq. (124), it is natural to define the post-decision state as the quantities obtained immediately after the time-update step. That is, $\mathcal{I}_t^p \triangleq (x_{t+1|t},\, P_{t+1|t})$. Transitions between the post-to-pre decision state occur through a realization of the innovations term, whose covariance is given by $\mathcal{R}_{\epsilon_{t+1|t}}$.

To construct X_{sam}^p, the aforementioned NMPC controller (with a prediction horizon of 9) was used to conduct closed-loop experiments (each lasting 50 time units) to bring the system from 15 different pre-decision initial states, the set of which is (-10,-8,-6,-4,-2,0,2,4,6,8,10,13,15,17,20), to the origin. For each initial state, we considered 5 different but representative trajectories of ω and v. A total of 250 training points were obtained. The initial post-decision state value functions were approximated by computing the cost corresponding to running the NMPC controller over a horizon of 120 time steps. For the purpose of function approximation, a bandwidth (σ) of 0.10 (obtained through leave-one-out cross validation on the initial post-decision state value table, $\mathcal{T}_{[0]}^p$) was used. To avoid over-extrapolation, A was selected to be 8.7311 and ρ set to 0.1036. A termination criterion of $\|\hat{J}_{[i+1]}^{\mu^*,p} - \hat{J}_{[i]}^{\mu^*,p}\|_\infty \leq 0.10$ was enforced.

6.5.1.1 Closed-loop performance.

As seen from Figure 19, VI converged within 50 iterations, with the major changes in the value function occurring within the first 5 iterations. Results from a typical realization

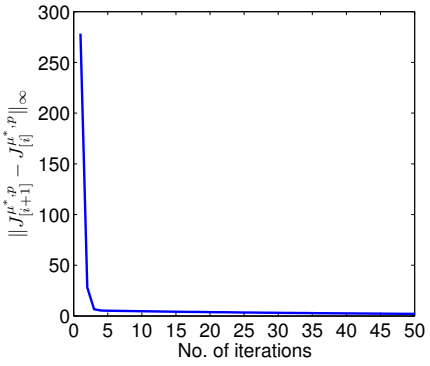
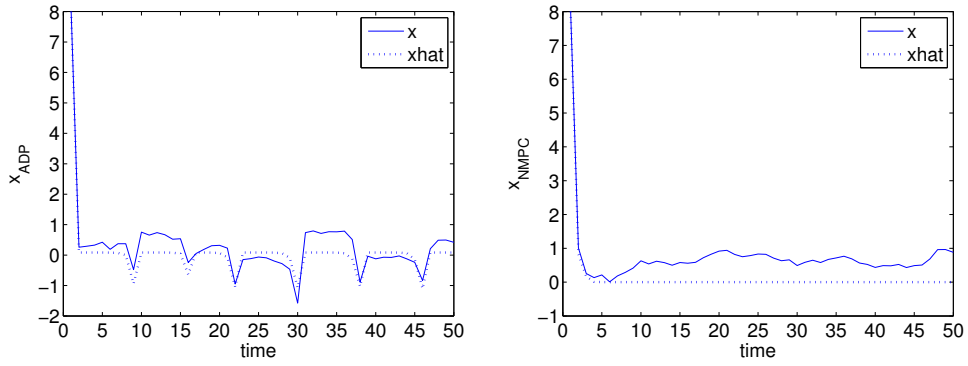


Figure 19: Interaction between estimation & control: plot of learning error vs. VI iterations.

comparing the performance of the proposed ADP and NMPC controllers in bringing the system from an initial state of $x = 8$[2] to the origin are given in Figs. 20 - 22. It is noted that with NMPC, the system is brought to the origin quickly. However, once the state estimates get to zero, there is a loss of observability (see Eqs. (124)-(125) and note the exponential increase in prediction error as shown in Figure 22(b)). Consequently, the control action remains, at the zero, thereby accounting for the uncontrolled drift-like behavior observed in the state (see Figure 20(b)).

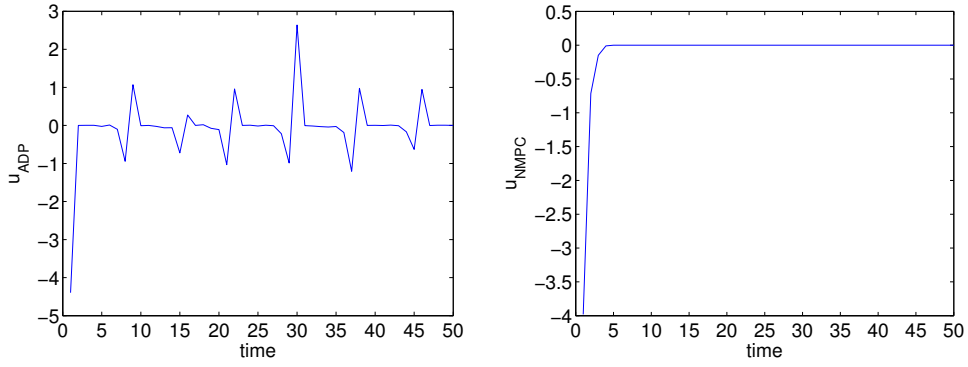
For the proposed ADP controller, the system is perturbed whenever the actual state drifts a significant distance away (as reflected in the increase in prediction error, shown in Figure 22(a)) from the reference state. Periodic perturbations can be seen in Figure 21(a). Similar observations have been made by the authors of [32]. Closed-loop performance results for 100 stochastic realizations are given in Table 10, where $\widehat{\mathbb{E}}$ refers to an empirical average (over 100 realizations).

[2]This choice is representative. It is noted that similar results are obtained with other initial states.



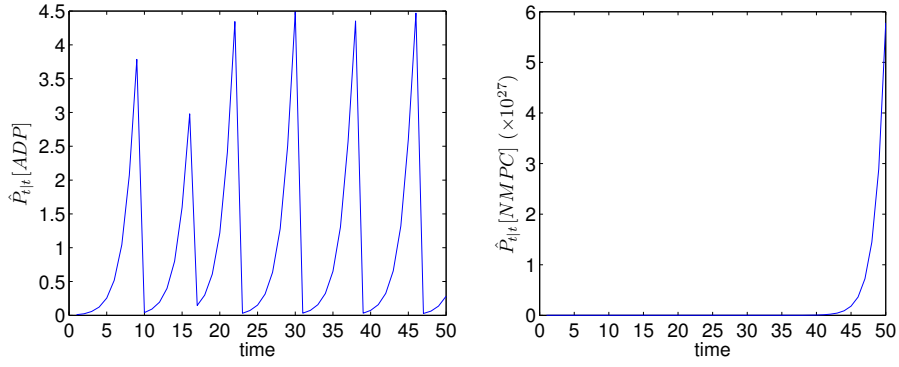
(a) Post-decision-state-based ADP algorithm. (b) NMPC.

Figure 20: Interaction between estimation & control: x vs. t for a typical realization



(a) Post-decision-state-based ADP algorithm. (b) NMPC.

Figure 21: Interaction between estimation & control: u vs. t for a typical realization



(a) Post-decision-state-based ADP algorithm. (b) NMPC.

Figure 22: Interaction between estimation & control: $P_{t|t}$ vs. t for a typical realization.

Table 10: Interaction between estimation & control: performance over 100 realizations

Score	ADP	NMPC
$\widehat{\mathbb{E}}\left[\sum_{t=0}^{50} 2x_t^2\right]$	120.67	295.48

6.5.2 Example 3: Dual control of an integrator with an unknown step change in gain

Controlling systems with unknown parameter values means that the optimal controller is faced with the dual, conflicting goals of i) keeping the system as close to the set-point as possible based on current knowledge and ii) exploration by injecting probing signals (which invariably detracts the system from short-term performance) so as to reduce uncertainty regarding the parameter values. Dual control theory was developed by Fel'dbaum [21, 22, 23, 24] who showed that the optimal solution can be found by dynamic programming. This, however, is often impractical. In this section, we demonstrate the proposed post-decision-state based-ADP approach in solving the dual control problem of an integrator with an unknown gain. The same problem was considered by [49] using a similar ADP approach based on the pre-decision state variable.

The integrator dynamics propagate according to Eqs. (127) and (128)

$$\begin{aligned} y_t &= y_{t-1} + b_t u_{t-1} + e_t \\ b_t &= b_{t-1} + \omega_t \end{aligned} \tag{127}$$

where $y_t \in \mathbb{R}$ is the controlled variable of interest at discrete time index t, u_t, the manipulated variable and e_t zero-mean, white Gaussian noise with covariance R. b is an unknown, time-varying gain described by Eq. (128), where ω_t is zero-mean, white Gaussian noise with covariance Q. A Kalman filter is employed to estimate the unknown parameter b for this

linear system. The governing filter equations are as follows:

$$\begin{aligned}
\mathcal{R}_{\epsilon_{t|t-1}} &= R + u_{t-1}^2 P_{t|t-1} \\
K_t &= P_{t|t-1} u_{t-1} \mathcal{R}_{\epsilon_{t|t-1}}^{-1} \\
y_{t|t-1} &= y_{t-1} + u_{t-1} b_{t|t-1} \\
b_{t+1|t} &= b_{t|t-1} + K_t (y_t - y_{t|t-1}) \\
P_{t+1|t} &= P_{t|t-1} + Q - K_t \mathcal{R}_{\epsilon_{t|t-1}} K_t'
\end{aligned} \tag{128}$$

Here, $\mathcal{R}_{\epsilon_{t|t-1}}$ is the innovations covariance matrix at time t. K_t refers to the Kalman filter gain, $y_{t|t-1}$, the one-step ahead prediction of the output, $b_{t+1|t}$, the one-step ahead prediction of b_t, and $P_{t+1|t}$, the corresponding error covariance matrix. It is evident that the input signal u_{t-1} at time $t-1$ affects the quality of the prediction error of $b_{t+1|t}$ through $\mathcal{R}_{\epsilon_{t|t-1}}$. With this, the post-decision (Eq. (129)) and pre-decision (Eq. (130)) information states are:

$$\mathcal{I}_{t-1}^p \triangleq [u_{t-1}, y_{t-1}, b_{t|t-1}, P_{t|t-1}]' \tag{129}$$

$$\mathcal{I}_t \triangleq [u_{t-1}, y_t, b_{t+1|t}, P_{t+1|t}]' \tag{130}$$

From Eq. (130), it is seen that $\mathcal{I}_{t-1}^p$ contains all the information given by the sequences $\{y_0, y_1, \ldots, y_{t-1}\}$ and $\{u_0, u_1, \ldots, u_{t-1}\}$. The transition from $\mathcal{I}_{t-1}^p$ to $\mathcal{I}_t$ occurs stochastically by adding a realization of the innovations (obtained by sampling the innovations covariance) to $y_{t|t-1}$ and the application of Eq. (128).

The control objective is to find the optimal policy, $\mu^* \in \Gamma$, that yields the following minimum variance cost criterion:

$$\lim_{T \to \infty} \mathbb{E}_{(\cdot|\mathcal{I}_t)} \left[\frac{1}{T} \sum_{k=0}^{T} y_{t+1+k}^2 \right] \tag{131}$$

In this case, the single-stage cost may be written as:

$$\begin{aligned}
\phi(\mathcal{I}_t, u_t) &\triangleq \mathbb{E}_{(\cdot|\mathcal{I}_t)} \left[y_{t+1}^2 \right] \\
&= y_{t+1|t}^2 + \mathcal{R}_{\epsilon_{t+1|t}}
\end{aligned} \tag{132}$$

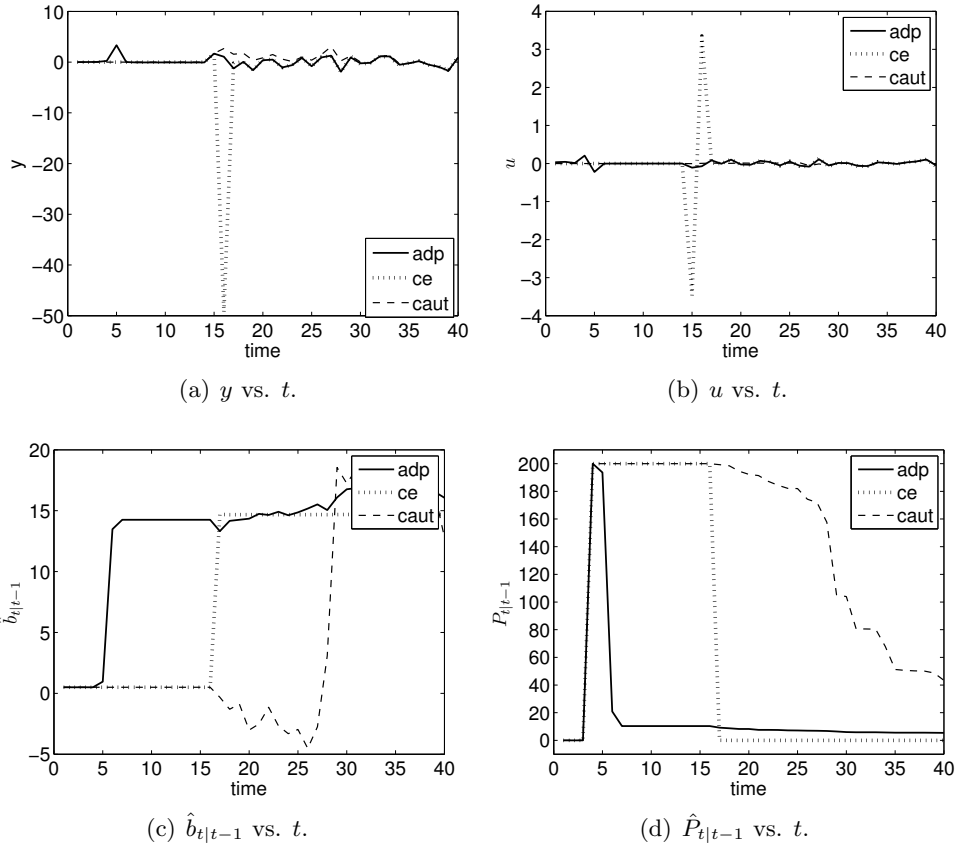
Well-known sub-optimal policies [3] include the one-step ahead certainty equivalence (Eq. (133)) and cautious controllers (Eq. (134)).

$$u_t^{CE} = -\frac{y_t}{b_{t+1|t}} \tag{133}$$

$$u_t^{caut} = -\frac{y_t \cdot b_{t+1|t}}{b_{t+1|t}^2 + P_{t+1|t}} \tag{134}$$

It is well-known that the Certainty Equivalence (CE) controller, which ignores the prediction error covariance in solving Eq. (131), is prone to a bursting phenomenon. On the other hand the cautious controller, derived by solving Eq. (131) with T set to 1, is prone to "turning-off" whenever the error covariance gets exceedingly large. In this example, Q and R are set to values of 0 and 1 respectively. In a fashion similar to [50], we consider the simple case where b may jump from an initial value of 0.5 to anywhere between ± 15 at a random time. For simplicity, it is assumed that the timing of the jump is known to the controller, meaning that the error covariance of the estimator is reset to a relatively high value of 200 when the jump in b occurs. The initial parameter value is assumed to be known exactly (that is, $P_{1|0} = 0$).

The corresponding post-decision-state Bellman's equations, with γ set to 0.98 (so as to ensure convergence of Value Iteration) are given by replacing x_t^p in Eqs. (90)-(91) with $\mathcal{I}_t^p$ and understanding that the $\mathbb{E}$ refers to a conditional expectation. To initialize the post-decision-state-based ADP algorithm, closed-loop simulations were performed using the CE and cautious controllers with and without dithering. The additive dither signals are uniformly distributed between the interval $[-0.1, 0.1]$. Jumps in the parameter b, during the middle of a simulation run (of duration 50 time steps) from an initial value of 0.5 to $b \in [\pm 2, \pm 4, \pm 6, \pm 8, \pm 10, \pm 12, \pm 15]$ were simulated. 3 different realizations of e were simulated. A total of 3480 data points were obtained in populating X_{sam}^p, all of which had their post-decision value functions initialized via employing the CE controller set to run for 120 step times. In designing the function approximator, no penalty is ascribed to the first dimension of $\mathcal{I}^p$ so that exploration is not excessively prohibited during VI and online control. The parameters for the function approximator are set as such: $A = 2$, $\sigma = 0.15$ and $\rho = 0.1$.



(a) y vs. t.

(b) u vs. t.

(c) $\hat{b}_{t|t-1}$ vs. t.

(d) $\hat{P}_{t|t-1}$ vs. t.

Figure 23: Dual control: a typical realization. b jumps from an initial value of 0.5 to 15 at $t = 3$

Figure 23 shows the results of a typical realization where b jumps to a value of 15 at $t = 3$. As in [50], the simulations are such that the measurement noise is kept to zero until $t = 15$, whereafter e behaves like a white, Gaussian disturbance signal with zero mean and covariance $R = 1$. As can be seen in Figure 23(b), the proposed post-decision state ADP algorithm injects probing signals early on when $\hat{P}_{t|t-1}$ increases due to the jump in b. Such an early injection of a probing signal actively reduces the uncertainty associated with $\hat{b}_{t|t-1}$ (as can be seen in Figures 23(c)-23(d). In contrast, the CE controller exhibits "bursting" at $t = 15$. On the other hand, the cautious controller is "turned-off" for most part of the simulation and as such, there is a significant delay in terms of reducing the uncertainty associated with $\hat{b}_{t|t-1}$.

Table 11[3] shows that the ADP controller yields superior closed-loop performance.

Table 11: Dual control of integrator with an unknown gain: mean performance over 100 realizations

Score	ADP	CE	Cautious
$\widehat{\mathbb{E}}\left[\sum_{t=0}^{40} y_t^2\right]$	49.2	1497	65.5

6.5.3 Example 4: Constrained linear stochastic system- double integrator problem

We consider the following constrained double integrator problem studied by [6] in the context of MPC for stochastic systems:

$$x_{t+1} = Ax_t + Bu_t + \Upsilon w_t \tag{135}$$

where matrix $A = [1\,0; 1\,1]$[4], $B = [1; 0]$, $\Upsilon = [1; 0]$ and ω_t is zero-mean, white Gaussian noise with its second moment, $\mathbb{E}[\omega_t \omega_t'] = 0.2, \forall t$. The control objective is to bring the system from an arbitrary initial state ($[0; 14]$ in the following simulations) to the origin whilst minimizing actuator movement. The second dimension of the state vector is constrained to be non-negative ($x_2 \geq 0$), as in the input vector $u_t \in [-0.5, 0.5]$. In this example, full

[3] $\widehat{\mathbb{E}}$ refers to an empirical average
[4] in Matlab notation

state-feedback is assumed.

We compare the performance of a Linear Model Predictive Controller (LMPC) (with prediction and control (p) horizon set to a sufficiently long duration of 15 time steps) against the proposed ADP approach based on the post-decision state variable. The post-decision state is defined as the quantity obtained after an action is taken but before the uncertainty is realized. That is, $x_t^p \triangleq Ax_t + Bu_t$. Since ω is an unbounded signal, we employ a soft-constraint approach for both LMPC (to avoid running into infeasibility issues) and the proposed ADP strategy. As is typically done, LMPC is implemented assuming ω remains at its nominal value of 0 over the prediction horizon. Namely, for LMPC, we solve at each time step, t:

$$\min_{\{\nu_k\}_{k=0}^{p}} \sum_{k=0}^{p} 0.7\|\tilde{x}_{k+1}\|_2^2 + 0.33\|\nu_k\|_2^2 + 100||\epsilon_k||_2^2 \tag{136}$$

where $\tilde{x}_0$ is initialized as x_t, and ϵ_k, $k = 0, 1, ..., p \geq 0$ are non-negative auxiliary decision variables representing the least amount of slack required to make the LMPC problem feasible. That is, $[0\ 1]\,\tilde{x}_k + \epsilon_k \geq 0$. These inequalities are easily incorporated into the math program defined by Eq. (136). The relationship between $\tilde{x}_{k+1}$ and ($\tilde{x}_k$, ν_k) is given by Eq. (135), with the noise term assumed to be zero throughout the entire prediction horizon. Also, the input vector is constrained to satisfy the aforementioned bounds of ± 0.5.

For the proposed ADP approach, we set the discount factor to a value close to unity, that is, $\gamma = 0.98$ and modify the stage wise cost to penalize deviations from the state constraints. Namely, the stage-wise cost used during VI is given by:

$$\phi(x_t, u_t) \triangleq 0.70\|x_t\|_2^2 + 0.33\|u_t\|_2^2 + 100 \max\left(0, -[0,\ 1]x_t\right)^2$$

Hard constraints on u are imposed during the off-line value iteration process and on-line implementation of the ADP-based controller.

To construct X_{sam}^p, we used the aforementioned LMPC controller and conducted closed-loop experiments bringing the system from 40 different initial pre-decision states to the origin. Note that the initial state used for on-line testing is excluded from these 40 initial states. Namely, we consider all permutations of the sets $\{-2, 0, -1, 1, 2\}$ and $\{-4, -2, 0, 2, 4, 6, 8, 10\}$

to create various values for the first and second dimension of the initial state respectively. Consequently, a total of 3500 training points, whose initial cost-to-go values were initialized by computing the cost for LMPC over a sufficiently long horizon (of 120), was obtained as a result of the initialization scheme. For the purpose of function approximation, we employed kernel regression with the bandwidth, σ, set to 0.10. To avoid over-extrapolation, we selected $A = 2.6 \times 10^3$, and $\rho = 0.1036$. Value-iteration converged within 50 iterations, where the relative error termination criterion is set to $\left|\left| \frac{\hat{J}^{\mu^*,p}_{[i+1]} - \hat{J}^{\mu^*,p}_{[i]}}{\hat{J}^{\mu^*,p}_{[i]}} \right|\right|_\infty \leq 0.1$.

Results from 500 stochastic realizations presented as follows. As can be seen from Table 12, the proposed ADP controller has an average[5] finite horizon score an order of magnitude lower than a deterministic approach typified by LMPC. In particular, LMPC suffers from excessively high variance in terms of closed-loop performance. A look at the time series plots of the second dimension of x for both methods (see Figure 24) reveals that LMPC results in significant constraint violation. On the other hand, the majority of the realizations based on the ADP approach do not violate the lower bound constraint.

Table 12: Double integrator example: comparing closed-loop performance

Score	ADP	LMPC
$\widehat{\mathbb{E}}\left[\sum_{t=0}^{30} \phi(x_t, u_t)\right]$	1600	10000

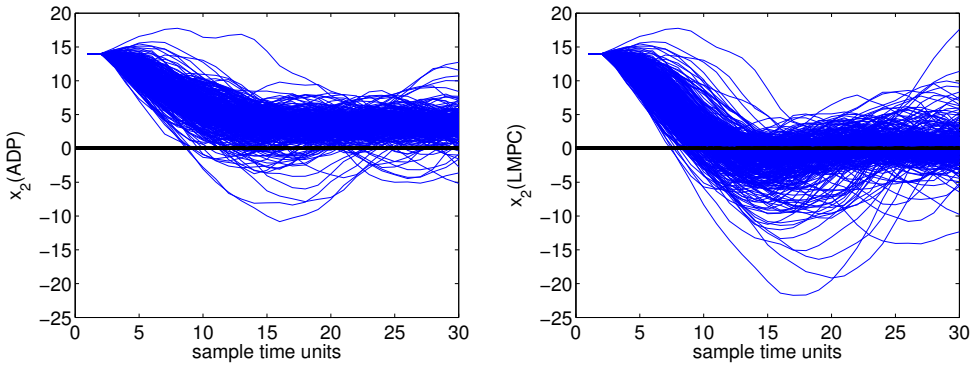
6.5.4 Example 5: Constrained chemostat - maximizing productivity subject to conversion constraints

Consider the governing equations of an archetypal chemostat.

$$\begin{aligned} \dot{x}_1 &= x_1 \tfrac{\mu_{max} x_2}{\kappa + x_2} - x_1 u \\ \dot{x}_2 &= u[x_{2f} - x_2] - \tfrac{\mu_{max}}{Y} \tfrac{x_1 x_2}{\kappa + x_2} \end{aligned}$$

where $x = [x_1; x_2] \in \mathbb{R}^2$ is the state vector composed of the instantaneous concentration of the product (x_1) and substrate (x_2) respectively. $0 \leq u \in \mathbb{R}$, the dilution rate, is the non-negative manipulated variable. x_{2f} refers to the instantaneous concentration of the

[5] $\widehat{\mathbb{E}}$ is based on sample averaging



(a) ADP algorithm. (b) LMPC.

Figure 24: Double integrator example: x_2 vs. t for 500 realizations. Lower bound for x_2 is 0.

substrate feed. The maximum specific growth rate μ_{max} is set to 1, the yield coefficient to 1 and κ to 0.02. For the following simulations, the sampling rate is set to 0.5. This yields the following discrete-time equations:

$$x_{t+1} \quad = \quad f_h(x_t, u_t, x_{2f,t}) \tag{137}$$

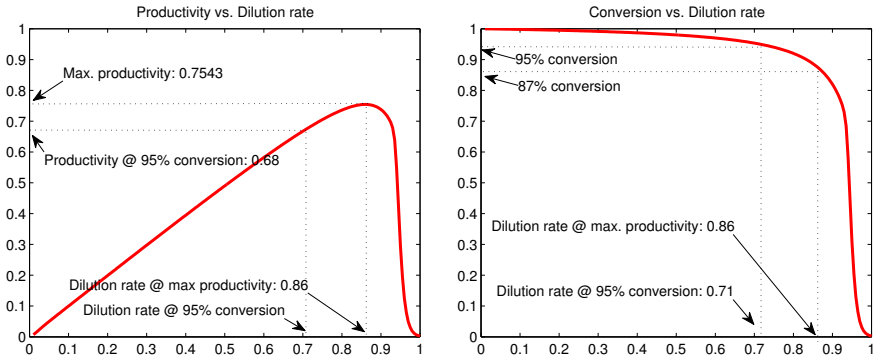
where f_h refers to the integration of the aforementioned ordinary differential equations over one sample-time unit with initial conditions set to x_t, and the control input and feed concentration of the substrate held constant over the said period of numerical integration.

For the purpose of simulation, we assume that the feed concentration (x_{2f}) fluctuates around a mean value of 1, and is perturbed by zero-mean, white Gaussian noise (ω):

$$x_{2f,t} = 1 + \omega_t,\ \mathbb{E}[\omega_t \omega_t'] = 10^{-3} \tag{138}$$

It is desirable to maximize the productivity of the product, $\mathcal{P}_t \triangleq x_{1,t} u_t$, whilst ensuring that the conversion of the substrate, $\psi_{x_2} \triangleq 1 - \frac{x_2}{x_{2f}}$, does not go lower than a relatively high value of 95%. Such an economically motivated constraint is common in several key process industries, such as bioethanol production. There is a tradeoff between productivity and conversion, as reflected in the steady state profiles depicted in Figure 25. Productivity increases with dilution rate and then decreases as the system approaches washout. Conversion, on the other hand, is a decreasing function of space-velocity or equivalently the dilution rate.

Maximum productivity ($\mathcal{P}^* = 0.7543$) occurs at a dilution rate that corresponds to conversion levels significantly below the required 95% threshold. We compare the performance of



(a) Productivity profile. (b) Conversion profile

Figure 25: Steady state profiles

Non-linear MPC (NMPC) against the proposed ADP strategy, assuming full feedback (of the state and instantaneous feed concentration) is available. Instead of full-fledged NMPC, we employ successive-linearization based MPC (slMPC), a computationally efficient alternative proposed by [45]. For this example, we have found the closed-loop performance of slMPC to be similar to that of NMPC. For slMPC, we employed a prediction and control horizon, p, of 10 sample units. The following math program is solved at each time instant:

$$\min_{\{\nu_k\}_{k=0}^{p}} \quad \sum_{k=0}^{p} \left(||\tilde{\mathcal{P}}_{k+1} - \mathcal{P}^*||_2^2 + 100||\epsilon_k||_2^2 \right) \tag{139}$$

where the math program is initialized with $\tilde{x}_0 = x_t$, the relationship between $\tilde{x}_{k+1}$ and $\tilde{x}_k$ is governed by Eq. (137), and the corresponding productivity and conversion given by $\tilde{\mathcal{P}}_k \triangleq \tilde{x}_k \nu_k$ and $\tilde{\psi}_k \triangleq 1 - \frac{\tilde{x}_{2,k}}{x2,t}$ respectively. Note that the slMPC controller assumes that the feed conditions remain at its current value throughout the prediction horizon. $\epsilon \geq 0$, is a non-negative variable representing the least amount of slack required for conversion to be greater than 95%. That is, $\epsilon_k + \tilde{\psi}_{x_2,k} \geq 0.95$. The idea is to regulate the system at an equilibrium point that corresponds to the largest possible value of the dilution rate without exceeding the conversion bound so that productivity is maximized. The dynamics of the system are assumed to be governed by matrices obtained through linearization of the governing ordinary differential equations about the current state and past input vector.

This results in a convex quadratic program.

Concatenating Eqs. (137)-(138), we have the following combined dynamics:

$$\begin{pmatrix} x_{t+1} \\ x_{2f,t+1} \end{pmatrix} = \begin{pmatrix} f_h(x_t, u_t, x_{2f,t}) \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ \omega_t \end{pmatrix} \tag{140}$$

Consequently, the pre-decision ($\mathcal{I}_t$) and post-decision states ($\mathcal{I}_t^p$) may be defined as follows:

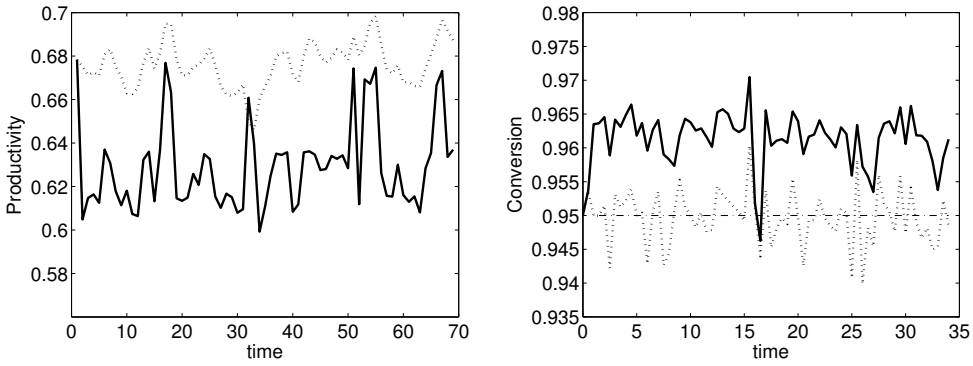
$$\mathcal{I}_t \triangleq \begin{pmatrix} x_t \\ x_{2f,t} \end{pmatrix}, \mathcal{I}_t^p \triangleq \begin{pmatrix} x_{t+1} \\ 1 \end{pmatrix} \tag{141}$$

For the proposed ADP approach, γ is set to 0.98 and the stage-wise cost defined as such: $\phi(\mathcal{I}_t, u_t) \triangleq ||\mathcal{P}_t - \mathcal{P}^*||_2^2 + 100 \max(0, 0.95 - \psi_{x_2,t})^2$. To determine X_{sam}^p, we used an slMPC controller (with horizon length of 10 time units) and conducted closed-loop experiments regulating the system at an initial state corresponding to a conversion of 0.95. A total of 300 training points was obtained from the initialization scheme. We used kernel regression for function approximation with the bandwidth, σ, set to 0.15, A to 1.93 and ρ to 0.087, in order to prevent over-extrapolation.

Results from a typical realization are depicted in Figure 26. It is apparent that the ADP-based approach, compared to slMPC, results in minimal constraint violation at the expense of slightly lower productivity. It is noted that the steady-state productivity corresponding to 95% conversion is 0.68.

6.6 Conclusions

In this chapter, it has been argued that, for process control problems, the post-decision-state formulation offers the ability to use deterministic math programming solvers to be utilized, both off-line and on-line and therefore may be more convenient than the more conventional pre-decision-state formulation. In addition, the use of function approximators with nonexpansion properties offer stable learning. Robustness against over-extrapolation can be achieved through the use of a tailor-made penalty function. Key examples are used to highlight the importance of treating uncertainty in a systematic fashion and to demonstrate the proposed algorithm.



(a) Productivity vs time.

(b) Conversion vs time.

Figure 26: Constrained non-linear chemostat example: closed-loop performance of a typical realization. ADP: solid line (-); slMPC: dotted line(..); lower bound on conversion: dash-dot (-.)

CHAPTER VII

CONTRIBUTIONS & FUTURE RESEARCH

7.1 Summary of contributions

This thesis tackles the issue of uncertainty in the context of process control. Disturbance modeling and the optimal control of jump (and non-jump) stochastic systems constitute the major thrust of the work. The main contributions are listed in the following sections.

7.1.1 Disturbance modeling

- Realistic disturbance modeling using a Hidden Markov Model (HMM) framework: see Chapter 3. The overall result is a more sophisticated model used by an existing state estimator for jump systems. At the expense of slightly higher computational costs (due to the state estimator), the proposed HMM disturbance model provides better tracking compared to a state estimator based on the commonly employed (in process control) integrated white noise disturbance model. Better tracking performance translates to superior closed loop performance without any redesign of the controller, through the typical assumption of separation and certainty equivalence. As a result, this provides a tool that can be readily adopted by process control practitioners.

- Application of the said HMM framework (see Chapter 3) for providing flexibility in imparting integral action under a wide number of scenarios. Typical off-set free control algorithms suffer from poor transient performance when there is significant mismatch between where the disturbance enters and where it is *assumed* to enter.

- Demonstrating the flexibility of the HMM framework in rejecting *deterministic* step changes. See Chapter 3.

- Application of the said HMM framework for *in-silico* experiments for second generation bio-ethanol production, an important element of sustainable energy initiatives. Specifically, the HMM framework is used to describe highly-varying feed conditions

which occur as a result of the multitude of feedstock used. The HMM framework is shown to provide superior performance to the typical random-walk description of the disturbance patterns. See Chapter 4.

- Applying the HMM framework for the purpose of fault detection. This is a generalization of the commonly used Mixture of Gaussians assumption. Novel application of the HMM framework to the problem of valve stiction. See Chapter 5.

7.1.2 Post-decision-state-based approximate dynamic programming

- Extending current ADP framework based on the pre-decision state variable to the post-decision state variable. Numerical advantages allow for the use of popular off-the-shelf solvers and the parallelization of Value Iteration. See Chapter 6.

- Demonstration of the need for the systematic treatment of uncertainty through a wide array of control problems. See examples section in Chapter 6.

7.2 Future work

- System identification of jump systems, a difficult problem in itself, will be tackled in the future.

- Application of the HMM framework for the fault-detection and diagnosis in large-scale systems.

- Application of the HMM framework for blood glucose control in type-1 diabetics. The integrated-white noise assumption is common in the blood glucose control literature [58, 59]. As explained throughout the first part of the thesis, this oftentimes requires tedious tuning of the observer/ Kalman filter. However, experimental data [19] suggests that blood sugar levels exhibit multi-regime behavior. Specifically, before a meal, blood sugar levels fluctuate like white-noise, whereas during a meal, blood sugar levels exhibit ramp-like behavior. This may potentially be modeled via the proposed HMM framework used in the first part of this thesis.

- The non-parametric function approximation approach used in the ADP work is computationally intensive compared to methods which assume basis functions and then use (recursive) least squares to regress the coefficients [60]. However, choosing good basis functions, is a difficult task in general. The combination of both methods may be employed in the context of Policy Iteration where the latter approach is first used to model, in broad-strokes, the cost-to-go function. The residuals may then be fitted using the proposed non-parametric methods. Policy iteration is then employed to ensure convergence. This is followed up by policy improvement, completing one iteration of Policy Iteration.

APPENDIX A

DETAILS FOR HYBRID CYBERNETIC MODEL-BASED SIMULATION OF CONTINUOUS PRODUCTION OF LIGNOCELLULOSIC ETHANOL

A.1 Suboptimal state estimation via GPB2: time and measurement update steps

Details steps comprising the `filter` operator (Section 4.2.2) are given as such:

Time Update:

$$\begin{aligned}
\eta_{t+1|t}(\mathcal{R}_t^{t+1}) &= \mathcal{F}_{r_{t+1}}(\eta_{t|t}(r_t), u_t) \\
P_{t+1|t}(\mathcal{R}_t^{t+1}) &= A_t^o P_{t|t}(r_{t-1}) A_t^{o\prime} + \Gamma_t^o R_{r_t}^{\xi} \Gamma_t^{o\prime} \\
y_{t+1|t}(\mathcal{R}_t^{t+1}) &= \mathcal{G}(\eta_{t+1|t}(\mathcal{R}_{t-1}^{t})) \\
\mathcal{R}_{\epsilon}(\mathcal{R}_t^{t+1}) &= C^o P_{t+1|t}(\mathcal{R}_t^{t+1}) C^{o\prime} + R^v
\end{aligned} \tag{142}$$

The innovations (i.e. one-step ahead output prediction) covariance matrix is given by $\mathcal{R}_{\epsilon}(\cdot)$. Here, the triplet $(A_t^o, \Gamma_t^o, C_t^o)$ (which depends on both r_t and r_{t+1}) reflects the successive-linearization assumption underpining the Extended Kalman Filter (EKF), where

$$\begin{bmatrix} A_t^o & \Gamma_t^o \\ 0 & I \end{bmatrix} = \exp\left(h \cdot \begin{bmatrix} \tilde{A}_t^o & \tilde{\Gamma}_t^o \\ 0 & I \end{bmatrix} \right) \tag{143}$$

$$\tilde{A}_t^o = \left. \frac{\partial \mathcal{F}_{r_{t+1}}}{\partial x} \right|_{\eta_{t|t}(r_t), u_t} \tag{144}$$

$$\tilde{\Gamma}_t^o = \left. \frac{\partial \mathcal{F}_{r_{t+1}}}{\partial d} \right|_{\eta_{t|t}(r_t), u_t} \tag{145}$$

$$C^o = \frac{\partial \mathcal{G}}{\partial x} \tag{146}$$

Measurement Update:

$$\begin{aligned}
\eta_{t+1|t+1}(\mathcal{R}_t^{t+1}) &= \eta_{t+1|t}(\mathcal{R}_t^{t+1}) + L_{t+1}(\mathcal{R}_t^{t+1}) \cdot (y_{t+1} - y_{t+1|t}(\mathcal{R}_t^{t+1})) \\
L_{t+1}(\mathcal{R}_t^{t+1}) &= P_{t+1|t}(\mathcal{R}_t^{t+1}) C^{o\prime} [\mathcal{R}_{\epsilon}(\mathcal{R}_t^{t+1})]^{-1} \\
P_{t+1|t+1}(\mathcal{R}_t^{t+1}) &= [I - L_{t+1}] \cdot P_{t+1|t}(\mathcal{R}_t^{t+1})
\end{aligned} \tag{147}$$

A.2 Details of the hybrid cybernetic model

For convenience, the equations governing the dynamics of the chemostat (Eqs. (45)-(46)) are reproduced:

$$\frac{d\tilde{x}}{dt} = [S_{\tilde{x}}Z] \cdot \mathrm{diag}([\tilde{\nu}_1, \ldots, \tilde{\nu}_{12}]) \cdot \tilde{r} \cdot x_{BIOM} + D(\tilde{x}_{in} - \tilde{x})$$

$$\frac{dx_E}{dt} = \alpha + \mathrm{diag}([\tilde{u}_1, \ldots, \tilde{u}_{12}]) \cdot \tilde{r}_E - \mathrm{diag}([\beta_1, \ldots, \beta_{12}]) \cdot x_E - \mu \cdot I x_E$$

The various terms and parameter values, adapted from [92], will be defined and provided in this section.

A.2.1 Abbreviations

$\tilde{x}$ is defined as concentrations of various metabolites:
$[x_{GLC}, x_{XYL}, x_{ACTx}, x_{BIOM}, x_{CO_2}, x_{ETH}, x_{GOLx}, x_{MAINT}, x_{XOLx}]'$. Table 6 provides the explanation of these abbreviations.

A.2.2 Fluxes and enzyme synthesis reaction rates

The definition of vector $\tilde{r}$ and $\tilde{r}_E$, with Monod-type kinetics, and values of associated parameters are given by Eqs. (148)-(149) and Table 13.

$$\tilde{r}_i = \begin{cases} k_i^{max} \frac{x_{GLC}}{K_G + x_{GLC}} \frac{1}{1 + x_{ETH}/K_{I,G}} \frac{x_{E,i}}{x_{E,i}^{max}}, \; i = 1, \ldots, 3 \\ k_i^{max} \frac{x_{XYL}}{K_X + x_{XYL}} \frac{1}{1 + x_{ETH}/K_{I,X}} \frac{x_{E,i}}{x_{E,i}^{max}}, \; i = 4, \ldots, 7 \\ k_i^{max} \frac{x_{GLC}}{K_G + x_{GLC}} \frac{1}{1 + x_{ETH}/K_{I,G}} \frac{x_{XYL}}{K_X + x_{XYL}} \frac{1}{1 + x_{ETH}/K_{I,X}} \frac{x_{E,i}}{x_{E,i}^{max}}, \; i = 8, \ldots, 12 \end{cases} \tag{148}$$

$$\tilde{r}_{E,i} = \begin{cases} k_{E,i} \frac{x_{GLC}}{K_G + x_{GLC}} \frac{1}{1 + x_{ETH}/K_{I,G}}, \; i = 1, \ldots, 3 \\ k_{E,i} \frac{x_{XYL}}{K_X + x_{XYL}} \frac{1}{1 + x_{ETH}/K_{I,X}}, \; i = 4, \ldots, 7 \\ k_{E,i} \frac{x_{GLC}}{K_G + x_{GLC}} \frac{1}{1 + x_{ETH}/K_{I,G}} \frac{x_{XYL}}{K_X + x_{XYL}} \frac{1}{1 + x_{ETH}/K_{I,X}}, \; i = 8, \ldots, 12 \end{cases} \tag{149}$$

Here, Elementary Flux Modes (EFMs) 1-3 belong to the glucose cluster, 4-7, the xylose cluster and the remaining ones (8-12) to the glucose/ xylose cluster.

$S_{\tilde{x}}Z$, the product of the extra-celluar stoichiometric matrix and the vertical concatenation of the twelve EFMs , is the following 9 (equalling the number of extra-celluar species)

by 12 (corresponding to the number of effective elementary flux modes) matrix:

$$\begin{bmatrix}
-1 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\
0 & 0 & 0 & -1 & -1 & -1 & -1 & -4 & -4 & -10.9 & -0.375 & -0.0123 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 9.28 & 0.188 & 0.147 \\
0 & 0.03956 & 0.0247 & 0 & 0 & 0.031 & 0.0072 & 0 & 0 & 0.418 & 0 & 0.0215 \\
2 & 1.64 & 1.40 & 1.83 & 1 & 1.546 & 1.16 & 2 & 9.33 & 15.8 & 1.59 & 1.30 \\
2 & 1.31 & 1.20 & 1.58 & 0.75 & 1.039 & 0.849 & 0 & 8.33 & 3.78 & 0.844 & 1.04 \\
0 & 0 & 0.374 & 0 & 0 & 0 & 0.609 & 0 & 0 & 0 & 1.41 & 0.478 \\
2 & 0 & 0 & 1.58 & 0.75 & 0 & 0 & 2 & 8.33 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 4 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Table 13: Kinetic parameters for Eqs. (148)-(149).

EFM	k_i^{max} (mM/h)	$x_{E,i}^{max}$ (mM)	$(k_{E,i}, \alpha_i, \beta_i)$ (1/h)	K_G/K_X (mM)	$K_{I,G}/K_{I,X}$ (mM)
EFM_1	173	5.5	$(1, 0.1, 0.2)$	$K_G = 3.14$	$K_{I,G} = 410$
EFM_2	28.6	0.83	$(1, 0.1, 0.2)$	$K_G = 3.14$	$K_{I,G} = 410$
EFM_3	32.1	1.11	$(1, 0.1, 0.2)$	$K_G = 3.14$	$K_{I,G} = 410$
EFM_4	21.0	5.5	$(1, 0.1, 0.2)$	$K_X = 22.7$	$K_{I,X} = 224$
EFM_5	18.8	5.5	$(1, 0.1, 0.2)$	$K_X = 22.7$	$K_{I,X} = 224$
EFM_6	11.9	1.92	$(1, 0.1, 0.2)$	$K_X = 22.7$	$K_{I,X} = 224$
EFM_7	14.3	3.62	$(1, 0.1, 0.2)$	$K_X = 22.7$	$K_{I,X} = 224$
EFM_8	0.978	5.5	$(1, 0.1, 0.2)$	-	
EFM_9	2.53	5.5	$(1, 0.1, 0.2)$	-	
EFM_{10}	0.912	1.89	$(1, 0.1, 0.2)$	-	
EFM_{11}	1.02	5.5	$(1, 0.1, 0.2)$	-	
EFM_{12}	0.045	5.47	$(1, 0.1, 0.2)$	-	

APPENDIX B

PROOF: CONVERGENCE OF POST-DECISION-STATE-BASED VI USING LOCAL FUNCTION APPROXIMATORS

We first note that the following property holds for arbitrary functions $g, h : \mathcal{A} \rightarrow \mathbb{R}$:

$$|\min_a g(a) - \min_a h(a)| \leq \max_a |g(a) - h(a)|$$

Then,

$$\begin{aligned}
&\left|\hat{J}^{\mu^*,p}_{[i+1]}(x^p) - \hat{J}^{\mu^*,p}_{[i]}(x^p)\right| = \\
&\quad \left|\mathbb{E}_{(w|x^p)}\left[\min_u \left\{\phi(f_2(x^p,w),u) + \gamma F\left(\hat{J}^{\mu^*,p}_{[i]}\right)(f_1(f_2(x^p,w),u)\right\}\right]\right. \\
&\quad \left. - \mathbb{E}_{(w|x^p)}\left[\min_u \left\{\phi(f_2(x^p,w),u) + \gamma F\left(\hat{J}^{\mu^*,p}_{[i-1]}\right)(f_1(f_2(x^p,w),u)\right\}\right]\right| \\
&\quad = \left|\mathbb{E}_{(w|x^p)}\left[\min_u \left\{\phi(f_2(x^p,w),u) + \gamma F\left(\hat{J}^{\mu^*,p}_{[i]}\right)(f_1(f_2(x^p,w),u)\right\}\right.\right. \\
&\quad \left.\left. - \min_u \left\{\phi(f_2(x^p,w),u) + \gamma F\left(\hat{J}^{\mu^*,p}_{[i-1]}\right)(f_1(f_2(x^p,w),u)\right\}\right]\right| \\
&\quad \leq \gamma\left|\mathbb{E}_{(w|x^p)}\left[\max_u \left|F\left(\hat{J}^{\mu^*,p}_{[i]}\right)(f_1(f_2(x^p,w),u) - F\left(\hat{J}^{\mu^*,p}_{[i-1]}\right)(f_1(f_2(x^p,w),u)\right|\right]\right| \\
&\quad \leq \gamma\left|\mathbb{E}_{(w|x^p)}\left[||F\left(\hat{J}^{\mu^*,p}_{[i]}\right) - F\left(\hat{J}^{\mu^*,p}_{[i-1]}\right)||\right]\right| \\
&\quad = \gamma||F\left(\hat{J}^{\mu^*,p}_{[i]}\right) - F\left(\hat{J}^{\mu^*,p}_{[i-1]}\right)||
\end{aligned}$$

This implies that

$$\|\hat{J}^{\mu^*,p}_{[i+1]} - \hat{J}^{\mu^*,p}_{[i]}\|_\infty \leq \gamma\|F\left(\hat{J}^{\mu^*,p}_{[i]}\right) - F\left(\hat{J}^{\mu^*,p}_{[i-1]}\right)\|_\infty$$

Since the function approximator is a local, non-expansive averager, it can also be shown that:

$$\|F\left(\hat{J}^{\mu^*,p}_{[i]}\right) - F\left(\hat{J}^{\mu^*,p}_{[i-1]}\right)\|_\infty \leq \|\hat{J}^{\mu^*,p}_{[i]} - \hat{J}^{\mu^*,p}_{[i-1]}\|_\infty \tag{150}$$

Putting the above together,

$$\|\hat{J}^{\mu^*,p}_{[i+1]} - \hat{J}^{\mu^*,p}_{[i]}\|_\infty \leq \gamma^i\|\hat{J}^{\mu^*,p}_{[1]} - \hat{J}^{\mu^*,p}_{[0]}\|_\infty \tag{151}$$

As $\gamma \in [0, 1)$, we have uniform convergence (due to convergence in sup-norm).

REFERENCES

[1] Almeida, G. M. and Park, S. W., "Fault detection and diagnosis in the damadaics benchmark actuator systems - a hidden markov model approach," in *17th IFAC World Congress*, (Seoul, Korea), p. 12419, 2008.

[2] Arkun, Y. and Stephanopoulos, G., "Studies in the synthesis of control structures for chemical processes. part iv. design of steady-state optimizing control structures for chemical process units.," *AIChE Journal*, vol. 26, pp. 975–991, 1980.

[3] Astrom, K. and Wittenmark, B., *Computer controlled systems*. Prentice Hall, 3rd ed., 1997.

[4] Bar-Shalom, Y. and Li, X.-R., *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, 1993.

[5] Basseville, B., "Detecting changes in signals and systems: a survey," *Automatica*, vol. 24, pp. 309–326, 1988.

[6] Batina, I., *Model predictive control for stochastic systems by randomized algorithms*. PhD thesis, Technische Universiteit Eindhoven, 2004.

[7] Bellman, R. E., *Dynamic Programming*. New Jersey: Princeton University Press, 1957.

[8] Bertsekas, D. and Tsitsiklis, J., *Neuro-dynamic programming*. Athena Scientific, 1 ed., 1996.

[9] Blom, H. and Bar-Shalom, Y., "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE transactions on automatic control*, vol. 33, pp. 780–783, 1988.

[10] Boyan, J. A. and Moore, A. W., "Generalization in reinforcement learning: Safely approximating the value function," in *Advances in Neural Information Processing Systems 7*, pp. 369–376, MIT Press, 1995.

[11] Boyd, S. and Vandenberghe, L., *Convex Optimization*. Cambridge University Press, 2004.

[12] Bradtke, S., "Reinforcement learning applied to linear quadratic regulation," in *Adavances in neural information processing systems*, pp. 295–302, 1993.

[13] Chiang, L., Russell, E., and Braatz, R., *Fault detection and diagnosis in industrial systems*. Advanced textbooks in control and signal processing, Springer, 2001.

[14] Choudhury, M. S., Jain, M., and Shah, S., "Stiction - definition, modeling, detection and quantification," *Journal of Process Control*, vol. 18, pp. 232–243, 2008.

[15] Choudhury, M. S., Thornhill, N. F., and Shah, S., "Modeling valve stiction," *Control Engineering Practice*, vol. 13, pp. 641–658, 2005.

[16] Clarke, D., Mohtadi, C., and Tuffs, P., "Generalized predictive control - part 1. the basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.

[17] Costa, O., Fragoso, M., and Marques, R., *Discrete-Time Markov Jump Linear Systems*. Springer, 2005.

[18] Cutler, C. and Ramaker, B., "Dynamic matrix control - a computer control algorithm," in *AIChE Meeting*, (Houston, TX), 1979.

[19] Dassau, E., Bequette, B. W., Buckingham, B., and Doyle-III, F., "Detection of a meal using continuous glucose monitoring: Implications for an artificial β cell," *Diabetes Care*, vol. 31, no. 2, pp. 295–300, 2008.

[20] Eliceche, A. M., Petracci, N. C., Hoch, P., and Brignole, E. A., "Optimal operation of an ethylene plant at variable feed conditions," *Computers & Chemical Engineering*, vol. 19, Suppl., pp. S223–S228, 1995.

[21] Fel'dbaum, A., "Dual control theory, part i," *Automation remote control*, vol. 21, pp. 874–880, 1960.

[22] Fel'dbaum, A., "Dual control theory part iii:," *Automation remote control*, vol. 21, pp. 1453–1464, 1960.

[23] Fel'dbaum, A., "Dual control theory part iii," *Automation remote control*, vol. 22, pp. 1–12, 1961.

[24] Fel'dbaum, A., "Dual control theory part iv," *Automation remote control*, vol. 22, pp. 109–121, 1961.

[25] Gadkar, K., III, F. D., Crowley, T., and Varner, J., "Cybernetic model predictive control of a continuous bioreactor with cell recycle," *Biotechnoloy Progress*, vol. 19, pp. 1487–1497, 2003.

[26] Galbe, M. and Zacchi, G., "A review of the production of ethanol from softwood," *Applied Microbiology and Biotechnology*, vol. 59, pp. 618–628, 2002.

[27] Gordon, G. J., "Stable function approximation in dynamic programming," tech. rep., School of Computer Science, Carnegie Mellon University, 1995.

[28] Hahn-Hagerdal, B., Galbe, M., Gorwa-Grauslund, M., Liden, G., and Zacchi, G., "Bio-ethanol- the fuel of tomorrow from the residues of today," *Trends in Biotechnology*, vol. 24, no. 12, pp. 549–556, 2006.

[29] Hahn-Hagerdal, B., Karhumaa, K., Fonseca, C., Spencer-Martins, I., and Gorwa-Grauslund, M. F., "Towards industrial pentose-fermenting yeast strains," *Applied Microbiology and Biotechnology*, vol. 74, pp. 937–953, 2007.

[30] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statisical Learning: Data Mining, Inference and Prediction*. Springer-Verlag, 2 ed., 2008.

[31] Ho, N., Chen, Z., and Brainard, A., "Genetically engineered *Saccharomyces* yeast capable of effective cofermentation of glucose and xylose," *Applied and environmental microbiology*, vol. 64, no. 5, pp. 1852–1858, 1998.

[32] Hovd, M. and Bitmead, R. R., "Interaction between control and state estimation in nonlinear mpc," in *7th International symposium on dynamics & control of process systems*, (Boston, MA), 2004.

[33] Huang, B., "Bayesian methods for control loop monitoring and diagnosis," *Journal of process control*, vol. 18, pp. 829–838, 2008.

[34] Johnston, L. A. and Krishnamurthy, V., "An improvement to the interacting multiple model algorithm," *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 2909–2923, 2001.

[35] Kaelbling, L. P., Littman, M. L., and Moore, A. W., "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[36] Kaisare, N., Lee, J., and Lee, J., "Simulation-based optimization for nonlinear optimal control," in *IFAC, 15th Triennial World Conference*, (Barcelona, Spain), 2002.

[37] Kesavan, P. and Lee, J. H., "Diagnostic tools for multivariable model-based control systems," *Industrial & engineering chemistry research*, vol. 36, pp. 2725–2738, 1997.

[38] Kim, C., "Dynamic linear models with markov-switching," *Journal of Econometrics*, vol. 60, pp. 1–22, 1994.

[39] Kim, J., Varner, J., and Ramkrishna, D., "A hybrid model of anaerobic e. coli gjt001: combination of elementary flux models and cybernetic variables," *Biotechnology Progress*, vol. 24, no. 5, pp. 993–1006, 2008.

[40] Kompala, D., Ramkrishna, D., Jansen, N., and Tsao, G., "Investigation of bacterial growth on mixed substrates: experimental evaluation of cybernetic models," *Biotechnology and Bioengineering*, vol. 28, pp. 1044–1055, 1986.

[41] Konidaris, G. and Osentoski, S., "Value function approximation in reinforcement learning using the fourier basis," tech. rep., University of Massachusetts Amherst, 2008.

[42] Krishnan, M., Xia, Y., Ho, N., and Tsao, G., "Fuel ethanol production from lignocellulosic sugars- studies using a genetically engineered saccharomyces yeast.," *Fuels and chemicals from biomass*, vol. 666, pp. 74–92, 1997.

[43] Laird, C. D. and Biegler, L. T., "Large-scale nonlinear programming for multi-scenario optimization," in *Modeling, Simulation and Optimization of Complex Processes* (Bock, H. G., Kostina, E., Phu, H. X., and Rannacher, R., eds.), pp. 323–336, Springer Berlin Heidelberg, 2008.

[44] Lee, J. H. and Lee, J. M., "Approximate dynamic programming based approach to process control and scheduling," *Computers & Chemical Engineering*, vol. 30, no. 10-12, pp. 1603–1618, 2006.

[45] Lee, J. H. and Ricker, N. L., "Extended kalman filter based nonlinear model predictive control," *Industrial & Engineering Chemistry Research*, vol. 33, pp. 1530–1541, 1994.

[46] Lee, J. M., Kaisare, N. S., and Lee, J. H., "Choice of approximator and design of penalty function for an approximate dynamic programming based control approach," *Journal of process control*, vol. 16, pp. 135–156, 2006.

[47] Lee, J. M. and Lee, J. H., "Simulation-based learning of cost-to-go for control of nonlinear processes," *Korean Journal of Chemical Engineering*, vol. 21, no. 2, pp. 338–344, 2004.

[48] Lee, J. M. and Lee, J. H., "Approximate dynamic programming-based approaches for input-ouptut data-driven control of nonlinear processes," *Automatica*, vol. 41, pp. 1281–1288, 2005.

[49] Lee, J. M. and Lee, J. H., "Value function-based approach to the scheduling of multiple controllers," *Journal of process control*, vol. 18, pp. 533–542, 2008.

[50] Lee, J. M. and Lee, J. H., "An approximate dynamic programming approach based approach to dual adaptive control," *Journal of process control*, vol. 19, pp. 859–864, 2009.

[51] Lee, W. and Weekman, V., "Advanced control practice in the chemical process industry: a view from industry," *AIChE Journal*, vol. 22, pp. 27–38, 1976.

[52] Lin, Y. and Tanaka, S., "Ethanol fermentation from biomass resources: current state and prospects," *Applied Microbiology and Biotechnology*, vol. 69, pp. 627–642, 2006.

[53] Ljung, L., *System Identification: Theory for the User*. PTR Prentice Hall, 2 ed., 1999.

[54] Loeblein, C. and Perkins, J. D., "Structural design for on-line process optimization: I. dynamic economics of mpc," *AIChE Journal*, vol. 45, p. 1018, 1999.

[55] Logothetis, A. and Krishnamurthy, V., "Expectation maximization algorithms for map estimation of jump markov linear systems," *IEEE transactions on Signal Processsing*, vol. 47, no. 8, pp. 2139–2156, 1999.

[56] L.T. Narraway, J. P. and Barton, G., "Interaction between process design and process control: economic analysis of process dynamics.," *Journal of process control*, vol. 1, pp. 243–250, 1991.

[57] Lu, Y. and Mosier, N. S., "Current technologies for fuel ethanol production from lignocellulosic plant biomass," in *Genetic Improvement of Bioenergy Crops*, Springer New York, 2008.

[58] Lynch, S. M. and Bequette, B. W., "Estimation-based model predictive control of blood glucose in type 1 diabetics: a simulation study," in *Proceedings of the IEEE 27th Annual Northeast Bioengineering Conference*, pp. 79–80, 2001.

[59] Lynch, S. M. and Bequette, B. W., "2002," in *Proceedings of the American Control Conference*, 2002.

[60] Ma, J. and Powell, W., "A convergent recursive least squares policy iteration algorithm for multi-dimensional markov decision process with continuous state and action spaces," in *IEEE Conference on Approximate Dynamic Programming and Reinforcement Learning (part of IEEE Symposium on Computational Intelligence)*, (Nashville, TN), 2009.

[61] Morales-Menendez, R., Freitas, N. d., and Poole, D., "Estimation and control of industrials processes with particle filters," in *American Control Conference*, (Denver, Colorado), 2003.

[62] Morari, M. and Stephanopoulos, G., "Minimizing unobservability in inferential control schemes," *International Journal of Control*, vol. 31, no. 2, pp. 367–377, 1980.

[63] Morari, M. and Stephanopoulos, G., "Studies in the synthesis of control structures for chemical processes. part iii: Optimal selection of secondary measurements within the framework of state estimation in the presence of persistent unknown disturbances," *AICHE Journal*, vol. 26, no. 2, p. 247, 1980.

[64] Murphy, K., "Switching kalman filters," tech. rep., DEC/Compaq Cambridge Research Labs, 1998.

[65] Muske, K. R. and Badgwell, T. A., "Disturbance modeling for offset-free linear model predictive control," *Journal of Process Control*, vol. 12, pp. 617–632, 2002.

[66] Muske, K. and Rawlings, J., "Model predictive control with linear models," *AIChE Journal*, vol. 39, no. 2, pp. 262–287, 1993.

[67] Ormoneit, D. and Sen, S., "Kernel-based reinforcement learning," *Machine Learning Journal*, vol. 49, pp. 161–178, 2002.

[68] Pannocchia, G. and Rawlings, J., "Disturbance modeling for offset-free model predictive control," *AICHE Journal*, vol. 49, pp. 426–437, 2003.

[69] Pavlovic, V., Rehg, J. M., Cham, T.-J., and Murphy, K. P., "A dynamic bayesian network approach to figure tracking using learned dynamic models," in *The Proceedings of the 7th IEEE International Conference on Computer Vision*, vol. 1, (Kerkyra, Greece), pp. 94–101, 1999.

[70] Pena, D. M. d. l., Bemporad, A., and Alamo, T., "Stochastic programming applied to model predictive control," in *44th IEEE Conference on Decision and Control, and the European Control Conference*, (Seville, Spain), pp. 1361–1366, 2005.

[71] Powell, W. B., *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley Series in Probability and Statistics, Wiley-Interscience, 2007.

[72] Prett, D. and Garcia, C., *Fundamental Process Control*. Butterworth, Boston, 1988.

[73] Qin, S. J. and Badgwell, T. A., "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, pp. 733–764, 2003.

[74] Rabiner, L. R., "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[75] Ramakrishna, R., Ramkrishna, D., and Konopka, A., "Cybernetic modeling of growth in mixed, substitutable substrate enviroments: preferential and simultaneous utilization," *Biotechnology and Bioengineering*, vol. 52, pp. 141–151, 1996.

[76] Ramkrishna, D., Kompala, D., and Tsao, G., "Are microbes optimal strategists," *Biotechnology Progress*, vol. 3, no. 3, pp. 121–126, 1987.

[77] Ramkrishna, D., "On modeling of bioreactors for control," *Journal of process control*, vol. 13, pp. 581–589, 2003.

[78] Rawlings, J., Meadows, E., and Muske, K., "Nonlinear model predictive control: a tutorial and survey," in *ADCHEM Conference*, (Kyoto, Japan), pp. 203–214, 1994.

[79] Robertson, D. and Lee, J., "A method for the estimation of infrequent abrupt changes in nonlinear systems," *Automatica*, vol. 34, no. 2, pp. 261–270, 1998.

[80] Roy, B. V., Bertsekas, D. P., Lee, Y., and Tsitsiklis, J., "A neuro-dynamic programming approach to retailer inventory management," in *IEEE Conference on Decision and Control*, vol. 4, pp. 4052–4057, 1997.

[81] Rutquist, P., Breitholtz, C., and Wik, T., "On the infinite time solution to state-constrained stochastic optimal control problems," *Automatica*, vol. 44, no. 7, pp. 1800 – 1805, 2008.

[82] S. Kolas, B. F. and Schei, T., "Noise modeling concepts in nonlinear state estimation," *Journal of Process Control*, vol. 19, pp. 1111–1125, 2009.

[83] Sabes, P., "Approximating q-values with basis function representations," in *Fourth Connectionist Models Summer School*, (Hillsdale, NJ), 1993.

[84] Schuster, S., Fell, D., and Dandekar, T., "A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks," *Nature Biotechnology*, vol. 18, pp. 326–332, 2000.

[85] Schuster, S., Hilgetag, C., Woods, J., and Fell, D., "Reaction routes in biochemical reaction systems: algebraic properties, validated calculation procedure and example from nucleotide metabolism," *Journal of mathematical biology*, pp. 153–181, 2002.

[86] Scokaert, P. and Mayne, D., "Min-max feedback model predictive control for constrained linear systems," *IEEE Transactions of Automatic Control*, vol. 43, no. 8, pp. 1136–1142, 1998.

[87] Shinskey, F., *Feedback Controllers for the Process Industries*. McGraw-Hill, New York, 1994.

[88] Singhal, A. and Seborg, D. E., "Dynamic data rectification using the expectation maximization algorithm," *AICHE Journal*, vol. 46, no. 8, p. 1556, 2000.

[89] Smart, W. and Kaelbling, L., "Practical reinforcement learning in continuous spaces," in *17th international conference on machine learning*, pp. 903–910, 2000.

[90] Smyth, P., "Hidden markov models for fault detection in dynamic systems," *Pattern recognition*, vol. 27, pp. 149–164, 1994.

[91] Song, H.-S., Morgan, J. A., Ho, N. W. Y., Varma, A., and Ramkrishna, D., "Optimization of batch and continuous fermenters for increasing bioethanol productivity using hybrid cybernetic models," in *American Institute of Chemical Engineers Annual Meeting*, (Philadelphia, PA), 2008.

[92] Song, H.-S., Morgan, J. A., and Ramkrishna, D., "Systematic development of hybrid cybernetic models: application to recombinant yeast co-consuming glucose and xylose (in review)," *Biotechnoloy and Bioenegineering (2009)*, 2009.

[93] Srinivasan, R. and Rengaswamy, R., "Control loop performance assessment: 1. a qualitative approach for stiction diagnosis," *Industrial & Engineering Chemistry Research*, vol. 44, pp. 6708–6718, 2005.

[94] Stenman, A., Gustafsson, G., and Forsman, K., "A segementation-based method for detection of stiction in control valves," *International journal of adaptive control and signal processing*, vol. 17, pp. 625–634, 2003.

[95] Sutton, R. S. and Barto, A. G., *Reinforcement learning: an introduction*. Cambridge, MA: MIT Press, 1998.

[96] Tesauro, G., "Practical issues in tempora-difference learning," *Machine Learning Journal*, vol. 8, pp. 257–277, 1992.

[97] Thrun, S. and Schwartz, A., "Issues in using function approximation for reinforcement learning," in *Fourth Connectionist Models Summer School*, (Hillsdale, NJ), 1993.

[98] Tosukhowong, T. and Lee, J. H., "Approximate dynamic programming based optimal control applied to an integrated plant with a reactor and a distillation column with recycle," *AIChE Journal*, vol. 55, no. 4, pp. 919–930, 2009.

[99] Tsitsiklis, J. and Roy, B. V., "Feature-based methods for large scale dynamic programming," *Machine Learning Journal*, vol. 22, no. 1, pp. 59–94, 1996.

[100] Tugnait, J., "Adaptive estimation and identification for discrete systems with markov jump parameters," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 1054–1065, 1982.

[101] Tugnait, J., "Detection and estimation for abruptly changing systems," *Automatica*, vol. 18, pp. 607–615, 1982.

[102] US.DOE, "Breaking the biological barriers to cellulosic ethanol: a joint research agenda," 2005.

[103] Vidal, R., Soatto, S., Ma, Y., and Sastry, S., "An algebraic geometric approach to the identificatino of a class of linear hybrid systems," in *IEEE Conference on Decision and Control*, vol. 1, pp. 281–286, 2003.

[104] Willsky, A. S., "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, pp. 601–611, 1976.

[105] Wong, W. C. and Lee, J. H., "Disturbance modeling for process control via hidden markov models," in *8th International IFAC Symposium on Dynamics and Control of Process Systems*, (Cancun, Mexico), pp. 235–240, 2007.

[106] Wong, W. C. and Lee, J. H., "Realistic disturbance modeling using hidden markov models: applications in model-based process control, journal of process control," *Journal of process Control*, In press.

[107] Wyman, C. E., "Twenty years of trials, tribulations, and research progress in bioethanol technology: selected key events along the way," *Applied Biochemistry and Biotechnology*, vol. 91-93, pp. 5–21, 2001.

[108] Young, J., Morgan, J., Konopka, A., and Ramkrishna, D., "Integrating cybernetic modeling with pathway analysis provides a dynamic, systems-level description of metabolic control," *Biotechnology and Bioengineering*, vol. 100, no. 3, pp. 542–559, 2008.

[109] Young, J. and Ramkrishna, D., "On the matching and proportional laws of cybernetic models," *Biotechnology Progress*, vol. 23, pp. 83–99, 2007.

VITA

Wee Chin Wong was born in Singapore. He graduated from Raffles Junior College in 1997 and served in the Singapore Armed Forces from 1998 to 2000. In 2003, he graduated with first class Honors from the National University of Singapore, where he obtained a Bachelor's Degree in Chemical Engineering. He also spent two semesters on an exchange programme at the University of Edinburgh, Scotland. He interned with Owens Corning, a Fortune 500 company, over the course of two summers in 2006 and 2008. There, he worked on a variety of process control and supply chain problems. In 2008, he received a Masters of Science in Chemical Engineering with a certificate in the Management of Technology from Georgia Institute of Technology. His dissertation was on the "Estimation and Control of Jump Stochastic Systems". He defended in August 2009 and obtained a PhD in Chemical Engineering in December, 2009. Research interests include stochastic optimal control, machine learning, statistical data analysis and optimization.