Stability and Continuity of Nonlinear Model Predictive Control

by

EDWARD SCOTT MEADOWS, JR., B.S., M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN
December 1994

Stability and Continuity of Nonlinear Model Predictive Control

APPROVED BY
DISSERTATION COMMITTEE:

Supervisor: ______________________

____

____

____

____

Acknowledgments

I would like to express my appreciation to my advisor, Dr. James B. Rawlings. His difficult questions have pushed this work forward when it stalled and his insight has allowed me to progress when it seemed impossible. His vision of computing in this profession has provided his research group with the best computer facilities at the University. The work reported in this dissertation would have been impossible without the hardware and software tools he provided.

I would like to thank my committee members, Rafael de la Llave, Jeff S. Shamma, David M. Himmelblau, and Thomas F. Edgar for their contributions. Professors Himmelblau, Edgar and Rawlings have provided a superb atmosphere of collaboration among Chemical Engineering students and faculty.

Of the other students and former students with whom I have worked, several deserve special thanks: Tony Mullins, for listening to my mathematical speculations, John Eaton for masterfully administering our computer facilities and providing `octave`, and Ken Muske, whose contributions provided a foundation for parts of this work.

Edward Scott Meadows, Jr.

The University of Texas at Austin
December 1994

Stability and Continuity of Nonlinear Model Predictive Control

Publication No. _______________

Edward Scott Meadows, Jr., Ph.D.
The University of Texas at Austin, 1994

Supervisor: James B. Rawlings

This work provides an analysis of some important properties of model predictive control (MPC), an algorithm for feedback control of dynamic systems.

This dissertation contains a record of work in four related areas: stability of model predictive control, continuity of model predictive control feedback laws and objective functions, resolution of some implementation issues for model predictive control using linear models, and the use of model predictive control for stochastic systems. It makes connections between some well known results for linear quadratic optimal control, which may be viewed as an MPC method, and dynamic programming.

Sufficient conditions for stability are provided for general nonlinear systems. They include non-negativity of the objective and continuity at the origin. Stability is obtained through a Lyapunov stability argument using the MPC objective as a Lyapunov function. A matrix rank condition related to the constraint set that ensures continuity is also provided.

Continuity of the feedback control law derived using model predictive control and the corresponding objective function can have important consequences for the stability and performance of the closed-loop system. Through use of an unusual example, this dissertation investigates continuity and provides a sufficient condition to ensure that the objective function and feedback control law are continuous with respect to the state.

Some of the results reported here concerning implementation of linear model predictive control are based on previous work by Rawlings and Muske at the University of Texas. The issues discussed herein are technical issues important for applications, including the replacement of a state stability constraint in the original proposal by one that is better suited for numerical implementation, and the replacement of an infinite series of state constraints with an equivalent finite set.

This work also demonstrates that analysis methods from dynamic programming can be used to analyze the model predictive control algorithm and subsume many standard results into a more general and comprehensive theory. This connection has not been explicitly stated in the literature to date and remains a rich topic available for future research.

Two results concerning stochastic or perturbed systems are presented. The first provides conditions under which an asymptotically stable control method can retain its stabilizing ability in the presence of perturbations arising from an exponentially stable state observer. The second examines the performance and demonstrates the suboptimality of model predictive control when applied to certain stochastic systems.

Table of Contents

List of Figures

List of Tables

List of Symbols

For each symbol, the number in parentheses is the section in which the symbol first appears. In some cases, the same symbol may be used differently in different context. These symbols will be referenced more than once below using a different section reference.

Lower case symbols

a, b	—	Class K functions that bound Lyapunov function (4.4)
c	—	Class K function that bounds ΔV (4.4)
c	—	Class K function that upper bounds ΔV (4.4)
c_k	—	Constant in recursive solution to stochastic control problem (5.2.1)
d	—	Right hand side of linear input constraints (4.1)
e_k	—	Perturbation to stable dynamic system (4.4)
f	—	Function in dynamic system equation (1.2)
g	—	Function in system equation for uncontrolled dynamic system (2.1)
h	—	Right hand side of linear state constraints (4.1)
i, j, k, l	—	Integer-valued time indices
$\bar{k}$	—	Bounding index in Theorem 1 (1.4)
k_1	—	In Rawlings-Muske control problem, index at which first state constraints can be active (4.1.3)
k_2	—	In Rawlings-Muske control problem, index at which state constraints no longer can be active (4.1.2)
m	—	Dimension of control space (1.2)
n	—	Dimension of state space (1.2)
p	—	Distributional parameter in stochastic control problem (5.2.1)
r	—	Radius used in stability definitions (2.1)
r	—	Radius in Hermes problem $\sqrt{x^2 + y^2}$ (3.3)
u	—	Arbitrary element of control space (1.3)
u_k	—	Control at time k (1.2)

v_i^* — The i-th optimal control in solution of MPC problem (1.2)
v_i — Open-loop controls (1.2)
v_k — Control in stochastic control problem (5.2.2)
w_k — Random variable in stochastic control problem (5)
x — Arbitrary element of state space (1.3)
$\tilde{x}$ — Reference point in stability definitions (2.1)
$\hat{x}_k$ — State estimate (4.5)
x_k — State at time k (1.2)
$\tilde{x}_k$ — Trajectory of perturbed dynamic system (4.4)
z_i — States resulting from open-loop controls (1.2)

Upper case symbols

A, B — System matrices in linear control problem (2.4.1)
A_f, B_f — Linearization matrices for dynamic system function f (2.3.1)
A_s — Operator for matrix A in stable subspace of A (4.1.1)
$\mathcal{C}$ — Controllability matrix (4.1.4)
D — Control variable constraint matrix (4.1)
E — Expectation operator (5)
F — Composite function for final state (3.6)
F_u — Jacobian matrix of F with respect to the control u (3.6)
H — State constraint matrix (4.1)
J_k^* — Dynamic programming objective function (5.2.1)
J, J' — Function mapping $\Re^n \to \Re$ (2.4.2)
K — Feedback gain matrix (2.3.1)
K_c — Lipschitz bound on feedback control law (4.5)
L — Stage cost (1.2)
L — Lipschitz constant (4.4)
$\mathcal{L}$ — Subset of control space used in Theorem 4 (1.4)
N — Horizon length or number of stages in dynamic programming problem (1.2)
N_{cs} — Constrained stabilizability index (4.1.4)
O_t, O_∞ — Subsets of state space used in Gilbert-Tan algorithm (4.1.2)

P	—	Probability (5.2)
P, P_N	—	Solutions to Algebraic Riccati and Riccati Difference Equations, respectively (2.4.1)
$\bar{P}$	—	Unique stabilizing solution of Algebraic Riccati Equation (2.4.1)
$\bar{Q}_N$	—	Weighting matrix in Fake Algebraic Riccati Equation (2.4.1)
Q, R, S	—	Weighting matrices in LQG and Rawlings-Muske control problem (2.4.1)
R	—	Radius used in stability definitions (2.1)
$\mathcal{S}$	—	Stabilizability matrix (4.1.4)
T	—	Dynamic programming operator
T^N	—	Dynamic programming operator applied N times
U, D	—	Up or down controls for Dreyfus problem (5.1)
U, T	—	Matrix elements of Real Schur decomposition (4.1.1)
U_k	—	Constraint region in Theorem 1 (1.4)
$\mathcal{U}$	—	Input variable constraint set (1.2)
U_s, U_u	—	Stable and unstable partitions of Real Schur decomposition matrix U (4.1.1)
V	—	Lyapunov Function (2.1)
W_α	—	Final state constraint region in Mayne-Michalska control problem (2.3.1)
W_c	—	Class-K function in Keerthi and Gilbert control problem (2.3.2)
$\mathcal{X}$	—	State variable constraint set (1.2)
$\mathcal{X}_j$	—	Nested sets in the state space corresponding to k_1 in Rawlings-Muske control problem (4.3)

Greek letter symbols

α	—	Non-decreasing function used in Lyapunov Stability Theorem (2.1)
Δ	—	Difference between objective functions at successive iterations of dynamic programming equation (2.4.2)
γ	—	Nonnegative function lower bounding L (1.4)
γ	—	Non-decreasing function used in Lyapunov Stability Theorem (2.1)
ϵ	—	State estimate error (4.5)

ϕ_N^* — Optimal MPC objective function (1.2)

ϕ_0 — Final state penalty function (1.2)

ϕ_N — Model predictive control objective function with N stages (1.2)

ψ_N — Positive definite function from stabilizable linear-quadratic system (2.4.2)

λ — Parameter in definition of U_k in Theorem 1 (1.4)

λ — Distributional parameter in stochastic control problem (5.2.1)

μ — Feedback control law considered in Hermes problem (3.1)

π_N — Input sequence with N elements (1.2)

$\tilde{\pi}$ — Element of $\mathcal{L}$ in Theorem 4 (1.4)

π_N^* — Optimal control sequence in MPC problem (1.2)

ρ — Lipschitz constant from Hager Theorem (4.2)

ρ, λ — Constants that define bound on exponentially stable state observer error (4.4)

Ξ — Quadratic weighting matrix for infinite horizon, suboptimal objective (2.4.2)

Other symbols

ΔV — Change in Lyapunov Function as state follows system equation

$\Re$ — The real numbers (1.2)

$\Re^+$ — The non-negative real numbers (1.2)

Chapter 1

Introduction

The goal of this research is to analyze model predictive control (MPC) algorithms for stability and performance.

Model predictive control (MPC) refers to a class of control methods in which a process model is used to select a control input based on predictions of the effect of the control on the process output. Model predictive control is also know as receding horizon control and moving horizon control control. Specific implementations of model predictive control concepts are often identified as distinct control methods in their own right. Examples of these include Linear-Quadratic-Gaussian (LQG) optimal control, Dynamic Matrix Control (DMC) and Generalized Predictive Control (GPC).

The definition of MPC given above is broad and could perhaps even encompass classical control methods involving frequency domain design of linear, single-variable systems. To distinguish MPC from other control design techniques, features of model predictive control methods include the following:

- An model is available that can be used to predict future process output based on currently available information and a specified control trajectory. The form of the model differs among different researchers and practitioners. In this work, this model takes the form of a discrete-time nonlinear state space model in which the current state is assumed available.

- A mathematical expression of a control objective is defined over some fixed time period beginning at some specified initial time. It is of theoretical and practical interest to consider the limiting case as the time period (called the prediction horizon) approaches infinity. This case will be considered in this work.

- The control is determined by seeking the control that minimizes the value of the objective function. The control is implemented and at some future time the problem is solved again. This ability to pose and repeatedly solve the optimization problem as new information is available allows model predictive control to operate as a feedback controller.

- The controller can incorporate constraints on the input and state variables. More than any other, it is this feature that has enabled model predictive control to gain its foothold in industrial applications and is one of its most significant distinguishing features.

The above features can be incorporated into a model predictive control problem statement in various ways. Although discrete-time models are the emphasis of this work, Mayne and Michalska have achieved significant success in their analysis and design of stabilizing, robust and implementable MPC methods for continuous-time systems [40, 41, 42, 47, 50, 51]. Earlier work in continuous-time linear systems by Kleinman [32, 33] and Kwon, Pearson and Kailath [36, 37, 35] revealed the critical significance of final state constraints to ensure nominal stability of model predictive control formulations, a topic which receives considerable attention in this research. Similar work was conducted by Chen and Shaw [9] for nonlinear continuous-time systems.

Early work in model predictive control usually considered continuous-time system. Increased interest in the discrete-time case, coincided with the dramatic drop in the cost of computation since discrete-time formulations are more amenable to computer implementation. Industrial applications developed ahead of academic interest and by the late 1970's model predictive control was being successfully applied on a mostly *ad hoc* basis in industry, with little theoretical justification or analysis.

Many of these early industrial applications used linear convolution or impulse response models. Researchers followed this lead and by the early 1980's had established some theoretical foundation for MPC as it was practiced at that time. A review of MPC using impulse response models is provided by García, *et al.* [18].

Of researchers dealing with the general nonlinear discrete-time case, Keerthi and Gilbert [29, 28, 30, 31] have provided the most detailed analysis to date. They analyze both finite and infinite horizon control schemes and provide sufficient conditions for existence and stability of MPC controllers. Their work is described in more detail in Section 2.3.2. The linear, discrete-time case has been analyzed by Rawlings and Muske [53, 54, 57] at the University of Texas at Austin. The key contribution of their work was to find a parameterization of the problem such that the resulting controller is nominally stable regardless of the choice of tuning parameters. A significant portion of this work extends their results, and is discussed in Chapter 4.

The remainder of this dissertation addresses the following topics:

- Chapter 1 contains a description of the model predictive control formulation that is the basis for this work. The MPC problem is defined for both finite and infinite horizon cases. Although existence of solutions to the model predictive control problem is not the primary purpose of this research, sufficient conditions are provided for existence of both finite and infinite horizon cases.

- Chapter 2 provides the heart of this work with regards to nonlinear MPC. Sufficient conditions for nominal stability of model predictive control are provided. Since necessary and sufficient conditions are not available for the stability result, a clear distinction is made between the sufficient conditions for existence and those for stability. Some of the results of this section have been previously presented in other works by the author [45, 46].

- Chapter 3 discusses the continuity of feedback control laws arising from model predictive control. The bulk of the chapter is devoted to study of an unusual example inspired by Hermes [24] that *requires* a discontinuous control for stability. It also extends the stability theorem of Chapter 2 by providing a way to verify one of the sufficient condition for stability.

- Chapter 4 extends the work of Rawlings and Muske for linear, discrete-time MPC. This work is primarily directed toward resolving outstanding computational issues. Some of these include selecting a minimal set of constraints from a larger set of constraints on the decision variables and improving the numerical implementation of the theory.

- Chapter 5 addresses the issue of suboptimality of the MPC controller when applied to stochastic systems, an issue which has been discussed in the literature, but largely ignored by the community of MPC practitioners.

1.1 Comments Concerning Nomenclature

A mathematical representation of concepts from model predictive control present some difficulties. Solutions of the MPC control problem consist of open-loop control sequences and corresponding state trajectories, yet the actually response of the system does not necessarily follow the open-loop state trajectory. One of the most common ways to express this in the literature has been to use double subscripts such as $\left\{x_{k|k}, x_{k+1|k}, \ldots\right\}$. I have always found this to be imprecise and difficult to follow. In this dissertation, when it is important to distinguish between open-loop states and control and those that are actually implemented, I will use z_i and v_i to be open-loop states and controls, respectively. The time index i will indicate time in the MPC prediction horizon. The symbols x_k and u_k will represent actually implemented states and controls, respectively, with k representing the extrinsic time variable. The state variables z_i and x_k are connected through the initial condition $z_0 = x_k$. When such a distinction is unnecessary for clarity, I will sometimes revert to x_k-u_k notation. Such shifts in notation will be indicated clearly when they occur.

Another notational difficulty involves representing open-loop control sequences when then number of elements is unspecified or infinite. I will generally use the symbol π to represent a sequence of controls $\{v_0, v_1, \ldots\}$. A control sequence that is know to have a specified number of elements N will be represented by π_N. Optimizing sequences will use a $*$ superscript as in π_N^*.

The $*$ superscript will generally denote an optimal variable. For example, an N-stage control objective with initial state x_k, evaluated at some arbitrary π_N may be given by $\phi_N\left(x_k, \pi_N\right)$ with optimum over π_N given by $\phi_N^*\left(x_k\right)$.

Another potentially confusing distinction is found in distinguishing between objective functions between model predictive control and dynamic programming. Since the two coincide for the deterministic systems that are the primary focus of this work, I have purposely blurred the distinction by using ϕ to represent both. When it becomes necessary to distinguish the two in Chapter 5, I use J for dynamic programming and ϕ for MPC.

Finally, as another possible source of confusion, I note that engineering usage rarely distinguishes between functions as distinct objects (f) and their instantiation at some specific point in their domain $(f(x))$. I have not attempted to enforce any such distinctions in this dissertation and hope that it causes no confusion to the discriminating reader.

1.2 Model Predictive Control Problem Definition

Given a nonlinear, deterministic, discrete-time dynamic system

$$x_{k+1} = f(x_k, u_k) \tag{1.1}$$

with f continuous and satisfying $f(0,0) = 0$, we wish to design a stabilizing feedback controller subject to general state and input constraints $x_k \in \mathcal{X} \subseteq \Re^n$ and $u_k \in \mathcal{U} \subseteq \Re^m$. Throughout this dissertation, the goal of the controller is to regulate the state of the system to the origin.

For the model predictive control approach, we construct an N-stage objective function as part of a nonlinear optimization problem whose solution yields a minimizing control sequence that satisfies the constraints within the prediction horizon. The first move of the sequence is implemented, the system moves to a new state and from the new state the optimization problem is re-solved. The method is often called *receding horizon control* because at each time k, the objective function is computed on a horizon that extends N steps beyond k.

An MPC controller is specified by the following:

Horizon Length	N
Stage Cost	L
System Dynamics	f
State Constraint	$\mathcal{X}$
Input Constraint	$\mathcal{U}$
Final State Penalty	ϕ_0

Although left general in the above problem statement, the constraint sets $\mathcal{X} \times \mathcal{U}$ are typically polyhedral regions in $\Re^n \times \Re^m$ defined by linear inequality constraints. In our statement of the model predictive control problem, the constraints do not depend upon the time index; however, $\mathcal{X}$ and $\mathcal{U}$ must be closed and contain the point $(0,0)$ in the interior of $\mathcal{X} \times \mathcal{U}$.

In the above, I use ϕ_0 to represent the final state penalty function. Conceptually, this could be considered a "zero-stage" MPC problem. In subsequent sections, when dynamic programming is introduced as a solution and analysis method for MPC problems, it will be quite natural to use ϕ_0 as an initial condition for a recursive equation that provides the MPC control and objective function.

It is common to use a final state penalty ϕ_0 that is different from the stagecost L. In some formulations, such as linear-quadratic optimal control, the final state penalty function is the determining factor for the stability of the algorithm. Usually ϕ_0 is required to satisfy $\phi_0(0) = 0$ and $\phi_0(x) \geq 0$ for $x \neq 0$. Most of the nonlinear examples provided in this dissertation will use a final state constraint in place of a distinct ϕ_0 and choice of ϕ_0 will not play a role in the stability of MPC.

If we define the control objective function by

$$\phi_N(x_k, \pi_N) = \sum_{i=0}^{N-1} L\left(z_i, v_i\right) + \phi_0\left(z_N\right) \tag{1.2}$$

then the optimal control program takes the form

$$\phi_N^*\left(x_k\right) = \min_{\pi_N}\ \phi_N(x_k, \pi_N) \tag{1.3}$$

$$\begin{array}{rrcl} \text{Subject to:} & z_{i+1} & = & f\left(z_i, v_i\right) \\ & z_i & \in & \mathcal{X} \\ & v_i & \in & \mathcal{U} \\ & [\ z_N & = & 0] \end{array}$$

It is common to include the final state constraint $z_N = 0$ to provide desirable stability properties to the closed-loop response of the system. I also discuss other constraint formulations that will provide stability. In the above problem statement, infinite horizons are allowed as the limiting case for $N \to \infty$. This problem is not always well-defined as the limit of finite horizon problems. Sufficient conditions that provide existence for the infinite horizon case are discussed in subsequent sections.

The current state x_k enters as an external parameter in the nonlinear program. The solution to the problem, written as $\pi_N^* = \left\{v_0^*, v_1^*, \ldots v_{N-1}^*\right\}$, and the optimal objective function ϕ_N^* are functions of x_k, L, f, $\mathcal{X}$, $\mathcal{U}$, N. Since L, f, $\mathcal{X}$, $\mathcal{U}$ are usually understood in the context of a given problem, we will not include these as arguments of the non-optimal objective function $\phi_N(x_k, \pi)$ or the optimal objective function $\phi_N^*(x_k)$.

In the model predictive control method, the entire optimal control sequence is not applied. Instead, the initial control move $u_k = v_0^*$ is implemented and the problem is posed again from a new initial state $x_{k+1} = f\left(x_k, v_0^*\right)$. Since all of the open loop controls π_N^*, including v_0^*, are functions of x_k, $u_k = v_0^*$ is a feedback control law.

1.3 Solution Methods

In this section, I discuss two methods of solving the nonlinear programs posed in the model predictive control problem: batch optimization and dynamic programming.

Batch optimization is the relatively familiar process of posing optimization problems in multiple variables, which are solved numerically by use of computers. This is a flexible process that can be applied to a wide range of optimization problems, not simply model predictive control problems. When we obtain numerical solutions in this work, it will usually be through this process, using the commercially available optimization code `npsol` [20], often with the freely available `octave` [15] as a front end.

The term "batch optimization" is used above to distinguish what is thought of as the normal numerical optimization method from dynamic programming. The dynamic programming method [4, 5, 12, 13] was developed for control of multi-stage stochastic processes. For such systems, it can be proven that dynamic programming provides the minimizing control sequence, which is not true for the batch method. For deterministic systems, which are the primary concern of this work, dynamic programming offers no advantages over batch optimization. Both optimization schemes produce the same result [3].

Because model predictive control problems can be expressed as equivalent dynamic programming problems, the methods and results of dynamic programming can play an important role in the analysis of deterministic model predictive control problems. For this reason, I provide here a brief description of the algorithm. The work of Bertsekas [3, 4] is highly recommended for more detailed discussion.

Consider the N-stage optimization of Equation 1.3. Dynamic programming divides the the optimization into N sub-problems, connected through the following recursive equation:

$$\phi_j^* (x) = \min_u \Big\{ L(x, u) + \phi_{j-1}^* (f(x, u)) \Big\} \qquad j \in \{1, 2, \dots N\} \tag{1.4}$$

The recursion is backward in the index j with an initial condition at $j = 1$ corresponding to the final stage cost of the optimal control problem ϕ_0^*.

The following features will be important for subsequent analyses:

- The DP objective function ϕ_j^* corresponds exactly to that for MPC using the same horizon horizon length. This is a special result that applies to deterministic systems only. Chapter 5 shows that for stochastic systems, the same does not hold. In subsequent sections, I will not always tie the dynamic programming objective to a specified N-stage optimization and will freely use ϕ_N to represent either an MPC objective function of a DP objective function, depending on context.

- The optimization indicated in Equation 1.4 takes a state x as an external parameter; therefore, the solution and the optimal value depend on the state. The functional relationship between state and control values for a given j provide a feedback control law for a model predictive control problem with horizon length j. This highlights a conceptual difference between MPC based on batch optimization and dynamic programming. Since dynamic programming returns optimal *functions*, it might be better to represent Equation 1.4 by

$$\phi_j^*(\cdot) = \min_u \left\{ L(\cdot, u) + \phi_{j-1}^*(f(\cdot, u)) \right\} \qquad j \in \{1, 2, \ldots N\}$$

  This form emphasizes that the problem requires a function as a solution, rather than control at a specific state value; however, the notation of Equation 1.4 is entrenched and I will accept it for the remainder of this dissertations, with occasional reference to the conceptual differences between batch optimization and dynamic programming.

- The minimization indicated in Equation 1.4 is subject to the input and state constraints of the original problem. The standard analyses of dynamic programming incorporate state constraints as additional restrictions on the set of feasible controls. This introduces some additional complexity to the standard formulations in determining whether a specific state constraint formulation actually has an input constraint equivalent. These questions will not be addressed in this work.

Following Bertsekas [4], I will adopt a shorthand notation for the dynamic programming equation by defining the functional T as follows:

$$T(J)(x) = \min_u \{ L(x, u) + J[f(x, u)] \} \tag{1.5}$$

This notation provides a convenient way to analyze finite horizon model predictive control and especially to consider the limiting case as $N \to \infty$. For example, taking $J(x) = \phi_0(x)$, the objective functions of the N-stage MPC problem is given by $T^N(J)(x)$, in which T^N is the composition of function given by

$$T^N(J)(x) = \underbrace{T(T(T\cdots}_{N \text{ times}}(J)))(x)$$

The function J represents a final state weighting function, but the stage cost L is not explicit in this formulation.

Since the dynamic programming algorithm provides the same control results as batch optimization for deterministic problems, the feedback control law for an N-stage model predictive control problem can be represented as

$$u_k = v_0^* = \arg\ T^N(J)(x_k) \tag{1.6}$$

The functional dependence of the control on the state x_k is emphasized in this expression to highlight that this is a feedback control law, exactly as in model predictive control.

1.4 Existence of Solutions to MPC Problems

This dissertation was prepared primarily with an eye toward applications. I have therefore focussed attention toward obtaining sufficient conditions for stability of MPC control laws and not toward sufficient conditions for existence, since existence problems will usually become obvious during the early phases of implementation. Even so, it is helpful to be aware of some conditions that will provide existence, which is not guaranteed for arbitrary x, L, f, $\mathcal{X}$, $\mathcal{U}$, and N.

Before providing sufficient conditions for existence, the following result of Bertsekas [3, 4] will be required:

Theorem 1 (Bertsekas) *If the following conditions hold:*

- *L is non-negative.*
- *$\phi_0(x) = 0$ for all $x \in \Re$.*
- *There exists $\bar{k}$ such that the sets $U_k(\lambda)$ given by*

$$U_k(\lambda) = \left\{ u \in \mathcal{U} \middle| L(x,u) + T^k(\phi_0)(f(x,u)) \leq \lambda \right\} \tag{1.7}$$

  are compact for all $\lambda \in \Re$ and all $k \geq \bar{k}$

Then the infinite horizon objective function satisfies

$$\phi_\infty^*(x) = \lim_{N\to\infty} T^N(\phi_0)(x) \tag{1.8}$$

It must be noted before proceeding that the convergence indicated above is point-wise. Since Bertsekas explicitly allows extended real-valued functions as limit points in Equation 1.8, we must use other information to verify that $\phi_\infty^*(x)$ is finite for states of interest. When the question arises in this dissertation, I show bounded by finding a suboptimal control that satisfies the constraints. This provides an upper bound on the optimal objective. This method is used in Theorem 2 below, for example.

Arbitrary non-negative L meeting the criteria of Section 1.2 will not be able to satisfy the conditions of Theorem 1, but with appropriate choice of L, the following existence theorem follows directly:

Theorem 2 *If L is continuous, $L(0,u) \to \infty$ as $||u|| \to \infty$, and $\mathcal{X}$ and $\mathcal{U}$ are closed, then the existence of a feasible open-loop control $\pi_N = \{v_0, v_1, \ldots v_{N-1}\}$ that yields a bounded objective function $\phi_N(x, \pi_N)$ for bounded initial condition x implies the existence of a bounded open-loop optimal control sequence for this initial condition. This result holds for either finite or infinite horizon problems.*

Proof:

- Finite Horizon: Recall from the Weierstrass Maximum Theorem [39] that a maximum (or minimum) of a nonlinear program is achieved if the objective is continuous and the constraint set is compact.

  Continuity of $\phi_N(x, \pi_N)$ follows from continuity of f and L. From the growth condition on $L(0, u)$ it follows that $\phi_N(x, \pi_N) \to \infty$ as $||\pi_N|| \to \infty$.

  Consider the level set

  $$\mathcal{L} = \{\tilde{\pi}_N \,|\phi_N(x, \tilde{\pi}_N) \leq \phi_N(x, \pi_N)\}$$

  From the hypotheses, $\mathcal{L}$ is closed and bounded and is therefore compact. It is also non-empty. Let C consist of those $\tilde{\pi}_N$ that satisfy the constraints $\mathcal{X}$, $\mathcal{U}$ and the final state constraint $z_N = 0$ (if applicable). This is a closed set; therefore, a minimizing π_N^* exists in the compact set $\mathcal{L} \cap C$.

- Infinite Horizon: Let $U_k(\lambda)$ be as given in Theorem 1. Since L is non-negative, the sets

  $$\tilde{U}_k(\lambda) = \{u \in \mathcal{U} \,| L(x, u) \leq \lambda\} \tag{1.9}$$

  satisfy $U_k(\lambda) \subseteq \tilde{U}_k(\lambda)$ for all k. From the growth condition on L, these sets are compact. This satisfies the conditions of Bertsekas' theorem. For boundedness, we know from hypothesis that there exists a control sequence which provides and upper bound on $\phi_\infty^*(x)$. This is sufficient for existence of a bounded infinite horizon objective and corresponding optimal control sequence. *QED*

Keerthi [28] provided existence theorems for a difference class of finite and infinite horizon problems and points out that the finite horizon case can be proven as a special case of the infinite horizon problem. Since my problem statement is somewhat different from Keerthi's, I have included the above for both infinite and finite horizon cases for completeness.

As demonstrated in the next chapter, stability of the closed-loop dynamic system obtained with model predictive control is closely related to the properties of the stage cost L. Minimal properties of L must include the following:

1. $L(0,0) = 0$

2. There exists non-decreasing $\gamma : [0,\infty) \to [0,\infty)$ such that $\gamma(0) = 0$ and $0 < \gamma(||x,u||) \leq L(x,u)$ for all $(x,u) \neq (0,0)$ ($||\cdot,\cdot||$ represents a norm on $\Re^n \times \Re^m$).

These lead to the following additional properties of L:

3. $L(x,u) > 0 \;\; \forall \;\; (x,u) \neq (0,0)$

4. $L(x,u) \to 0 \Rightarrow (x,u) \to 0$

5. $L(x,u) = 0 \Rightarrow (x,u) = (0,0)$

Property 3 follows from property 2. To demonstrate property 4, it suffices to prove the following equivalent statement: For any $\delta > 0$, there exists $\epsilon > 0$ such that $L(x,u) < \epsilon$ implies $||(x,u)|| < \delta$.

Let $\epsilon(\delta) = \gamma(\delta)$ and take (x,u) such that $L(x,u) < \gamma(\delta)$. Since $L(0,0) = 0$ such an (x,u) always exists. Suppose property 4 is false and $||x,u|| \geq \delta$. Since γ is non-decreasing, we have $\gamma(||x,u||) \geq \gamma(\delta)$. Combining inequalities yields

$$0 < \gamma(||x,u||) \leq L(x,u) < \gamma(\delta) \leq \gamma(||x,u||)$$

which is a contradiction. A similar argument will yield property 5.

With these properties of L, the objective function ϕ_N^* in the model predictive control algorithm with final state stability constraint has the following properties (when it exists):

1. $\phi_N^*(x) \geq 0$

2. $\phi_N^*(x) = 0 \Leftrightarrow x = 0$

3. $\gamma(||x||) \leq \phi_N^*(x)$

4. $\phi_{N+k}^*(x) \leq \phi_N^*(x) \; \forall \; k \geq 0$

The conditions on L have purposely been left general to allow the widest possible selection for applications. In the examples to follow, L has been chosen to be a quadratic or weighted quadratic function of x and u.

1.5 Convergence

Under some sufficient conditions, Keerthi and Gilbert [28, 29, 31, 30], show that the solutions and objective functions for finite horizon model predictive control converge to the infinite horizon case. This will be discussed in greater detail in Section 2.3.2 and especially in Section 2.4. Convergence of MPC costs and controls could provide a basis for design of finite horizon model predictive control problems that would approximate the behavior of the infinite horizon case. Unfortunately, for the general nonlinear MPC problem, convergence cannot be expected. The following example illustrates that MPC solutions using the final state stability constraint do not always converge to the infinite horizon solution. In the example, open-loop states and controls are represented by z_i and v_i, respectively.

Consider the scalar dynamic system whose trajectories follow

$$x_{k+1} = x_k^2 + u_k^2 - \left(x_k^2 + u_k^2\right)^2 \tag{1.10}$$

with no state or control variable constraints and initial state in $[-1, 1]$. Using a quadratic stage cost, the finite horizon model predictive control problem is stated as follows:

$$\phi_N^*(x) = \min_{v_i} \sum_{i=0}^{N-1} \left(z_i^2 + v_i^2\right) \tag{1.11}$$

in which x is an arbitrary state in $[-1, 1]$. The optimization is subject to the final state constraint $z_N = 0$ and initial condition $z_0 = x$. If we take $N = 2$, the final state constraint demands that

$$z_2 = (z_1^2 + v_1^2)[1 - (z_1^2 + v_1^2)] = 0$$

which is satisfied if $z_1 = v_1 = 0$ or $z_1^2 + v_1^2 = 1$. If the latter is chosen then it is easy to show that the cost is given by $1 + x^2$. If we choose the former, then the control law is given by $\sqrt{1 - x^2}$ and $z_1 = z_2 = v_1 = 0$. The cost associated with this control law is

$$\phi^*(x) = \left\{ \begin{array}{ll} 0 & x = 0 \\ 1 & 0 < |x| \leq 1 \end{array} \right] \tag{1.12}$$

The corresponding control law is

$$u(x) = \left\{ \begin{array}{ll} 0 & x = 0 \\ \pm\sqrt{1 - x^2} & 0 < |x| \leq 1 \end{array} \right] \tag{1.13}$$

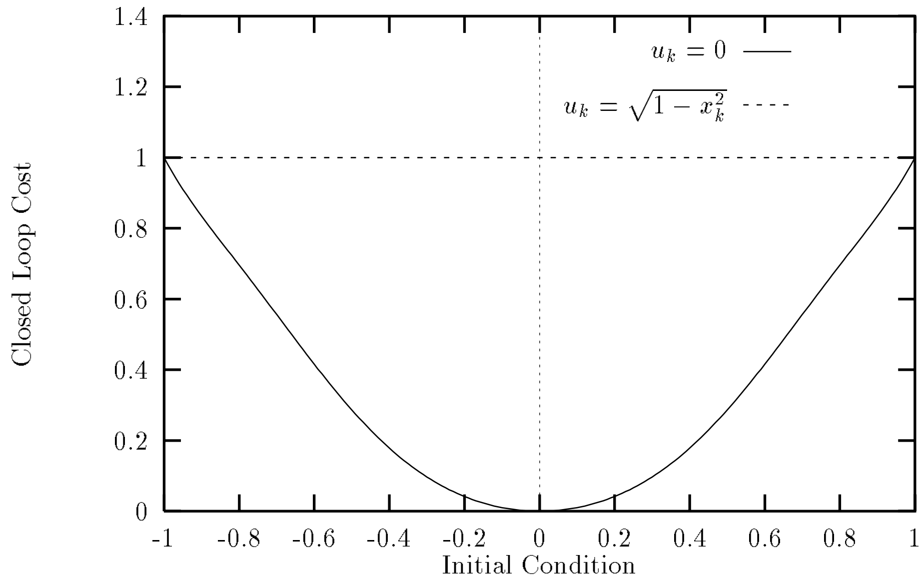

Figure 1.1: Closed-loop Cost vs. Initial Condition

Since this cost is less than $1 + x^2$, it is optimal and is the solution to the problem.

Somewhat surprisingly, the solution to the problem does not depend on the horizon length. The cost to drive the state to zero from any state in $[-1, 1]$ is always 1. The optimal control always forces the state to zero at the first move. There is no cost advantage in delaying the enforcement of the final state stability constraint, since non-zero states occurring after the first stage only add to the cost function without decreasing the penalty associated with the final state constraint.

Consider next the problem of minimizing the objective of Equation 1.11 without the final state stability constraint. If we take the feedback control law $u(x) = 0$, the state evolves according to

$$x_{k+1} = x_k^2 - x_k^4$$

With an infinite horizon cost function $\sum_{k=0}^{\infty} x_k^2$, we find that the cost is finite and shown as the solid line in Figure 1.1. Since the cost of any finite horizon controller that satisfies the final state constraint is lower bounded by the dotted line in Figure 1.1, this indicates that the no finite horizon control will be able to approach the cost of the infinite horizon controller. We may conclude that, in general, a nonlinear model predictive controller defined using the final state stability constraint does not converge to the infinite horizon result.

Chapter 2

Stability Results

This section reviews the basic stability concepts in discrete time, including some significant stability results by other researchers, and provides the main stability result for nonlinear model predictive control. The concluding sections illustrate the utility of the dynamic programming algorithm for analysis of MPC and demonstrate that some linear systems stability results obtained through the Matrix Riccati Difference Equation may be obtained as a special case of the dynamic programming analysis.

2.1 Asymptotic Stability for Discrete-Time Systems

This section reviews the Lyapunov Stability concept for discrete-time systems of the form

$$x_{k+1} = g\left(x_k\right) \tag{2.1}$$

in which $g(0) = 0$. We begin with the following basic definitions which are standard:

Definition 1 *An equilibrium point $\tilde{x}$ is stable if, for every $R > 0$, there exists $r > 0$, such that if $||x_k - \tilde{x}|| < r$, then $||x_{k+l} - \tilde{x}|| < R$ for all $l \geq 0$.*

Definition 2 *An equilibrium point $\tilde{x}$ is asymptotically stable if it is stable, and if in addition there exists some $r > 0$ such that $||x_k - \tilde{x}|| < r$ implies that $||x_{k+l} - \tilde{x}|| \to 0$ as $l \to \infty$.*

Given these definitions, we proceed directly to the main result:

Theorem 3 *Given $V : \Re^n \to \Re^+$ which satisfies the following:*

1. *There exists a nonnegative, nondecreasing function α such that $\alpha(0) = 0$ and*
$$0 < \alpha\left(||x_k||\right) \leq V\left(x_k\right) \tag{2.2}$$
whenever $x_k \neq 0$.

2. *There exists a strictly increasing function γ, continuous at the origin with $\gamma(0) = 0$ and*
$$V\left(x_k\right) - V\left(x_{k+1}\right) \geq \gamma\left(||x_k||\right) \tag{2.3}$$

3. *$V(0) = 0$ and V is continuous.*

Then the process defined by Equation 2.1 is asymptotically stable.

Proof: First to address stability: Choose some $R > 0$. Since V is continuous, there exists $r > 0$ such that $V\left(x_k\right) < \alpha(R)$ for $||x_k|| < r$. Suppose the process is unstable. Then there exists some $l > 0$ such that $||x_{k+l}|| \geq R$ and we have

$$V\left(x_k\right) < \alpha\left(R\right) \leq \alpha\left(x_{k+l}\right) \leq V\left(x_{k+l}\right) \tag{2.4}$$

However, from the hypotheses $V\left(x_k\right) - V\left(x_{k+l}\right) \geq 0$. This is a contradiction; therefore, the process is stable.
We next address convergence: From the hypotheses,

$$V\left(x_k\right) - V\left(x_{k+1}\right) \geq \gamma\left(||x_k||\right) \tag{2.5}$$

With nonnegativity of V, summing from k to $k + l$ gives

$$V\left(x_k\right) \geq V\left(x_k\right) - V\left(x_{k+l}\right) \geq \sum_{j=k}^{k+l} \gamma\left(||x_j||\right) \tag{2.6}$$

Since $\gamma\left(||x_j||\right)$ is non-negative and the partial sums $\sum_{j=k}^{k+l} \gamma\left(||x_j||\right)$ are upper bounded by $V\left(x_k\right)$ (which is not a function of l), we have

$$\lim_{l \to \infty} \gamma\left(||x_{j+l}||\right) = 0 \tag{2.7}$$

Since $\gamma(0) = 0$ and is continuous, this implies that $||x_{k+1}|| \to 0$, which was to be proven. *QED*

It is not uncommon in the literature to cite the above result without the third sufficient condition, perhaps because the seminal paper of Kalman [27] omits it. The conclusion does not follow if V is not continuous at the origin. As a counterexample, consider the dynamic system $x_{k+1} = g(x_k)$ defined for $x_k \in [0, 1]$ in which

$$g(x) = \left\{ \begin{array}{cc} \sin(2\pi x) & x \in \left[0, \frac{1}{2}\right) \\ 0 & \text{otherwise} \end{array} \right] \tag{2.8}$$

A graph of $g(x)$ on $[0, 1]$ is shown in Figure 2.1, which shows that trajectories

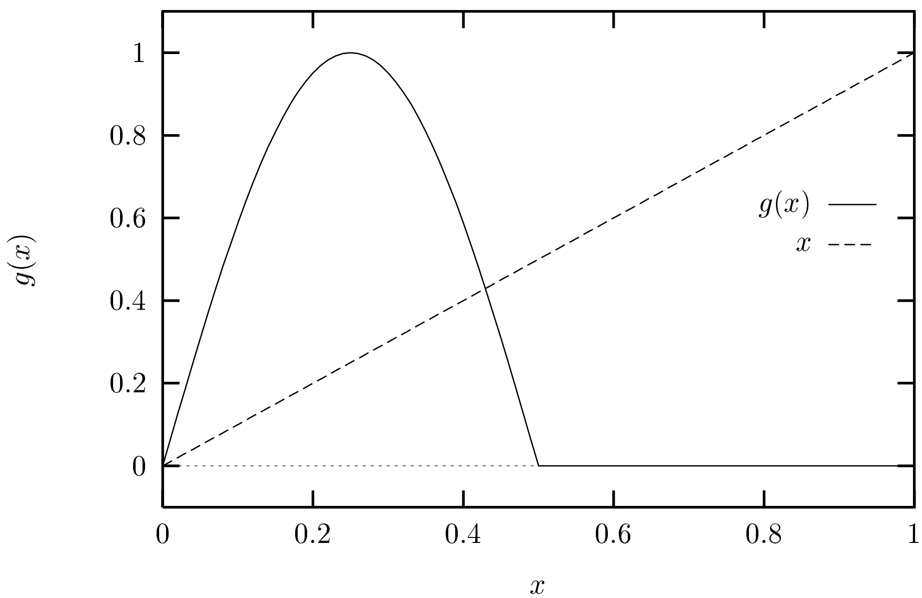

Figure 2.1: System Dynamic Equation $g(x)$ vs. x

with initial conditions in $\left(0, \frac{1}{2}\right)$ must pass through the region $\left[\frac{1}{2}, 1\right]$ before reaching zero. This is *unstable* behavior. Yet if we take $r < 0.4$ and

$$V(x) = \left\{ \begin{array}{cc} 2 - x^2 & x \neq 0 \\ 0 & x = 0 \end{array} \right] \tag{2.9}$$

then the conditions of Theorem 3 are satisfied. Figure 2.2 shows $V(x)$ and $\Delta V(x)$, which clearly satisfy the sufficient conditions of Theorem 3. This counterexample demonstrates one of the shortcomings of the literature to date

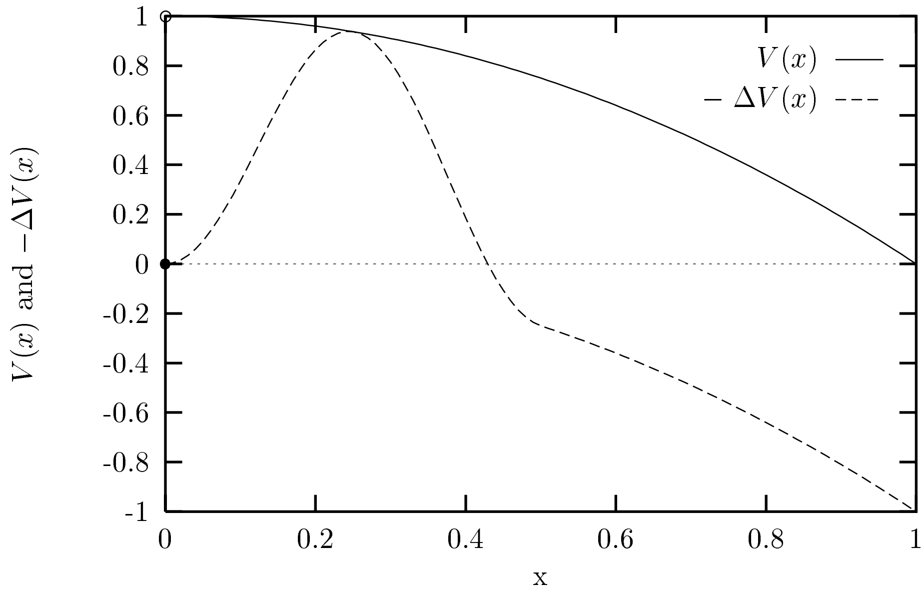

Figure 2.2: Lyapunov Function $V(x)$ and $\Delta V(x)$

in the area of Lyapunov stability theory: Most results in the field are concerned with continuous time systems. During the pre-digital computer age, discrete-time systems were of less concern that those in continuous time. Consequently, a large body of literature developed that neglected the discrete-time case. As discrete-time formulations became increasingly important for computer-based controllers, a few researchers have provided discrete-time results that are usually very close analogs to their continuous-time counterparts.

Unfortunately, few authors provide detailed proofs of the discrete-time results. Considering the importance of discrete-time formulations in computer-based control, it is advisable to carefully check discrete-time results appearing in the literature whose proofs are dismissed as "proceeding as in the continuous-time case."

2.2 Stability Results

Under fairly weak conditions, receding horizon control can provide asymptotic stability. Without further discussion, the main stability result follows:

Theorem 4 *If ϕ_N^* is continuous at the origin, $x = 0$, and L satisfies Properties 1 and 2 above then the origin is an asymptotically stable equilibrium* *point for the dynamic system $x_k = f(x_k, v_0^*)$ with a region of attraction consisting of those points in $\Re^n$ for which a solution to the nonlinear program in Equation 1.3 exists. This result holds in the limit as $N \to \infty$.*

Proof: We consider both finite N and $N \to \infty$.

- **Convergence:**

  Infinite Horizon: Let the infinite horizon objective function ϕ_∞ be represented without subscript by ϕ and its optimal value be represented as $\phi^*(x)$. From the definition of receding horizon control we have the following:

$$\begin{aligned}
\phi^*(x_k) &= L(z_0, v_0^*) + L(z_1, v_1^*) + L(z_2, v_2^*) + \cdots \\
&= L(z_0, v_0^*) + \phi\left[f(z_0, v_0^*), \tilde{\pi}\right] \\
&\geq L(z_0, v_0^*) + \phi^*\left[f(z_0, v_0^*)\right]
\end{aligned}$$

  with $\tilde{\pi} = \{v_1^*, v_2^*, \ldots\}$, z_j given by $z_{j+1} = f(z_j, v_j^*)$ and $z_0 = x_k$. The feasible set for the indicated minimization includes the effects of the constraint sets $\mathcal{X}$ and $\mathcal{U}$. Taking $x_{k+1} = f(x_k, v_0^*)$, the optimal objective function under control law $u_k = v_0^*$ satisfies

$$\phi^*(x_k) - \phi^*(x_{k+1}) \geq L(x_k, v_0^*) \tag{2.10}$$

  This indicates that $\phi^*(x_k)$ is a non-increasing sequence. By positivity of L, $\phi^*(x_k)$ is bounded below by zero and therefore has a limit. As $k \to \infty$, the left-hand side of Equation 2.10 approaches zero and therefore $L(x_k, v_0^*) \to 0$. By Property 4 of L, this implies that the sequence $(x_k, u_k) = (x_k, v_0^*)$ also converges to zero.

  Finite Horizon: Similar to the above, we have the following:

$$\begin{aligned}
\phi_N^*(x_k) &= L(z_0, v_0^*) + L(z_1, v_1^*) + \cdots + L\left(z_{N-1}, v_{N-1}^*\right) \\
&= L(z_0, v_0^*) + \phi_{N-1}\left[f(z_0, v_0^*), \tilde{\pi}\right] \\
&\geq L(z_0, v_0^*) + \phi_{N-1}^*\left[f(z_0, v_0^*)\right]
\end{aligned}$$

  in which $\tilde{\pi}$ in this case is given by $\left\{v_1^*, v_2^*, \ldots v_{N-1}^*\right\}$. From Property 4 of the receding horizon control objective function in Section 1.2, the last inequality implies

$$\phi_N^*(x_k) \quad \geq \quad L(z_0, v_0^*) + \phi_N^*\left[f(z_0, v_0^*)\right]$$

  With this inequality established for the finite horizon objective function, the proof proceeds as for the infinite horizon case.

- **Stability:** Recognizing that the inequality of Property 2 of the receding horizon objective function holds in the limit as $N \to \infty$, the following establishes stability for both finite and infinite horizon cases.

  Choose some $R > 0$. From continuity of ϕ^* at the origin, there exists some $r > 0$ such that $\phi^*(x_k) < \gamma(R)$ for all $||x_k|| < r$. From Equation 2.10, $\phi^*(x_j) \leq \phi^*(x_k)$ for all $j > k$. Suppose that receding horizon control is not stabilizing. Then for some sufficiently small R, $||x_k|| < r$ and $j > k$, $||x_j|| > R$. Since γ is nondecreasing and is a lower bound on ϕ^*, this gives $\gamma(R) \leq \gamma(||x_j||) \leq \phi^*(x_j)$. Combining inequalities gives

  $$\phi^*(x_j) \leq \phi^*(x_k) < \gamma(R) \leq \gamma(||x_j||) \leq \phi^*(x_j) \tag{2.11}$$

  which is a contradiction. Hence receding horizon control is stabilizing.

  Stability combined with convergence implies asymptotic stability. *QED*

2.3 Other Stabilizing Formulations

The results of the previous section provide sufficient, but not necessary, conditions for stability. Several other researcher have provided conditions that ensure stabilizing model predictive controllers. The work of Mayne and Michalska is reviewed below. The stabilizing method of Keerthi and Gilbert is discussed in Section 2.3.2.

2.3.1 The Mayne-Michalska "Dual Mode" Controller

The final state constraint $z_N = 0$ can cause numerical problems in determining an optimal control sequence. Motivated by this fact, Mayne and Michalska [40, 41, 42, 47, 50, 51] have investigated the possibility of relaxing the final state constraint while still retaining stability. Their work involved continuous-time systems, but the method is general and applies to either continuous- or discrete-time systems.

As before, we consider processes described by Equation 1.1:

$$x_{k+1} = f(x_k, u_k)$$

In which $f(0,0) = 0$. For processes whose linearization about the origin exists and is stabilizable, Mayne and Michalska propose a controller in a region near the origin that is based on the linearized controller. Outside the linearized control region, the control is obtained from a finite horizon model predictive controller whose final state is constrained to lie within the linearization region.

The success of the method is based on the following well-known result:

Theorem 5 *Let $A_f = f_x(0,0)$ and $B_f = f_u(0,0)$ be the linearization of f about the origin. If the pair (A_f, B_f) is stabilizable, then there exists a gain matrix K such that the closed loop system*

$$x_{k+1} = f\left(x_k, Kx_k\right) \tag{2.12}$$

is (locally) asymptotically stable.

Although difficult to find in the literature for discrete-time systems, proofs for the continuous-time version are common. An example of such a proof may be found in Sontag [58]. The discrete-time proof presents no special problems and proceeds as in the continuous-time case.

In the statement of the Theorem 5, I included the word "locally" in parentheses. Since asymptotic stability is a local property, the word is unneeded but was included to emphasize that the linearized control law $u_k = Kx_k$ only provides a stabilizing feedback in a neighborhood of the origin. Mayne and Michalska identify that neighborhood as a set $W_\alpha \in \Re^n$, in which α is an adjustable parameter related to the size of W_α.

The controller is considered "dual mode" because the control law changes depending on whether the state is inside or outside W_α. For $x_k \notin W_\alpha$, the control law is given as the first control move of the solution to the finite horizon optimal control problem

$$\min_{\pi_N} \sum_{j=0}^{N-1} L\left(z_i, v_i\right) \tag{2.13}$$

$$\begin{array}{rccc} \text{Subject to:} & z_i & = & f\left(z_i, v_i\right) \\ & z_i & \in & \mathcal{X} \\ & v_i & \in & \mathcal{U} \\ & z_N & \in & W_\alpha \end{array}$$

and for $x_k \in W_\alpha$ the control law is simply $u_k = Kx_k$. Clearly, W_α must be contained in $\mathcal{X}$, be invariant for $f(x_k, K_k)$ and must satisfy $Kx_k \in \mathcal{U}$ for all $x_k \in W_\alpha$. Since the origin must be contained in a neighborhood of $\mathcal{X}$ and $\mathcal{U}$ and W_α can be arbitrarily small, these conditions can be satisfied for $f(x,u)$ satisfying Theorem 5.

The Mayne and Michalska algorithm provides a stabilizing controller with a region of attraction equal to the set of feasible points of the nonlinear program of Equation 2.13. A detailed proof will not be provided here (a continuous-time proof is provided in [50]), but the following argument sketches the fundamental concepts: Through Theorem 5, the linearization provides an asymptotically stable controller whose region of attraction is W_α. Asymptotic stability is shown through this *local* argument only. The basin of attraction is enlarged through the action of the model predictive controller which has the ability to steer the feasible set of states into W_α. Therefore, trajectories starting within the feasible set converge to W_α which is sufficient for asymptotic stability of the entire set of feasible initial conditions.

2.3.2 The Work of Keerthi and Gilbert

Perhaps the most relevant study of receding horizon control using discrete-time systems is that of Keerthi and Gilbert [28, 29, 30, 31]. Their goal was to find a feedback control law for the general nonlinear system of Equation 1.1 that minimizes an infinite horizon cost function subject to constraints. In their view, finite horizon problems are a method of approximating the infinite horizon result. The finite horizon control problems are defined to include the final state stability constraint $z_N = 0$. The class of problems admitted for consideration is narrowed to those for which a control and a horizon N exist such that $z_N = 0$ and

$$\sum_{i=0}^{N-1} ||(z_i, v_i)|| \leq W_c\,(z_0) \tag{2.14}$$

in which W_c is a class K function, i.e., $W_c(0) = 0$ and strictly increasing. (As in Section 1.2, the current state enters the problem through the initial condition $z_0 = x_k$.) Without providing a proof, it would appear that this condition would guarantee that a finite horizon objective function would be continuous at the origin and thus satisfy the conditions of Theorem 4 to obtain a stable controller; however, Keerthi and Gilbert use different arguments in their proof.

Equation 2.14 represents a nonlinear analog to the more familiar linear system controllability criterion of Kalman [26]. Unfortunately, in some cases, it severely narrows that class of systems able to be considered. For example, the constrained linear system of Rawlings and Muske (Chapter 4) is not included, nor is the nonlinear example provided in Chapter 1.

For this restricted class of problems, Keerthi and Gilbert were able to provide conditions for the convergence of finite horizon control problems to the infinite horizon problem and gave an existence proof for the infinite horizon case. A possible reason for the inclusion of such a strong condition on the class of systems being considered is that Keerthi and Gilbert make no distinction between conditions for existence and conditions for stability. Since those conditions are not equivalent, I have provided results that distinguish between then, resulting in a wider class of admissible systems.

2.4 Stability Through Monotonicity in Dynamic Programming

Stability of the closed-loop feedback law has long been known for the infinite horizon Linear-Quadratic (LQ) optimal control solution. The usual derivation involves a dynamic programming method, although it is rare for this connection to be explicitly noted. The stability of infinite horizon LQ optimal control can be demonstrated through an argument involving monotonicity of the Riccati difference equation that follows naturally from the derivation.

In subsequent sections, I review the derivation of the LQ optimal control and then show how the LQ optimal control stability result is a special case of a more general result available for nonlinear systems.

2.4.1 Linear Quadratic Optimal Control

Consider the problem of maintaining the state of the system

$$x_{k+1} = Ax_k + Bu_k \tag{2.15}$$

close to the origin. Closeness to the origin is measured by a weighted square of the Euclidean length of the vector x_k, given by $||x_k||_Q^2 = x_k^T Q x_k$. We want to maintain the state close to the origin without using excessive control action, the size of the control action being measured using $||u_k||_R^2 = u_k^T R u_k$.

We want to choose control inputs so as to minimize the average value of the stage cost over some specified horizon N. Using the notation developed in Chapter 1, we pose the following optimal control program to find $\{v_0, v_1, \ldots v_{N-1}\}$:

$$\begin{aligned} \phi_N^*(x_k) = \quad & \min_{\{v_i\}} \sum_{i=0}^{N-1} \left(z_i^T Q z_i + v_i^T R v_i \right) + z_N^T S z_N \\ \text{Subject to:} \quad & \begin{aligned} z_{i+1} &= Az_i + Bv_i \\ z_0 &= x_k \end{aligned} \end{aligned} \tag{2.16}$$

There are no constraints on the state or control in the classic linear-quadratic problem.

The solution to this problem can be found numerically as a batch optimization over π_N with x_k taken as an external parameter; however, because of the special structure of the LQ problem, a feedback law can be obtained in closed form. The starting point for this derivation is the dynamic programming formulation of Equation 1.4. Since the following summary uses dynamic programming, there can be no confusion between open-loop and closed loop controls and the conventional x-u notation will be used.

$$\phi_j^*(x) = \min_u \left\{ L(x,u) + \phi_{j-1}^*(f(x,u)) \right\} \qquad j \in \{1, 2, \ldots N\} \tag{2.17}$$

Taking $L(x,u) = x^T Q x + u^T R u$, $f(x,u) = Ax + Bu$, and $\phi_0^*(x) = x^T S x$, the dynamic programming equation for a one-stage problem becomes

$$\phi_1^*(x) = \min_u \left\{ x^T Q x + u^T R u + \qquad (Ax + Bu)^T Q (Ax + Bu) \right\} \tag{2.18}$$

Since v is unconstrained, we can find v^* by solving the necessary condition

$$\frac{\partial \phi_1}{\partial u} = 2Rv + 2B^T S (Ax + Bu) = 0 \tag{2.19}$$

to obtain

$$u = \left(R + B^T S B \right)^{-1} B^T S A x \tag{2.20}$$

Substituting Equation 2.20 into Equation 2.18 gives

$$\phi_1^*(x) = x^T \left[Q + A^T S A - A^T S B \left(R + B^T S B \right)^{-1} B^T S A \right] x \tag{2.21}$$

Repeated application of the recursion of Equation 2.17 reveals that $\phi_j^*(x)$ is a quadratic function of its argument. Taking $\phi_N(x) = x^T P_N x$ and substituting into Equation 2.17 provides the classical LQG result, the Riccati difference equation (RDE) in P_N:

$$P_{N+1} = A^T P_N A - A^T P_N B \left(B^T P_N B + R \right)^{-1} B^T P_N A + Q \tag{2.22}$$

with initial condition $P_0 = S$. For any particular horizon length N, the feedback gain is given by $u = K_N x$ in which

$$K_N = -\left(B^T P_{N-1} B + R \right) B^T P_{N-1} A \tag{2.23}$$

In the model predictive control approach, an N-stage MPC problem is solved using batch optimization to provide the control action to be applied. For the LQ problem, this corresponds to a constant controller gain K_N and a stability analysis of the MPC problem becomes one of the stability of the system $x_{k+1} (A + BK_N) x_k$. A particularly readable discussion of this topic is contained in [6], from which remaining material in this section is summarized.

It was proven in Section 2.2 that an infinite horizon model predictive controller is stabilizing. This holds true in the LQ case. Provided the iterations of Equation 2.22 converge, we can form an Algebraic Riccati Equation (given below in Equation 2.24). The following result concerning existence of a solution to the Algebraic Riccati Equation is taken from [6], where it is cited from [11]:

Theorem 6 *Consider the Algebraic Riccati Equation (ARE) associated with an infinite horizon LQ control problem*

$$P = A^T PA - A^T PB \left(B^T PB + R\right)^{-1} B^T PA + Q \tag{2.24}$$

in which

- (A, B) *is stabilizable*
- $\left(A, Q^{1/2}\right)$ *has no unobservable modes on the unit circle*
- $Q \geq 0$ *and* $R > 0$

The there exists a unique, non-negative definite symmetric solution $\bar{P}$ *to Equation 2.24. Furthermore,* $\bar{P}$ *is a unique stabilizing solution, i. e.*

$$A - B \left(B^T \bar{P} B + R\right)^{-1} B^T \bar{P} A \tag{2.25}$$

has eigenvalues strictly within the unit circle.

Conditions under which the P_N of the RDE of Equation 2.22 converge to $\bar{P}$ are provided by the following, cited in [6]:

Theorem 7 *If* (A, B) *is stabilizable,* $\left(A, Q^{1/2}\right)$ *has no unobservable modes on the unit circle and* $P_0 \geq 0$*, then* $P_N \to \bar{P}$ *as* $N \to \infty$*.*

Taken together, Theorem 6 and 7 provide conditions for existence of a finite N such that the LQ optimal control is stabilizing.

A tool used in [6] to analyze the stability of the finite horizon model predictive controller is the Fake Algebraic Riccati Equation:

$$P_N = A^T P_N A - A^T P_N B \left(B^T P_N B + R\right)^{-1} B^T P_N A + \bar{Q}_N \tag{2.26}$$

which is derived from Equation 2.22 by making the substitution

$$\bar{Q}_N = Q - (P_{N+1} - P_N) \tag{2.27}$$

The advantage of this rewriting of Equation 2.22 is that Theorem 6 is immediately applicable to provide sufficient conditions for stability of the N-stage controller:

Theorem 8 *Consider the FARE of Equation 2.26 that defines the matrix $\bar{Q}_N$. If $\bar{Q}_N \geq 0$, $R > 0$, (A, B) is stabilizable, $(A, \bar{Q}_N^{1/2})$ is detectable, then P_N is stabilizing, i.e.,*

$$A - B\left(B^T P_N B + R\right)^{-1} B^T P_N A$$

has eigenvalues all strictly within the unit circle.

This theorem indicates that if we can satisfy a monotonicity condition on solutions of the RDE for some horizon, then stability will follow for all MPC controllers of the same or greater horizon. A substantial body of literature has developed that discusses how to choose an initial condition for the RDE that guarantees monotonicity. Several of these are reviewed in Bitmead *et al.* [6].

One way to stabilize the LQ optimal controller is to set the final state in the horizon to zero, just as for the nonlinear controller of Chapter 2. This can be shown to be equivalent to converting the RDE of Equation 2.22 to a related RDE involving P_N^{-1} using a zero initial condition. This concept is discussed in Thomas [59] in the context of linear observers. The milestone paper by Kwon and Pearson [36, 37] explore the zero end state constraint in detail. The stabilizing formulation of Kleinman [32, 33] provides another variant of the final state constraint for the case $Q = 0$.

Although their work deals with constrained linear systems, Muske and Rawlings [53, 54, 57] provide a stabilizing initial condition that can be applied to the LQ optimal controller to provide a stabilizing feedback gain. Their formulation is reviewed in Chapter 4.

2.4.2 Dynamic Programming Approach

In this section, I demonstrate that analysis using dynamic programming incorporates the results of Bitmead as a special case of a more general theory that is applicable to both linear and nonlinear systems.

It is helpful first to review some of the basic results of dynamic programming theory. Proofs are available from the works of Bertsekas [3, 4]. Recall from Section 1.3, that the dynamic programming algorithm can be expressed using an operator T:

$$T(J)(x) = \min_u \{\ L(x, u) + J[f(x, u)]\ \} \tag{2.28}$$

Using this operator notation, we have the following result, proven in [4]:

Lemma 1 *For any functions $J : \Re^n \to \Re^+$, $J' : \Re^n \to \Re^+$ such that*

$$J(x) \leq J'(x) \ \text{ for all } \ x \in \Re^n$$

then

$$T^N(J)(x) \le T^N(J')(x)$$

for all $x \in \Re^n$ and $N > 0$.

Compare this to the following result of Bitmead *et al.* [7] in which they are referring to the solution to the Riccati Difference Equation (RDE) of Equation 2.22:

Theorem 9 (Bitmead *et al.***)** *If the non-negative definite solution P_N of the RDE is monotonically non-increasing at one time*

$$P_{N+1} \le P_N$$

for some N, then P_N is monotonically nonincreasing for all subsequent times,

$$P_{N+j+1} \le P_{N+j}$$

for all $j \ge 0$.

Theorem 9 follows immediately from Lemma 1 when we recognize that iteration of the RDE is equivalent to a dynamic programming iteration. The corresponding DP approach to Theorem 9 would be as follows:

Theorem 10 *Suppose that*

$$T^{N+1}(J) \le T^N(J) \tag{2.29}$$

Then $T^{N+j+1}(J) \le T^{N+j}(J)$ for all $j \ge 0$.

The proof follows immediately by applying T^j to both sides of Equation 2.29 and invoking Lemma 1.

Lemma 1 was proven by Bertsekas without reference to linear systems theory. This suggests that many of the results in the literature that use properties of Algebraic Riccati Equations and Riccati Difference Equations may have nonlinear generalizations. One such application is to analyze the model predictive control problem for convergence as the horizon approaches infinity. As an example, the dynamic programming approach can be applied to analyze the existence of solutions to the Algebraic Riccati Equation (ARE) of Equation 2.24 as presented in Theorem 6. The proof given by de Souza, Gevers and Goodwin [11] is quite involved, resorting to theoretical justifications based on matrix pencils which cannot be extended to nonlinear systems. With the exception of uniqueness, all of the properties of the solution to the Algebraic Riccati Equation are immediately available through dynamic programming analyses. Consider the following alternative proof:

- Existence: Since the system is stabilizable, there exists a constant feedback gain K such the system $x_{k+1} = (A + BK)\, x_k$ is stable. Because $(A, Q^{1/2})$ has no unobservable modes on the unit circle, we can form the positive definite function

$$\psi_N(x) = \sum_{j=0}^{N} x^T \left[(A + BK)^T\right]^j Q(A + BK)^j \; x$$

  and in the limit as $N \to \infty$, we have

$$\lim_{N\to\infty} \psi_N(x) = x^T \Xi \; x$$

  in which Ξ is the positive definite solution of a matrix Lyapunov equation

$$(A + BK)^T \Xi (A + PK) - \Xi = Q$$

  Using Lemma 1, with $J = 0$, we see that the Riccati Difference Equation can be expressed through

$$T^{N+1}(J)(x) = \min_u \left\{ x^T Qx + u^T Ru + T^N(J)(Ax + Bu) \right\}$$

  with $T^N(J)(x) \le \psi_N(x)$ for all N. Since $\psi_N(x)$ is bounded, converges and each $\psi_N(x)$ is an upper bound for $T^N(J)(x)$, the sequence $T^N(J)(x)$ converges and is less that $x^T \Xi x$. This demonstrates existence.

- Stability: From Theorem 1, we find that $J_\infty(x) = \lim_{k\to\infty} T^N(J)(x)$ satisfies

$$J_\infty(x) = \min_u \left\{ x^T Qx + u^T Ru + J_\infty(Ax + Bu) \right\}$$

  Take $u^* = u^*(x)$ to be the minimizer of the above expression. Then we have

$$J_\infty(Ax + Bu^*) - J_\infty(x) = -\left(x^T Qx + (u^*)^T Ru^* \right)$$

  Since J_∞ satisfies $0 \le J_\infty(x) \le x^T \Xi x$, it is continuous at the origin and thus will provide stability using the argument of Theorem 4.

- Non-negativity: This follows immediately from Lemma 1 taking $J(x) = 0$.

The above discussion shows that dynamic programming theory can provide an alternate path to some well-known results in linear systems theory. For general nonlinear systems, the dynamic programming equation

$$T^N(J)(x) = \min_u \left\{ L(x, u) + T^{N-1}(J)\left(f\left(x, u\right)\right) \right\} \qquad (2.30)$$

with initial condition $J(x) = 0$, can be used to prove stability of infinite horizon model predictive control: If $T^N(J)$ converges to a function ϕ_∞ that is continuous at the origin, then the corresponding control law is asymptotically stabilizing, via the same argument of Theorem 4 from Section 2.2.

The Fake Algebraic Riccati Equation (FARE) method of Section 2.4.1 has a direct analog for nonlinear systems. From the dynamic programming equation of Equation 2.30, take $u_k^* = u^*(x_k)$ to be a minimizing control and $x_{k+1}^* = f(x_k, u_k^*)$. Then we can rewrite Equation 2.30 to become

$$T^N(J)(x_k) = L(x_k, u_k^*) + T^N(J)(x_{k+1}^*) + \left[T^{N-1}(J)\left(x_{k+1}^*\right) - T^N(J)\left(x_{k+1}^*\right)\right] \tag{2.31}$$

Rearranging gives

$$\begin{aligned} T^N(J)(x_{k+1}^*) - T^N(J)(x_k) &= -L(x, u_k^*) + \left[T^N(J)(x_{k+1}^*) - T^{N-1}(J)(x_{k+1}^*)\right] \\ &= -L(x, u_k^*) + \Delta\left(x_k^*, N\right) \end{aligned} \tag{2.32}$$

For provide a sufficient condition for stability, we must have $\Delta(x_{k+1}^* \leq 0$ for all x_{k+1}^*. If $J = 0$, then $\Delta \geq 0$, which will not provide stability.

By analogy to the LQ optimal control result for the Riccati Difference Equation, it may be possible to show that for sufficiently large N, the right-hand side of Equation 2.32 will become non-positive from an arbitrary initial J. Sufficient conditions for this case are currently unknown for the general nonlinear problem. However, it is possible to specify strong conditions on J that provide a receding horizon controller that is stabilizing for *any* horizon length:

Theorem 11 *Let* $L : \Re^n \times \Re^m \to \Re^+$ *and* $J : \Re^n \to \Re^+$ *be continuous and let* J *satisfy* $J \geq T(J)$ *and* $J(0) = 0$. *Then the closed-loop controller defined by* $u(x) = \arg\ T^N(J)(x)$ *is a stabilizing feedback control law for all* $N \geq 1$.

Proof: Since L, J and f are continuous, the objective function in the dynamic programming equation is continuous [2, 17] and $T^N(J)(0) = 0$ for all N. By monotonicity of the DP objective function, $T^N(J) \leq T^{N-1}(J)$ for all k; therefore, the right-hand side of Equation 2.32 is non-positive for all N.

Since $T^N(J)$ is continuous, non-negative, $T^N(J)(0) = 0$, and $T^N(J)(x_{j+1}^*) - T^N(J)(x_j) \leq 0$, $T^N(J)$ satisfies the conditions of Theorem 3 for asymptotic stability. *QED*

Chapter 3

Continuity and Robustness

The stability results of Section 2.2 require continuity of the objective function at the origin. In this chapter, I consider an example that provides a great deal of insight into this condition, since it *requires* a discontinuous controller for stability. MPC provides such a controller *and* it is continuous at the origin. Arising from the study of this example, I provide an easily verifiable sufficient condition for continuity of the objective function, not only at the origin, but at any point in the state space.

3.1 The Hermes Example Problem

The discussion contained in this section requires little distinction between open-loop and closed-loop controls, and I revert back to more familiar notation, taking x and y as scalar states and u_0, u_1, and u_2 to be open-loop controls.

Consider the dynamic system defined by the following two scalar equations:

$$x_{k+1} = x_k + u_k \tag{3.1}$$
$$y_{k+1} = y_k + u_k^3 \tag{3.2}$$

This example is motivated by the following continuous time example by Hermes [24]:

$$\begin{aligned} \dot{x} &= u \\ \dot{y} &= x \\ \dot{z} &= x^3 \end{aligned}$$

The Hermes example violates the onto condition of Brockett [8] for smooth feedback stabilization. Eaton [14] provides numerical results indicating that a continuous-time version of model predictive control can asymptotically stabilize this system. As far as I am aware, conditions analogous to Brockett's continuous-time results are not available for discrete-time systems.

Like Hermes' continuous-time example, the discrete-time system of Equations 3.1 and 3.2 has no continuous feedback law $u_k = \mu(x_k, y_k)$ that is asymptotically stabilizing. The argument that demonstrates this is similar to one presented by Coron [10]. First note that any stabilizing control law must allow both positive and negative values. If the control is strictly positive, trajectories originating in the first quadrant move away from the origin under positive control action. If the control is strictly negative, trajectories originating in the third quadrant also move away from the origin. Yet $\mu(x_k, y_k)$ cannot be zero for any nonzero (x_k, y_k). If it were, then this point would be a fixed point of the dynamic mapping and trajectories containing this (x_k, y_k) would not converge to the origin. We have the situation in which the feedback control law must assume both negative and positive values away from the origin, yet must be zero nowhere away from the origin. Therefore, the feedback control law must be discontinuous. This leads to the following:

Theorem 12 *There exist discrete-time, controllable dynamic systems that admit no asymptotically stabilizing feedback control law that is continuous in the state.*

An outline of the proof is contained in the above discussion. That the system of Equations 3.1 and 3.2 is controllable (the state can be driven to zero from any initial state with suitable choice of controls) is demonstrated in Section 3.2 below.

3.2 Choice of Horizon Length

We seek a controller to regulate the system to the origin. There are no state or input constraints except for the stability constraint $(x_N, y_N) = 0$. To apply model predictive control to this example, we must choose a horizon length.

First, we consider whether a one-step horizon is acceptable. The final state constraint requires the following equations to be satisfied:

$$\begin{aligned} x_0 + u_0 &= 0 \\ y_0 + u_0^3 &= 0 \end{aligned}$$

This constraint is only feasible along the curve $y_0 = x_0^3$. Model predictive control will provide a stabilizing control for the nominal plant for initial conditions along this curve. We would prefer a result with wider applicability; therefore, we increase the horizon length to $N = 2$. The final state constraint then requires an open-loop control profile as follows:

$$\begin{aligned} u_0 &= -\frac{1}{6} \pm \sqrt{\frac{3\left(4y_0 - x_0^3\right)}{x_0}} \\ u_1 &= -\frac{1}{6} \mp \sqrt{\frac{3\left(4y_0 - x_0^3\right)}{x_0}} \end{aligned}$$

With this choice of horizon length, a feasible model predictive control is now available for those points in the plane that allow nonnegative values for the radical in the above equations. This region is bounded by the y-axis and the curve $y_0 = x_0^3/4$. We have succeeded in increasing the domain of allowable initial conditions, but still cannot define a feedback law for all points in the x-y plane; in fact, not even a neighborhood of the origin can be included. We must consider a longer horizon and so choose $N = 3$.

With this choice, we have three open-loop control moves to achieve the final state constraint. Unlike the cases with $N = 1$ or $N = 2$, we now have more unknowns than equations, freeing one variable to be used to minimize the objective. For this choice of N, an open loop control is given by $\{u_0, u_1, u_2\}$. If we solve the final state constraint for u_1 and u_2 in terms of u_0, x_0 and y_0, we obtain the following:

$$u_1 = \frac{1}{6}\left(-3x_0 - 3u_0 \pm \sqrt{\frac{3\left(3u_0^3 - 3u_0^2x_0 - 3u_0x_0^2 - x_0^3 + 4y_0\right)}{u_0 + x_0}}\right) \tag{3.3}$$

$$u_2 = \frac{1}{6}\left(-3x_0 - 3u_0 \mp \sqrt{\frac{3\left(3u_0^3 - 3u_0^2x_0 - 3u_0x_0^2 - x_0^3 + 4y_0\right)}{u_0 + x_0}}\right) \tag{3.4}$$

The open-loop controls will be real only for those values of x_0, y_0 and u_0 that make the expression inside the radicals positive. By inspection, there exists large positive or large negative u_0 to achieve this for any x_0 and y_0. (From Theorem 1, this indicates that a solution to the problem exists with proper choices of L.)

Because feasible choices of u_0 must be real and bounded, the feasible regions for u_0 are bounded by surfaces in $\Re^3$ defined by

$$3u_0^3 - 3u_0^2x_0 - 3u_0x_0^2 - x_0^3 + 4y_0 = 0 \tag{3.5}$$

$$u_0 + x_0 = 0 \tag{3.6}$$

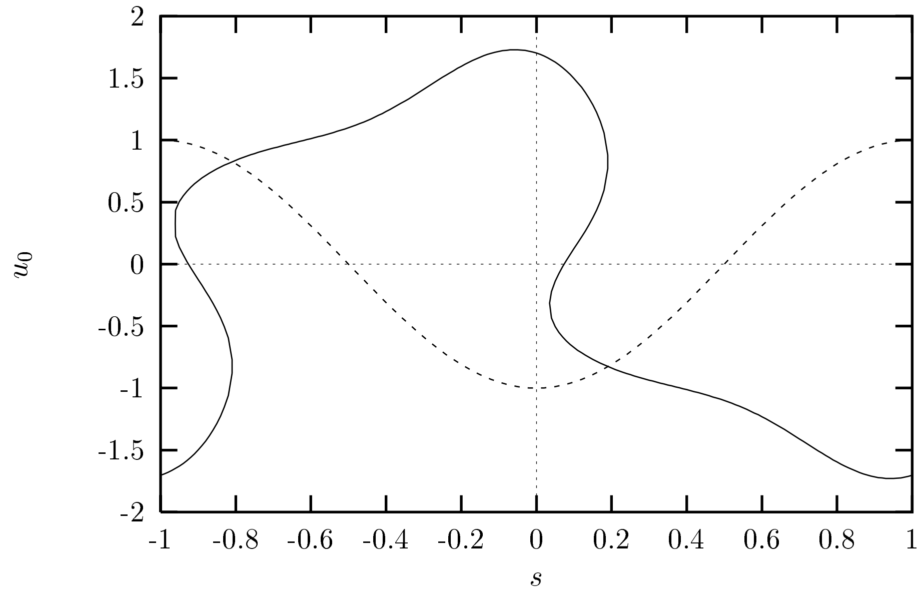

Figure 3.1: Feasible Control Regions Along Unit Circle

Figure 3.1 illustrates how the boundaries of the feasible region can induce discontinuities. The curves indicate the boundaries of the feasible control moves for the optimal 3-move controller for (x_0, y_0) along the unit circle parameterized by $(x_0, y_0) = (\cos \pi s, \sin \pi s)$. The solid curve in Figure 3.1 represents the solution curve to Equation 3.5 and the broken curve is that for Equation 3.6. The feasible regions are those where u_0 is either above or below both curves, i.e., the signs of the left-hand side of Equations 3.5 and 3.6 are either both positive or both negative.

To gain an intuitive understanding of how model predictive control produces a discontinuous feedback law, consider an objective function with a heavy penalty on the control. Such a controller would seek a small u_0 for a given (x_0, y_0) along the unit circle. At the point in Figure 3.1 corresponding to $s = 0$, we would expect u_0 to be close to the boundaries of the feasible region to keep u_0 small. As s increases, suddenly small u_0's become feasible and will be preferred because of the large penalty on u_0.

3.3 Stability of Receding Horizon Control Law

Having established that the system in Equations 3.1 and 3.2 does not admit a continuous, asymptotically stabilizing control, we show that a model predictive controller can asymptotically stabilize this system. We choose a quadratic objective function to form the following nonlinear program:

$$\phi_3^*(x_0, y_0) = \min_{u_0,u_1,u_2} \sum_{j=0}^{2} \left(x_j^2 + y_j^2 + u_j^2 \right) \tag{3.7}$$

$$\begin{array}{lrcl rcl}
\text{Subject to:} & x_0 & = & \text{given} & y_0 & = & \text{given} \\
& x_1 & = & x_0 + u_0 & y_1 & = & y_0 + u_0^3 \\
& x_2 & = & x_1 + u_1 & y_2 & = & y_1 + u_1^3 \\
& x_3 & = & x_2 + u_2 & y_3 & = & y_2 + u_2^3 \\
& & = & 0 & & = & 0
\end{array}$$

By this choice of L we satisfy the conditions of Theorem 1 to guarantee existence of an optimum. We show that the optimal objective function for this problem, $\phi_3^*(x_0, y_0)$, is continuous at the origin and therefore the model predictive controller obtained from the nonlinear program is asymptotically stabilizing by Theorem 4. To show continuity at the origin, we show that the optimal model predictive objective function is bounded below and above by continuous functions that are zero at the origin. This is sufficient for continuity of the optimal objective function at the origin.

Recall from Equations 3.3 and 3.4 that feasibility of the nonlinear program of Equation 3.7 was determined by the following term being real-valued

$$\sqrt{\frac{3u_0^3 - 3u_0^2 x_0 - 3u_0 x_0^2 - x_0^3 + 4y_0}{u_0 + x_0}} \tag{3.8}$$

We can guarantee a feasible point for the constraints if we choose u_0 to make the argument of the above radical non-negative. This will be achieved for all x_0 and y_0 if we choose

$$u_0 = r + 2\sqrt[3]{r} \tag{3.9}$$

in which $r = \sqrt{x_0^2 + y_0^2}$. Substituting this u_0 into Equations 3.3 and 3.4 provides a feasible open-loop control that allows us to evaluate the objective function of 3.7. At $r = 0$, u_1 and u_2 are not defined from Equation 3.9 because both the numerator and denominator are zero. We define the value of u_1 and u_2 as equal to their limits as $r \to 0$ to provide a continuous open loop control. This gives $u_1 = u_2 = 0$ for $(x_0, y_0) = 0$ and fully defines the open-loop control. Some observation about this (non-optimal) open-loop control and the resulting objective include the following:

- u_0, u_1, $u_2 \to 0$ as $r \to 0$.

- The suboptimal open-loop control profile is continuous for all (x_0, y_0). Therefore, the suboptimal objective function evaluated with this input sequence, which is a composition of continuous functions, is continuous in (x_0, y_0).
- The objective function is zero at $(x_0, y_0) = 0$ with the suboptimal open-loop trajectory defined by Equations 3.3, 3.4 and 3.9.
- The suboptimal objective function corresponding to Equations 3.3, 3.4 and 3.9 is an upper bound on ϕ_3^* for all (x_0, y_0).
- The optimal objective value ϕ_3^* is lower bounded by $x_0^2 + y_0^2$.

Since ϕ_3^* is both upper and lower bounded by continuous functions whose values are zero at the origin, ϕ_3^* must be continuous at the origin. The nonlinear program of 3.7 satisfies the sufficient conditions of Theorem 4; therefore the model predictive feedback law defined by the solution of the nonlinear program is asymptotically stabilizing. Since feasible solutions are available for all (x_0, y_0), the region of attraction is all of $\Re^2$.

3.4 Numerical Results

To examine the discontinuous behavior of the model predictive feedback law, we consider initial conditions (x_0, y_0) on the unit circle and examine the optimal feedback and optimal objective function as functions of the state.

Figure 3.2 clearly shows that the feedback control is discontinuous in the state, as must be any stabilizing feedback. Figure 3.3 shows the corresponding objective function. The locations of the discontinuities indicated may change with r, but remain present regardless of distance to the origin ($r = 0$).

Figure 3.4 shows the closed-loop behavior of the system under model predictive control for two points along the unit circle adjacent to the discontinuity shown in Figure 3.2 at approximately $s = -0.47$. The control actions produce two trajectories that initially diverge widely because of the relatively large difference in the initial control input. This wide divergence might be expected to decrease for longer horizon length. For the short horizon chosen, $N = 3$, the controller can be expected to be more aggressive than a controller based on longer horizons.

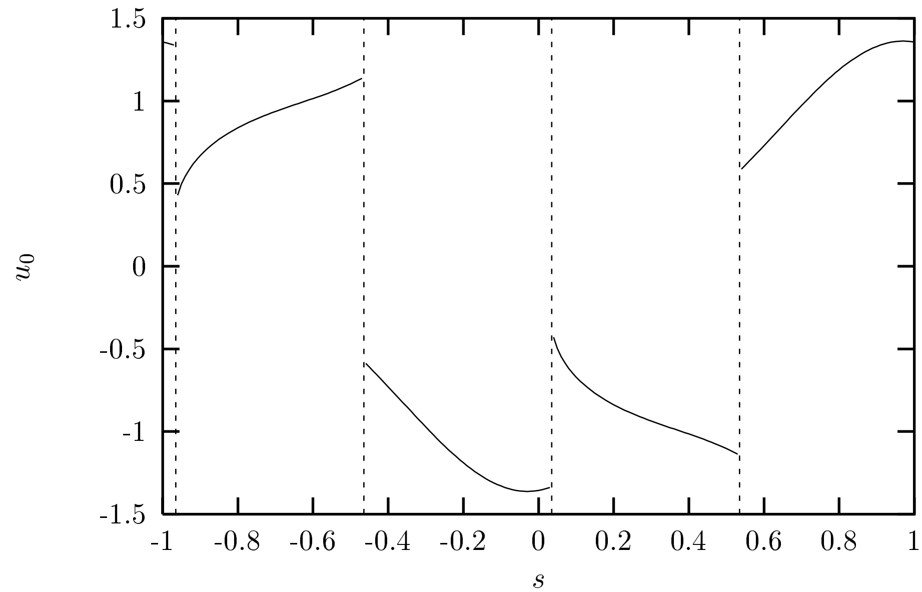

Figure 3.2: Optimal Feedback versus State Along Unit Circle

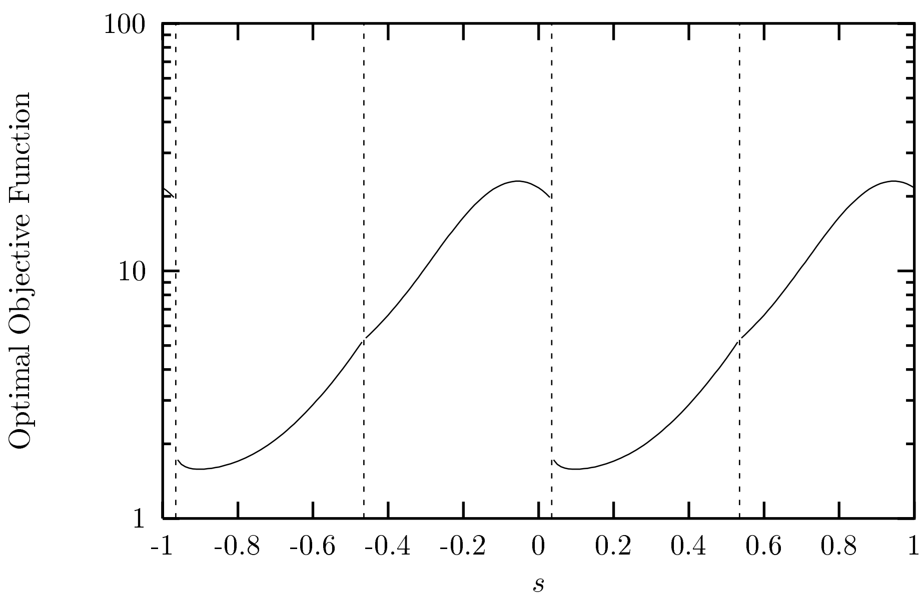

Figure 3.3: Receding Horizon Objective versus State Along Unit Circle

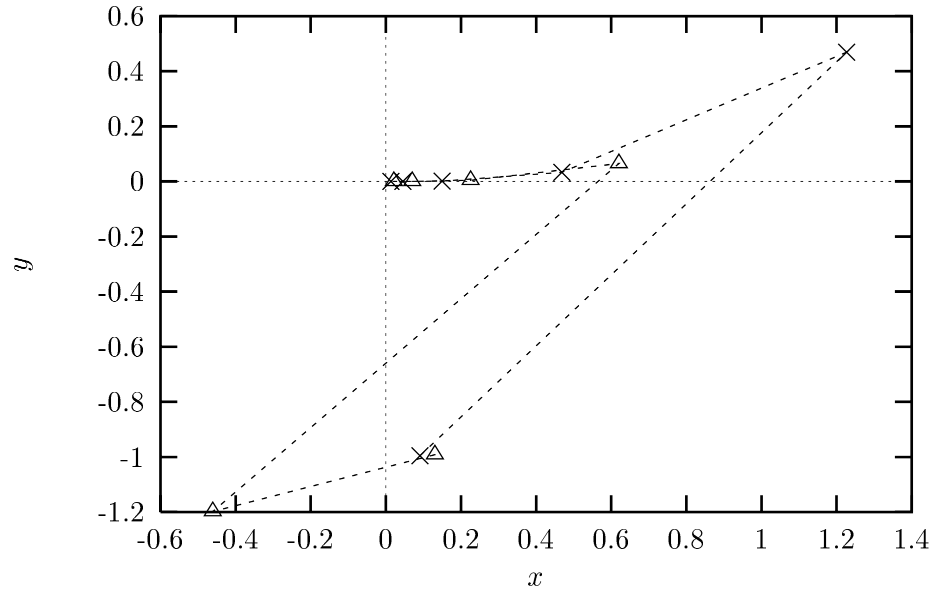

Figure 3.4: Closed–Loop State Trajectories

3.5 Effect of Horizon Length

The idea that longer horizons produce less aggressive control action tends to be borne out be experimental evidence; however, the effects are most pronounced for small N. For N greater than about 6, the objective function and corresponding optimal control input show little change, indicating a convergence of the algorithm for this system as N increases. Figure 3.5 shows the objective value for $N = 3, 4, 5,$ and 7. (Experimental data were available for $N = 6$, but the difference between $N = 6$ and $N = 7$ was so small that it rendered the figure difficult to distinguish.) Interestingly, by enlarging a portion of Figure 3.5, shown in Figure 3.6, it appears that the discontinuities observed for small positive s have disappeared from view for $N = 7$.

The discontinuities in the control are clearly visible for all N tested, as demanded for stability. The first control move u_0 for each N is shown in Figure 3.7.

In view of the results of Chapter 2, it is reasonable to ask whether a final end state penalty can replace the final state stability constraint. From Chapter 2, one way to ensure stability is to choose a final state penalty function J that satisfies

$$J(x) \geq T\left(J\right)(x) = \min_{u} \left\{L(x,u) + J\left(f(x,u)\right)\right\} \tag{3.10}$$

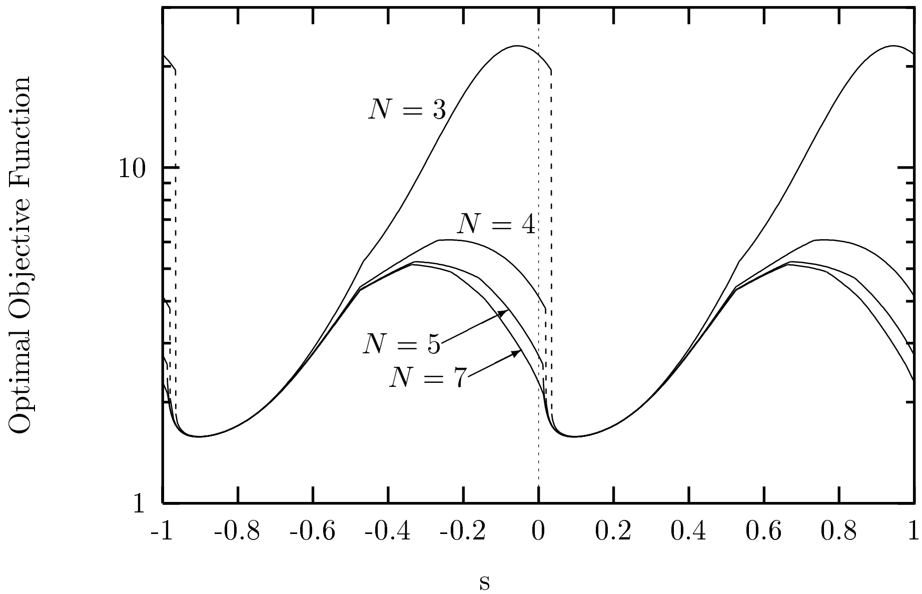

Figure 3.5: Objective Function for Various Horizon Lengths

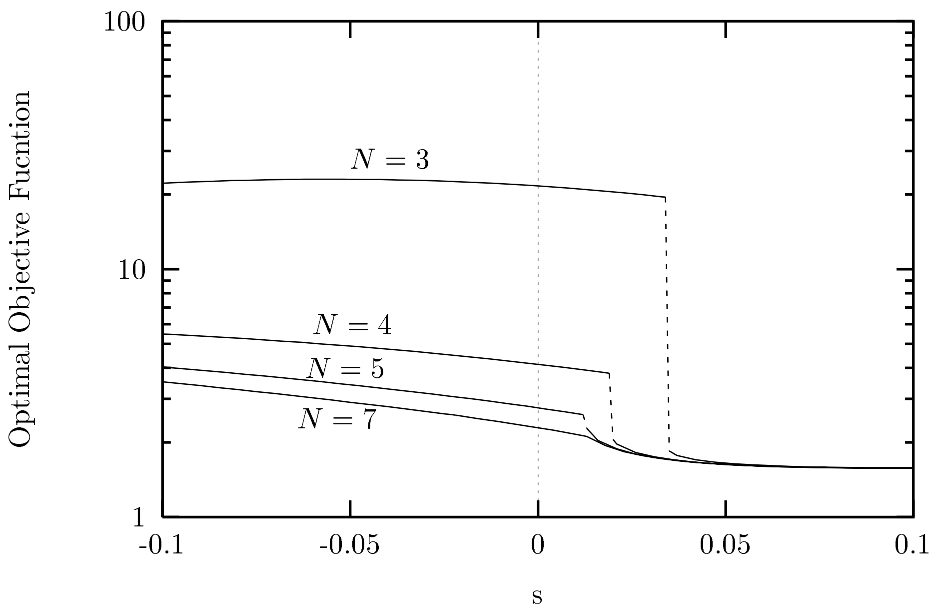

Figure 3.6: Objective Function Various Horizon Lengths: Close View of Discontinuities

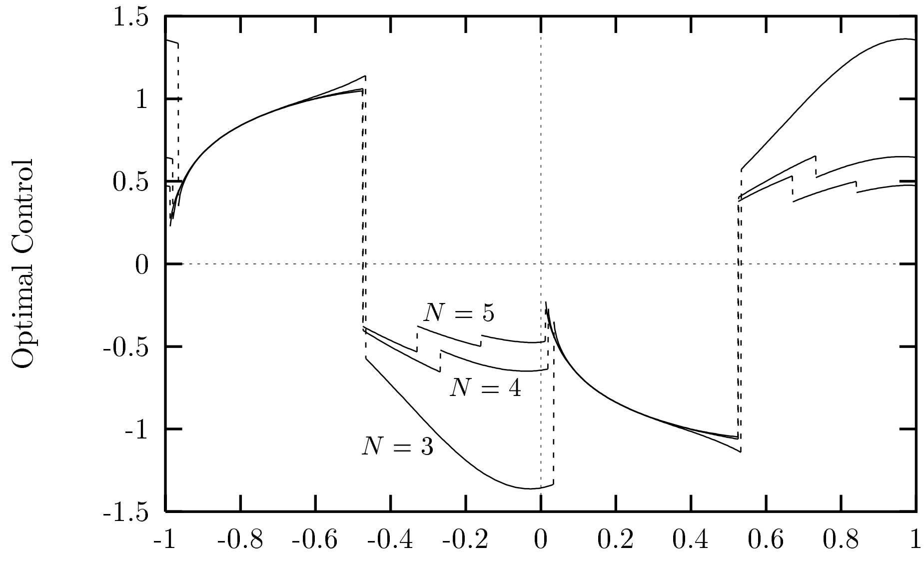

Figure 3.7: First Control Input u_0 For Various N

An obvious problem with this method is that the candidate J must be discontinuous. If a continuous J were chosen, the theorem of Berge (below, Section 3.6) indicates that the control would be continuous, which from Section 3.1, cannot be stabilizing. It may be that any J satisfying the inequality of Equation 3.10 is discontinuous for this system. More research in the properties of this inequality is indicated to better characterize solutions.

3.6 Analysis of Discontinuities

Since the stability result depends on the continuity of the objective function, it would be useful to provide easily verifiable sufficient conditions to determine whether the objective were continuous at the origin. This would obviate the kind of detailed analysis such as contained in Section 3.3. A highly relevant result is that of Berge [2], which is cited by Fiacco [17] in the discussion of continuity of objective functions for nonlinear programs. (I use Berge's original terminology and nomenclature, which may not be consistent with other usage in this dissertation.)

Theorem 13 (Berge) *Let Y and X be topological spaces. If ϕ is a continuous numerical function in Y and Γ is a continuous mapping of X into Y such* *that, for each x, $\Gamma x \neq \emptyset$, then the numerical function M defined by*

$$M(x) = \max \{\phi(y) \,|\, y \in \Gamma x\}$$

is continuous in X and the mapping Φ defined by

$$\Phi x = \{y \,|\, y \in \Gamma x, \phi(y) = M(x)\}$$

is an upper semi-continuous mapping of X into Y.

It must be noted that Berge's notion of semi-continuity of point-to-set maps is not equivalent to the standard definition for single-valued functions. See Appendix A for more discussion of Berge's semi-continuity concept.

It is clear that by consideration of $-\phi(y)$, we can convert the max operation of Berge's theorem into an equivalent minimization problem and retain the substance of the theorem. If we identify the model predictive control objective function with Berge's ϕ, the initial states with his x and the open-loop optimal control sequence with his y, this result can be used to provide sufficient conditions for continuity of the model predictive control objective function with respect to the state.

As discussed in Section 1.4, continuity of the non-optimal, unconstrained MPC objective function follows if we choose a continuous stage cost L, such as a quadratic, since the objective is a composition of continuous functions L and f. From Berge's theorem, we also need continuity of the constraints with respect to the initial conditions.

The constraint regions $\mathcal{X}$ and $\mathcal{U}$ do not vary with the initial conditions; therefore, they are continuous with respect to the initial state. The remaining question is whether the final state constraint $z_N = 0$ defines a constraint region for the controls that is continuous.

The final state constraint implicitly defines a relationship between the control and the initial state through the state equations. If we define F through the following

$$F(z_0, v_0, v_1, \ldots v_{N-1}) = f(f(\ldots f(z_0, v_0) \ldots v_{N-2}) \, v_{N-1}) \tag{3.11}$$

(For generality, I revert back to the open-loop optimal notation using z_i and v_i.) The final state stability constraint is expressed as $F(z_0, v_1, \ldots v_{N-1}) = 0$, which implicitly defines a point to set mapping from $\Re^n$, the state space, into $\Re^{m \times N}$, the Cartesian product of the control space over N control moves. Sufficient conditions for the existence of a continuous implicit mapping between z_0 and $\{v_0, v_1, \ldots v_{N-1}\}$ can be obtained from the Implicit Function Theorem. For this theorem to apply in this case, the matrix Jacobian

$$\{F_u\}_{ij}\Big|_{(z_0, v_0^*, v_1^*, \ldots v_{N-1}^*)} = \left.\frac{\partial F_i}{\partial v_j}\right|_{(z_0, v_0^*, v_1^*, \cdots v_{N-1}^*)} \tag{3.12}$$

must have full rank for any optimal solution that satisfies the final state stability constraint.

To gain more insight into the Hermes example, consider the points of discontinuity observed in the optimal objective shown in Figure 3.5. A necessary condition for a discontinuity is that F_u lose full rank. Taking the case $N = 3$ only, F and F_u are given by

$$F = \begin{bmatrix} x_0 + u_0 + u_1 + u_2 \\ y_0 + u_0^3 + u_1^3 + u_2^3 \end{bmatrix} \tag{3.13}$$

$$F_u = \begin{bmatrix} 1 & 1 & 1 \\ 3u_0^2 & 3u_1^2 & 3u_2^2 \end{bmatrix} \tag{3.14}$$

(The variables x_0, y_0, u_0, u_1 and u_2 are as described in Section 3.1.) The Jacobian matrix loses rank whenever $u_0^2 = u_1^2 = u_2^2$. Therefore, along the unit circle, a discontinuity in the objective value can be caused by the final state stability constraint only if the following conditions are satisfied:

$$x_0 + u_0 + u_1 + u_2 = 0 \tag{3.15}$$
$$y_0 + u_0^3 + u_1^3 + u_2^3 = 0 \tag{3.16}$$
$$u_0^2 - u_1^2 = 0 \tag{3.17}$$
$$u_0^2 - u_2^2 = 0 \tag{3.18}$$
$$x_0^2 + y_0^2 = 1 \tag{3.19}$$

Of the 24 separate solutions to this system of equations, eight are real-valued. The real solutions are shown below:

x_0	y_0	u_0	u_1	u_2
-0.994027	-0.109132	0.331342	0.331342	0.331342
-0.826031	-0.563624	-0.826031	0.826031	0.826031
-0.826031	-0.563624	0.826031	-0.826031	0.826031
-0.826031	-0.563624	0.826031	0.826031	-0.826031
0.826031	0.563624	0.826031	-0.826031	-0.826031
0.826031	0.563624	-0.826031	-0.826031	0.826031
0.826031	0.563624	-0.826031	0.826031	-0.826031
0.994027	0.109132	-0.331342	-0.331342	-0.331342

These solutions define four unique points on the unit circle in the x-y plane. For comparison to Figures 3.5, the four points correspond to $s = 0.035$, $s =$

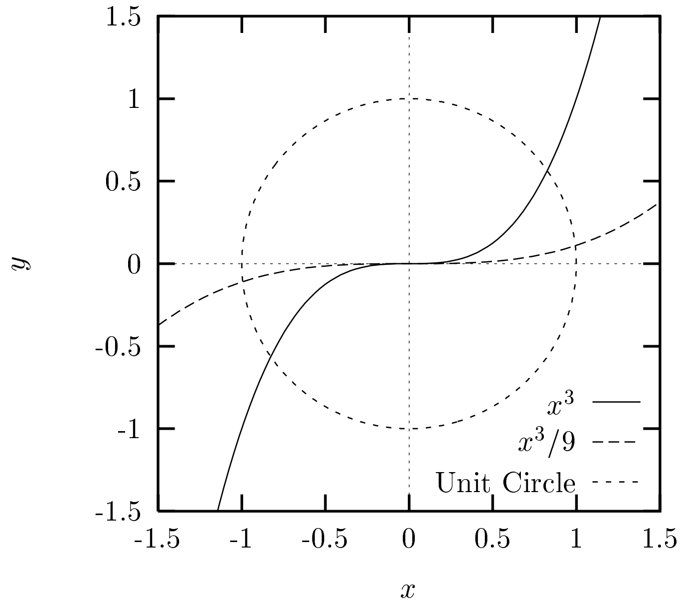

Figure 3.8: Locus of States in which F_u is Rank Deficient

-0.965,$s = 0.191$ and $s = -0.809$. Figure 3.5 shows a discontinuity for $s = 0.035$ and -0.965, but none at the other two values. Although somewhat unexpected, this illustrates that the rank condition for F_u is *necessary* but not *sufficient* for discontinuity of the model predictive control objective. This is fortunate in the Hermes case, since F_u loses rank at the origin, where the objective is continuous. However, the rank condition is sufficient for continuity; if F_u were to have full rank at the origin, the model predictive control objective function would be continuous there.

The stability of the closed-loop system depends on the continuity of the objective function at the origin. It may also be of interest to predict where the objective function may be discontinuous away from the origin. If we replace Equation 3.19 with $x^2 + y^2 = r^2$, we can solve the system of equations numerically to obtain a locus of points for $r \in [0, \infty)$. This is shown in Figure 3.8. As it happens, the curves shown correspond to the equations $y = x^3$ and $y = x^3/9$. (The unit circle is included to provide a point of reference for comparison to other results in this section.) Along the curve for x^3, no discontinuity in the MPC objective is observed, reinforcing the distinction between the sufficiency and the necessity of the rank condition on F_u.

Theorem 13 also addresses continuity of solutions of nonlinear programs and is applicable to the question of continuity of the feedback control law generated using model predictive control. The discontinuity exhibited by the

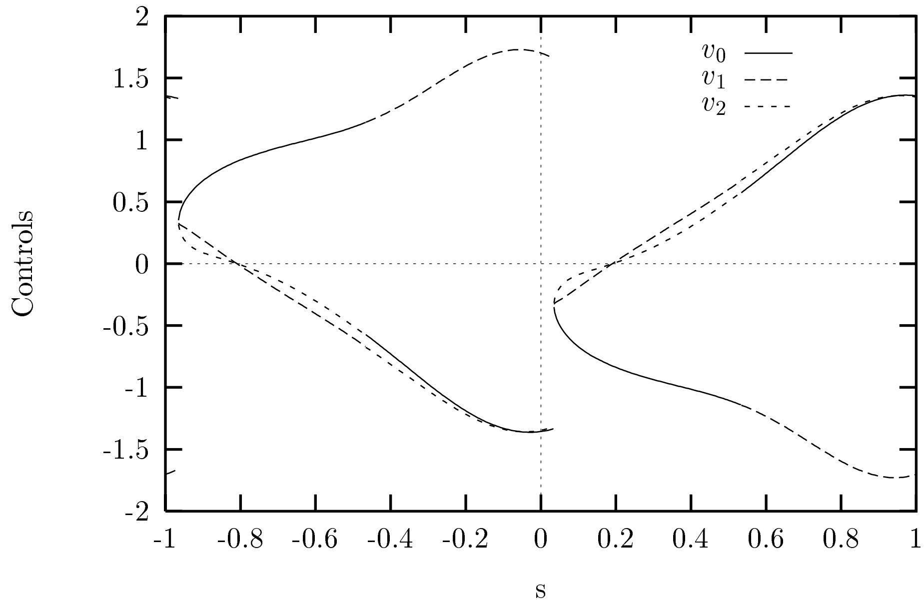

Figure 3.9: First Control Input u_0 For Various N

control law shown in Figure 3.2 for $s = -0.47$ and 0.53 appears to violate Berge's theorem. The u_0 displayed is clearly discontinuous there but the Jacobian matrix F_u has full rank. The upper semi-continuity result of Berge would seem to imply a continuous relationship between the initial state and u_0 at this point.

The solution to this apparent paradox lies in recognizing that Berge's notion of semi-continuity applies to point-to-set mappings, not single-valued functions. The solution of the model predictive control program consists of *three* controls, not one; therefore, the applicable point-to-set map is one from the state space $\Re^2$ to the control space $\Re^3$. Figure 3.9 shows all three open-loop optimal controls as a function of s. Although the controls are discontinuous in the state when considered individually, the set of three is continuous except for $s = 0.035$ or -0.965, where the Jacobian is rank deficient and Berge's theorem does not apply. The above discussion leads to the following formal statement concerning continuity of the model predictive control objective function:

Theorem 14 *For the model predictive control problem of Section 1.2, the optimal objective function $\phi_N^*(x_k)$ is continuous at x_k if the Jacobian matrix F_u, defined through Equation 3.11, has full rank at the optimum with controls $\left\{v_0^*, v_1^*, \ldots v_{N-1}^*\right\}$ and corresponding states.*

Chapter 4

Model Predictive Control for Linear Processes

4.1 Rawlings-Muske Control Formulation

The analysis of Bitmead and coworkers discussed in Section 2.4.1 was concerned with unconstrained linear systems. This section describes the work of Rawlings and Muske [53, 54, 57] for linear systems with constraints on the input and state variables. They consider time-invariant linear systems

$$x_{k+1} = Ax_k + Bu_k \tag{4.1}$$

subject to state and input constraints of the form

$$x_k \in \mathcal{X} = \{x \in \Re^n \,|Hx \le h\} \tag{4.2}$$

$$u_k \in \mathcal{U} = \{u \in \Re^m \,|Du \le d\} \tag{4.3}$$

in which $A \in \Re^{n \times n}$, $B \in \Re^{n \times m}$, $H \in \Re^{p \times n}$ and $D \in \Re^{q \times m}$. The variables x_k, u_k, h and d are vectors of corresponding dimensions. The inequalities are taken element-wise. To ensure that the origin $(x, u) = (0, 0)$ is included in the constraint sets, we can assume without loss of generality that elements of h and d are strictly positive.

Motivated in part by the stability of the infinite horizon LQG optimal controller, Rawlings and Muske form an optimization problem on an infinite horizon:

$$\min_{\{v_i\}} \sum_{i=0}^{\infty} \left(z_i^T Q z_i + v_i^T R v_i \right) \tag{4.4}$$

in which z_i is the solution to Equation 4.1 with control sequence $\{v_0, v_1, \ldots\}$. The optimization is subject to the linear constraints of Equations 4.2 and 4.3.

In theory, this optimization problem could be posed as the limit of a series of finite horizon problems, as in the unconstrained LQG problem, through the use of dynamic programming. The presence of constraints prevents the derivation of a state-independent matrix Riccati equation and corresponding control law for the general problem.

Since no closed form feedback control is known, it is necessary to resort to numerical solutions to the optimization problem of Equation 4.4. This requires a finite parameterization of the control input. Rawlings and Muske chose to parameterize the input as $\{v_0, v_1, \ldots v_{N-1}, 0, 0, \ldots\}$, taking zero control action after the N-th stage. With this parameterization, the MPC objective function can be divided into a finite control horizon and an infinite prediction horizon as follows:

$$\sum_{i=0}^{N-1} \left(z_i^T Q z_i + v_i^T R v_i \right) + \sum_{i=N}^{\infty} z_i^T Q z_i \tag{4.5}$$

If A is stable (all eigenvalues have magnitude strictly less than 1), the second summation in Equation 4.5 can be evaluated as the solution of a matrix Lyapunov equation. Recognizing that

$$\sum_{i=N}^{\infty} z_i^T Q z_i = z_N^T \left(\sum_{i=0}^{\infty} \left(A^i\right)^T Q A^i \right) z_N$$

If we define S by

$$S = \sum_{i=0}^{\infty} \left(A^i\right)^T Q A^i$$

then S satisfies

$$S - A^T S A = Q \tag{4.6}$$

and the value of the infinite horizon objective function of Equation 4.4 is given by

$$\sum_{i=0}^{N-1} \left(z_i^T Q z_i + v_i^T R v_i \right) + z_N^T S z_N \tag{4.7}$$

An immediate question that arises from this formulation is how to evaluate the objective for unstable A. The approach of Rawlings and Muske is to add a state equality constraint to the mathematical program that forces the state z_N to have no component in the unstable subspace of A. This *final state stability constraint* is derived through analysis of the stable and unstable subspaces of A via the Schur canonical form of A, as described below.

4.1.1 Final State Stability Constraint

Rawlings and Muske originally formulated the final state stability constraint [57] in terms of the Jordan canonical form of A. This is appropriate for theoretical purposes and is perhaps the most direct way to identify the unstable subspace of A, but it is known to be numerically unstable. For more general cases, the Real Schur decomposition provides a numerically stable method.

Every matrix $A \in \Re^{n\times n}$ has a Real Schur decomposition [21], expressed as

$$U^T A U = T \tag{4.8}$$

in which U is orthogonal and T is upper block triangular. The diagonal blocks of T are either 1×1 or 2×2 corresponding to simple or complex eigenvalues respectively and may be arbitrarily ordered. The columns of U provide a basis for the stable and unstable subspaces of A. If the diagonal blocks of T are ordered with the smallest magnitude eigenvalues in the upper left block, then U can be partitioned as

$$U = \begin{bmatrix} U_s & U_u \end{bmatrix} \tag{4.9}$$

and the columns of U_s and U_u are bases for the stable and unstable subspaces of A, respectively. (If A has no unstable modes, then U_u is empty and $U = U_s$ and conversely for stable modes.) With U_u so identified, the final state stability constraint is expressed as

$$U_u^T z_N = 0 \tag{4.10}$$

and the final state penalty matrix S of Equation 4.6 is given by

$$S - A_s^T S A_s = Q \tag{4.11}$$

in which A_s satisfies $A_s U_u = 0$ and $A_s U_s = A\,U_s$. The matrix A_s is derived through the partition of T corresponding to the stable and unstable eigenvalues. If the stable eigenvalues are contained in T_{11} and the unstable in T_{22} then A is given by

$$\begin{aligned} A &= UTU^T \qquad (4.12)\\ &= \begin{bmatrix} U_s & U_u \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} U_s^T \\ U_u^T \end{bmatrix} \qquad (4.13) \end{aligned}$$

With these partitions, A_s is given by

$$A_s = U_s T_{11} U_s^T \tag{4.14}$$

4.1.2 State Constraints on Infinite Horizon

A second complication introduced by the finite parameterization of the control input is that, although the objective function is defined on a finite horizon, the state constraints must be satisfied for all k. With zero control input, the set of constraints for $k \geq N$ takes the form

$$\begin{array}{rcll} Hz_N & \leq & h & (4.15) \\ HA_s z_N & \leq & h & (4.16) \\ HA_s^2 z_N & \leq & h & (4.17) \\ \vdots & \vdots & \vdots & \\ HA_x^k z_N & \leq & h & (4.18) \\ \vdots & \vdots & \vdots & \end{array}$$

Despite having a finite parameterization for the control input, the mathematical programming problem is not well posed with an infinite number of constraints. Rawlings and Muske partially addressed this concern by showing that for any $||z_N|| < \infty$, all active constraints must be contained in a finite number of stages. For A matrices that have distinct eigenvalues, an upper bound for the index of the last active constraint was provided by

$$k_2 = \max\{\ln\left(\frac{h_{\min}}{\kappa(V)||H||\,||z_N||}\right) / \ln(\lambda_{\max}), 0\} \tag{4.19}$$

in which $\lambda_{\max}$ represents the largest eigenvalue of A and $\kappa(V)$ is the condition number of V obtained from the Jordan canonical form $A = V^{-1}\Lambda V$. (For comparison with Rawlings and Muske's original definition of k_2, my definition differs by N from theirs. This difference emphasizes the idea that k_2 stages must be checked, rather than tying the end of the checks to a specific index.) The bound k_2 is given in terms of z_N. This makes it impossible to specify *a priori* the size of the mathematical program for the controller. To implement such a bound, a k_2 must be chosen, the optimization problem must be solved and k_2 checked to see if it satisfies Equation 4.19. If it fails, the optimization problem must be solved again with a larger value of k_2 until Equation 4.19 is satisfied. This state dependent approach is undesirable since it requires an iterative procedure whose convergence properties are not known, and requires a changing number of state constraints depending on the magnitude of the initial state.

Gilbert and Tan [19] have provided an algorithm to specify *a priori* the largest index in Equation 4.18 that can have a binding state constraint. Their method has the advantage that is does not depend on a particular z_N. The problem that they posed is to find the set

$$O_\infty(A,H,h) = \left\{ x \in \Re^n \,\middle|\, HA^k x \le h \;\forall\; k = 0,1,2\ldots \right\} \tag{4.20}$$

The problem is solved by defining

$$O_t(A,H,h) = \left\{ x \in \Re^n \,\middle|\, HA^k x \le h \;\forall\; k = 0,1,2\ldots t \right\} \tag{4.21}$$

Gilbert and Tan showed that if $O_t = O_{t+1}$ for some finite t, then $O_\infty = O_t$. Based on this result, they propose the following numerical procedure for determining O_∞:

1. Take $j = 0$.
2. For $i = 1$ to p (the row dimension of H), solve the following linear programming problem. $\{\cdot\}_i$ to be the ith row of the matrix argument:

$$J_i = \max_z \quad \left\{ HA^{j+1} \right\}_i z \tag{4.22}$$

$$\text{Subject to:} \quad HA^k z \le h \quad \forall\; k = 0,\ldots j \tag{4.23}$$

3. If for any J_i, if $J_i - h_i \ge 0$ or is unbounded, take $j = j+1$ and repeat previous step.
4. If each optimal J_i satisfies $J_i - h_i < 0$, then $O_\infty = O_j$ and $k_2 = j$.

The above algorithm requires the solution of a series of linear programming problems for which efficient algorithms and software implementations are available.

With current state x_k, the quadratic program for the MPC controller becomes

$$\min_{z_i, v_i} \quad \sum_{i=0}^{N-1} \left(z_i^T Q z_i + v_i^T R v_i \right) + z_N^T S z_N \tag{4.24}$$

$$\begin{array}{llcccl}
\text{Subject to:} & \text{(A)} & z_0 & = & x_k & \\
 & \text{(B)} & z_{i+1} & = & Az_i + Bv_i & \multirow{3}{*}{$\left.\begin{array}{l}\\ \\ \\ \end{array}\right\} \; i = 0,1,\ldots N-1$} \\
 & \text{(C)} & Dv_i & \le & d & \\
 & \text{(D)} & Hz_i & \le & h & \\
 & \text{(E)} & HA_s^j z_N & \le & h & \quad j = 0,1,\ldots k_2 \\
 & \text{(F)} & U_u^T z_N & = & 0 &
\end{array}$$

For the above quadratic program, the following summarizes the purpose of each constraint: (A) represents the state at time k for which a feedback controller is desired; (B) represents the linear system equations; (C) and (D) represent the specified input and state variable constraints; (E) represents the state constraints over the infinite horizon portion of the objective function; (F) is the final state stability constraint (not present if A is stable). Taking into account all z_i and v_i over the control horizon N, the quadratic program contains $(n+m+1)\times N$ decision variables.

This quadratic program is essentially the same as that presented by Rawlings and Muske [57] in which nominal stability was shown to be provided by the above control formulation. The key extensions in this work are

- The characterization of the unstable subspace of A through the Schur decomposition, and
- The application of the results of Gilbert and Tan [19] to convert the infinite series of state constraints to a finite one that is independent of the state.

The nominal stability result of [57] is not changed by these extensions, which are directed toward establishing the Rawlings-Muske controller as a tool for applications.

4.1.3 Infeasibilities in Quadratic Program

There is nothing in the above analysis to guarantee that the quadratic program of Equation 4.24 will contain a feasible point. The prevailing view of infeasibilities in the literature for model predictive control schemes seems to be negative. However, the view taken by Rawlings and Muske [57], continued in this dissertation, is that infeasibilities reveal conflicting demands that cannot be satisfied by any stable control scheme and that require a reformulation of control objectives. To determine how to reform them, it is necessary to examine the problem constraints more closely.

- State Equations and Initial Condition (A) and (B) : These constraints are based on measurements and on the physical characteristics of the system. Short of repeating or improving a model identification process or obtaining better process measurements, it is not reasonable to change these to obtain a feasible point in the quadratic program.
- Control Variable Inequality Constraints (C) : These constraints are normally associated with hard limits on control actuators such as valve stem

positions. For input variables that are not physically limited in such a way, it is conceivable that constraints (C) could be modified to provide a feasible point for the quadratic program. As the basis for analysis of the algorithm, I adopt the view that these constraints are not subject to change. The implication of this position is that hard constraints on the control variables should be carefully considered before inclusion in the quadratic program. If less control action is desired simply as a performance objective, this can be satisfied through increasing the weighting matrix R in the objective function.

- State Variable Inequality Constraints (D) and (E) : As posed in the quadratic program of Equation 4.24, the state variable constraints include the initial condition x_k. It is possible that the initial condition would not satisfy constraints (D), in which case we would seek a controller to bring the state to within the constraint as quickly as possible. To accomplish that we would simply remove the constraint (D) for $i = 0$ and pose a modified quadratic program that included constraint (D) for $i \geq 1$ only.

  It is still possible that relaxing the first state constraint would not provide a feasible point. The control objective then shifts to one in which the state satisfies the constraints as soon as possible without violating the input variable constraints. This approach was considered by Rawlings and Muske and was shown to have no effect on the nominal stability. They incorporated this into their algorithm through the identification of a time index k_1 which denoted the first index at which the state variable constraints (D) could be satisfied. Using k_1, constraints (D) become

  $$\text{(C)} \quad Hz_i \quad \leq \quad h \quad i = k_1, k_1 + 1, \ldots, N - 1 \tag{4.25}$$

  This adjustment to the algorithm requires the solution of a mixed-integer program to determine the minimum k_1. I will discuss numerical determination of k_1 in a subsequent section. I will also discuss the limiting case for large k_1, in which it is allowed to exceed N, and will show that the nominal stability of the basic algorithm is retained.

- State Variable Stability Constraint (F) : These constraints are absolutely required for nominal stability of the closed loop controller. It may occur that the combination of fixed control variable constraints (C) and the stability constraint (F) act to create an empty feasible set. Rawlings and Muske describe this situation as lacking *constrained stabilizability*. Increasing N may provide a feasible point for the quadratic program. In some cases, there may be *no* N that can provide a feasible point. This issues is discussed below.

4.1.4 Constrained Stabilizability and Choice of Horizon Length

The correct choice of N, the control horizon, is critical for stability of the Rawlings-Muske controller. The concept of controllability to the origin is of central importance in determining N:

Definition 3 *A dynamic system*

$$x_{k+1} = f\left(x_k, u_k\right)$$

is controllable to the origin if for any x_0 there exists a finite control sequence $\{u_0, u_1, \ldots u_{N-1}\}$ satisfying $u_k \in \mathcal{U}$ such that $z_N = 0$. The smallest such N is called the controllability index N_c.

For unconstrained, time-invariant linear systems, controllability to the origin is equivalent to controllability and can be checked by considering the rank of the controllability matrix $\mathcal{C}$, defined by

$$\mathcal{C} = \left[\begin{array}{cccc} B & AB & \cdots & A^{n-1}B \end{array}\right] \tag{4.26}$$

The controllability index N_c for these systems satisfies $N_c \leq n$ in which n is the dimension of A.

Stability of the Rawlings-Muske controller required only that the unstable modes of A be forced to the origin or, equivalently, that the state z_N have no component in the unstable subspace. To investigate the behavior of the unstable modes, it is useful to derive a dynamic system equation for the unstable modes only, beginning with the Schur decomposition for A found in Equation 4.13:

$$A = \left[\begin{array}{cc} U_s & U_u \end{array}\right] \left[\begin{array}{cc} T_{11} & T_{12} \\ 0 & T_{22} \end{array}\right] \left[\begin{array}{c} U_s^T \\ U_u^T \end{array}\right]$$

Substituting this into the system equation of Equation 4.1 and multiplying on the left by U^T provides the following:

$$\left[\begin{array}{c} U_s^T \\ U_u^T \end{array}\right] x_{k+1} = \left[\begin{array}{cc} T_{11} & T_{12} \\ 0 & T_{22} \end{array}\right] \left[\begin{array}{c} U_s^T \\ U_u^T \end{array}\right] x_k + \left[\begin{array}{c} U_s^T \\ U_u^T \end{array}\right] Bu_k \tag{4.27}$$

The unstable modes are obtained from the bottom half of the partitioned system above and we have the following:

$$U_u^T x_{k+1} = T_{22} U_u^T x_k + U_u^T B u_k \tag{4.28}$$

Since stabilizability for linear systems is equivalent to controllability of the unstable modes, we immediately identify a *stabilizability matrix* defined by

$$\mathcal{S} = \left[\begin{array}{cccc} U_u^T B & T_{22} U_u^T B & \cdots & T_{22}^{n_u} U_u^T B \end{array} \right] \tag{4.29}$$

in which n_u represents the dimension of S_{22}, the number of unstable modes $n_u \leq n$. For unconstrained linear systems, stabilizability is equivalent to $\mathcal{S}$ having full rank.

For constrained u_k, the rank condition on $\mathcal{S}$ is necessary for controllability of the unstable modes to the origin, but not sufficient, since in the constrained case, the input sequence $\{u_0, u_1, \ldots u_{N-1}\}$ must satisfy $Du_k \leq d$. If such an input sequence exists, then to achieve stability in the Rawlings-Muske controller, the control horizon must be at least N_{cs}, the *constrained stabilizability index*. N_{cs} is the controllability index for the system of Equation 4.28 subject to the input constraints.

In the presence of input constraints, stabilizability has the additional complication of being dependent on the initial state. This has the unfortunate consequence that some states may be stabilizable and other may not. For nominal stability, this represents a one-time problem: if the initial state is in the stabilizable set, then all subsequent states will be as well and the closed-loop system is stable. In the presence of disturbances, previously stable trajectories may leave the set of stabilizable states and cause the quadratic program to have no feasible point. The implications of this for control of unstable processes will be discussed later.

I am aware of no closed form or *a priori* method to verify controllability or stabilizability in the presence of input constraints. A numerical procedure that yields the stabilizability index N_{cs} is provided below. The concept behind this algorithm is to find the distance between the initial condition and the set of stabilizable states for a fixed test value for N_{cs}. If the distance is non-zero, then the initial condition is not stabilizable and the test value for N_{cs} is increased. The algorithm terminates if the distance becomes zero, indicating stabilizability with index N_{cs}, or if the test N_{cs} exceeds a predetermined maximum value.

If we take $\tilde{B} = U_u^T B$, a quadratic program for checking constrained stabilizability and finding the constrained stabilizability index takes the following form:

$$\min_{v_i} \left|\left| U_u^T x_k - z_0 \right|\right|^2 \tag{4.30}$$

$$\text{Subject to:} \quad \left.\begin{array}{rcl} z_{i+1} & = & T_{22} z_i + \tilde{B} v_i \\ D v_i & \leq & d \end{array}\right\} \ \forall\, i = 0, 1, \ldots N_{cs} - 1$$

$$z_{N_{cs}} = 0$$

Taking J^* as the optimal objective in the quadratic, the following algorithm produces the controllability index N_{cs} if it exists for $N_{cs} \leq N_{\max}$.

1. Choose $N_{\max}$.
2. Set $N = 1$.
3. Solve the above quadratic program.
4. If $J^* > 0$ and $N < N_{\max}$, then set $N = N + 1$ and repeat Step 3.
5. If $J^* < 0$ then set $N_{cs} = N$ and stop.
6. If $N \geq N_{\max}$, stop. The system is not constrained stabilizable for $N_{cs} \leq N_{\max}$.

4.2 State Constraint Violations and Stability

As mentioned in Section 4.1.3, infeasibilities in the Rawlings-Muske controller that are due to the state constraints may be relaxed without compromising stability. The strategy calls for the elimination of state constraints for k_1 initial stages in the prediction horizon until a feasible point can be obtained. Such a k_1 exists [53, 54, 57] if the system is constrained stabilizable and $N \geq N_{cs}$. For the nominal system, after the controller takes k_1 steps, all subsequent states satisfy the state constraints.

Looking ahead to Section 4.4, we are interested in a coupled observer/controller for the Rawlings-Muske controller. Of critical importance for analysis of stability for the controller/observer pair are the continuity properties of the feedback control law $u_k = v_0^*(x_k)$. Since the feedback law is given as the solution to a quadratic program, we turn to a result of Hager [22] that addresses the continuity of solutions of quadratic programs. (Notation and terminology are from Hager and may not correspond to other usage in this dissertation.)

Theorem 15 (Hager) *Given the quadratic program*

$$\min_{v \in \Re^n} \frac{1}{2} v^T R v + r^T v \tag{4.31}$$

$$\begin{aligned} \textit{Subject to:} \quad Av + a &\leq 0 \\ Bv + b &= 0 \end{aligned}$$

and defining the data set $d = (R, r, A, a, B, b)$, and data subsets $\delta(d) = (r, a, b)$ and $\Delta(d) = (R, A, B)$. Let $\mathcal{D}$ be any convex set of data satisfying the following for all $d \in \mathcal{D}$:

1. *There exists a unique solution $u(d)$ to the quadratic program.*
2. *There exist $\Gamma_1, \Gamma_2 < \infty$ such that $||R|| < \Gamma_1$ and $\left|\left|M(d)^T\right|\right| < \Gamma_2$, in which $M(d)$ is the matrix of binding constraints from A and B.*
3. *There exists α such that*

$$v^T R v \geq \alpha ||v||^2 \tag{4.32}$$

for all v satisfying $M(d)v = 0$.

4. *There exists β such that*

$$\left|\left|M(d)^T \lambda\right|\right| \geq \beta ||\lambda|| \tag{4.33}$$

for all λ.

Then there exists a constant $\rho < \infty$ such that for all $d_1, d_2 \in \mathcal{D}$, we have

$$||u(d_1) - u(d_2)|| \leq \rho||\delta(d_1) - \delta(d_1)|| + \rho^2 ||\Delta(d_1) - \Delta(d_2)|| \left(||\delta(d_1)|| + ||\delta(d_2)||\right) \tag{4.34}$$

Moreover, the constant ρ is given by

$$\rho \leq \alpha^{-1} + 2\kappa_1\beta^{-1} + 4\kappa_1\kappa_2\kappa_3 \tag{4.35}$$

in which

$$\kappa_1 = \Gamma_1/\alpha \tag{4.36}$$

$$\kappa_2 = \Gamma_2/\alpha \tag{4.37}$$

$$\kappa_3 = \max\{\Gamma_1/\beta, 1\} \tag{4.38}$$

Hager's theorem provides the foundation for analyzing the continuity of the Rawlings-Muske control formulation. Consider first only the case where $k_1 = 1$ (state constraints apply for all k). Table 4.1 provides an illustration of the constraints that apply in this case. The decision variables $\{z_1, z_2, \ldots z_N\}$ and $\{v_0, v_1, \ldots v_{N-1}\}$ are listed at the top of the table. Matrix equalities and inequalities can be obtained by left-multiplying the matrices below by the corresponding variable from the top of the table. The right-hand sides of the constraints are provided at the far right. In Table 4.1, the first grouping of constraints represent the N state equations and the second grouping represents the N state inequality constraints. The third group is the final state stability constraint and the fourth represents the state constraints on the infinite horizon, reduced to a finite number through the algorithm of Section 4.1.2. The final block of constraints represents the input variable inequality constraints.

z_1	z_2	z_3	$\cdots$	z_{N-2}	z_{N-1}	z_N	v_0	v_1	v_2	$\cdots$	v_{N-3}	v_{N-2}	v_{N-1}		RHS
I							$-B$								Ax_i
$-A$	I							$-B$							0
	$-A$	I							$-B$						
		$-A$	$\ddots$							$\ddots$				$=$	$\vdots$
			$\ddots$												
				I							$-B$				
				$-A$	I							$-B$			
					$-A$	I							$-B$		0
H															h
	H														
		H													
			$\ddots$											$\leq$	$\vdots$
				H											
					H										
						H									h
						U_u^T								$=$	0
						HA_s									h
						$\vdots$								$\leq$	$\vdots$
						$HA_s^{k_2-1}$									
						$HA_s^{k_2}$									h
							D								d
								D							
									D						
										$\ddots$				$\leq$	$\vdots$
											D				
												D			
													D		d

Table 4.1: Tableau of Constraints for $k_1 = 1$

By the visual presentation of Table 4.1, we will more easily be able to verify the sufficient conditions of Hager that are discussed below on a point-by-point basis:

- Convexity of data set $\mathcal{D}$: For fixed k_1, the only parameter that varies in the data set is the initial state x_k which corresponds to variations in

Hager's b through Ax_k. For fixed N and k_1, x_k lies within the feasible region defined by the remainder of the constraints of Table 4.1. Since h and d are strictly positive, this set of linear equality constraints defines a convex region in $\Re^n$ and so satisfies Hager's convexity condition.

- Uniqueness of Solution: Provided that the constraints are linearly independent at the optimum (this point is discussed below), it is unique if the weighting matrix R in Hager's formulation is positive definite [16]. For the Rawlings-Muske controller, this condition is satisfied by positive definite state and control weighting matrices Q and R, respectively.

- Existence of Γ_1 and Γ_2: Hager's R corresponds to Kronecker products of Q, R and S with the identity matrix, which do not vary with x_k or k_1; therefore, we may take $\Gamma_1 = \max\{||Q||, ||R||, ||S||\}$.

  An upper bound for $\left|\left|M(d)^T\right|\right|$, the norm of the binding constraints, is needed to get Γ_2. This bound can be easily obtained by recognizing that the norm of any matrix is equal to its largest singular value. From the interlacing property of the singular values of sub-matrices [21, 25], and the fact that, at most, the number of active constraints is equal to the number of decision variables, it follows that Γ_2 is equal to the largest singular value of the M consisting of the equality constraint matrices augmented by the subset of the inequality constraints which yields a square matrices having maximum norm.

- Lower Bound of $v^T Rv$: The condition of Equation 4.32 is trivially satisfied for positive definite state and input weighting matrices by taking $\alpha = \min\{\sigma_{\min}(Q), \sigma_{\min}(R), \sigma_{\min}(S)\}$ in which $\sigma_{\min}$ represents the smallest singular value of the matrix argument which will be non-zero. This bound holds regardless of the subsidiary condition $M(d)v = 0$.

- Lower Bound of $M(d)^T\lambda$: The sufficient condition of Equation 4.33 is a rank condition on the matrix of active constraints at the solution. This condition may not be satisfied for arbitrary constraints; however, the constraints may be structured in a way to ensure that Equation 4.33 is satisfied without any loss of generality, through the Constraint Reduction Algorithm presented in Appendix B. The set of possible constraints remaining after applying the Reduction Algorithm is necessarily an independent set. The lower bounding constant β is then available as the minimum singular value of the equality constraints augmented by the subset of the inequality constraints that provide the minimum singular value.

With Γ_1, Γ_2, α and β so defined, the sufficient conditions for Lipschitz continuity of the Rawlings-Muske controller for fixed k_1 are satisfied. We can determine a Lipschitz constant via Equation 4.35. Since the only variable under consideration is the initial state x_k, Equation 4.34 becomes

$$||u(d_1) - u(d_2)|| \leq \rho ||x_k - \tilde{x}_k|| \tag{4.39}$$

4.3 Continuity of Controller with Increasing k_1

The complete formulation of the Rawlings-Muske controller included the possibility of removing the state constraints up to time k_1 to allow a wider range of initial conditions to be stabilized. If we allow the initial state to range over all of $\Re^n$, there is no upper bound on k_1. We therefore need to analyze the continuity of the control law as k_1 varies.

Consider the sets

$$\mathcal{X}_j = \left\{ z_0 \in \Re^n \,\middle|\, \begin{array}{l} \text{A feasible point exists for the quadratic program of} \\ \text{the Rawlings-Muske controller with } k_1 = j \end{array} \right\} \tag{4.40}$$

These sets have the following properties:

- They are nested: $\mathcal{X}_j \subseteq \mathcal{X}_{j+1}$
- Since they are defined by non-strict inequalities or equality constraints, they are closed.
- They may or may not be bounded.
- They each contain a neighborhood of the origin.

As originally proposed, changes in k_1 would be induced by the absence of a feasible point for a constrained stabilizable system. For purpose of this analysis, it is more convenient to consider the sets $\mathcal{X}_j$ as having independent existence with the state of the controlled system (possibly) traversing the sets as time progresses.

Two issues are relevant to the continuity of the control law:

- Whether a Lipschitz bound exists in the limit as $k_1 \to \infty$; and
- Whether the control law in continuous at the boundaries of the $\mathcal{X}_j$ regions.

z_1	z_2	z_3	$\cdots$	z_{N-2}	z_{N-1}	z_N	y	v_0	v_1	v_2	$\cdots$	v_{N-3}	v_{N-2}	v_{N-1}		RHS
I								$-B$								Ax_i
$-A$	I								$-B$							0
	$-A$	I								$-B$						
		$-A$	$\ddots$												$=$	$\vdots$
			$\ddots$								$\ddots$					
				I								$-B$				
				$-A$	I								$-B$			
					$-A$	I								$-B$		0
								D								d
									D							
										D						
											$\ddots$				$\leq$	$\vdots$
												D				
													D			
														D		d
						U_u^T										0
							H									h
							HA_s									
							$\vdots$								$\leq$	$\vdots$
							$HA_s^{k_2-1}$									
							$HA_s^{k_2}$									h
						$A_s^{k_1-N}$	$-I$								$=$	0

Table 4.2: Tableau of Constraints for $k_1 \geq N$

4.3.1 Lipschitz Bound for Limiting $k_1 \to \infty$

For large k_1, the state constraints will no longer be enforced within the control horizon N. It is convenient to introduce a new variable into the quadratic program using $y = A_s^{k_1-N} z_N$. (Recall that the controller formulation only allows for the action of the stable modes of A on the state, for $i > N$.) The tableau of constraints takes a slightly different form as indicated in Table 4.2. The introduction of y causes the variable k_1 to enter the constraint tableau as a single matrix equality constraint, indicated as the last row in Table 4.2. For finite k_1, it is clear by inspection that the introduction of y will not lead to linear dependence in the active constraint set, since no other rows or combinations of rows multiply the same variables; however, as $k \to \infty$, we have $A_s^{k_1-N} \to 0$ and it seems possible that the remaining identity matrix might be linearly dependent on some combination of the matrices immediately above it in the tableau. This concern can be alleviated by observing that the right-hand side of the state inequality constraints involving y are fixed and greater than zero. Since the final constraint of the table requires $y \to 0$, none of the state inequality constraints can be binding. Furthermore, it is always possible to eliminate linearly dependencies in the binding set through the Constraint Reduction Algorithm of Appendix B.

This shows that a Lipschitz constant exists for any finite k_1 and the matrix of binding constraints converges.

4.3.2 Continuity of Control at Boundary of k_1 Regions

I showed in Section 4.3 that, for fixed k_1, the feedback control law given by the Rawlings-Muske controller is Lipschitz continuous in the state. We now seek to answer the question of whether the control law is continuous in k_1. Since k_1 is a function of th initial state, this is equivalent to continuity along the boundaries of the sets $\mathcal{X}_j$.

For a given control problem, it is possible that k_1 might need to be enlarged initially to allow a wider range of initial states that could be stabilized through the model predictive formulation. Rawlings and Muske showed that for nominal systems, k_1 would decrease by one at each new time step without producing a quadratic program with no feasible points. Without disturbances, k_1 is monotonically decreasing in time.

In the presence of disturbances, it is possible that a state trajectory could move outside the feasible set for a specific k_1 and the value of k_1 might need to be increased to provide a feasible point. Since increases and decreases in k_1 can occur dynamically under this scheme, we must consider whether the controller is continuous in k_1.

Numerical experiments indicate that the controller is not continuous in k_1. Figure 4.1 shows the control law versus the state variable for the SISO system studied by Muske and Rawlings [54] in which

$$A = \begin{bmatrix} 4/3 & -2/3 \\ 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{4.41}$$

There are no control constraints, but the state is constrained by $Hx_k \leq h$ in which

$$H = \begin{bmatrix} -2/3 & 1 \end{bmatrix} \qquad h = 1/2 \tag{4.42}$$

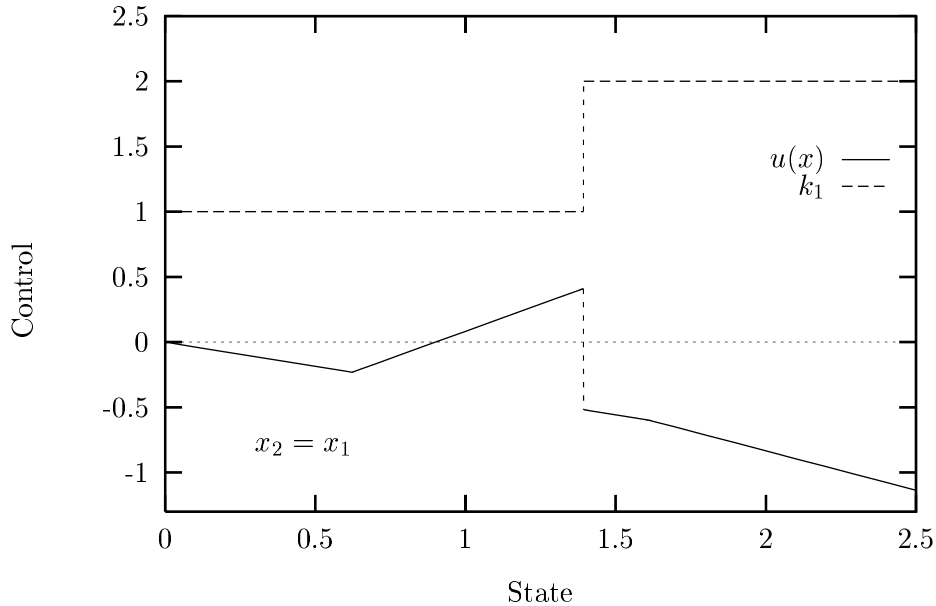

Figure 4.1: Discontinuity in Rawlings-Muske Controller

It appear possible that the discontinuity may have less impact than may be suspected. Numerical experiments continue to indicate asymptotic stability even with disturbance. This remains a topic of current research.

4.4 Stability Results for Perturbed Systems

The following result is central to the discussion of the coupled observer/controller pair (stability definitions are standard):

Theorem 16 *Let $f : \Re^n \to \Re^n$ be Lipschitz continuous with $f(0) = 0$ and let the origin be an asymptotically stable fixed point of $x_{k+1} = f(x_k)$. If e_k is an exponentially stable sequence, then the origin is an asymptotically stable fixed point of the system $\tilde{x}_{k+1} = f(\tilde{x}_k) + e_k$.*

Proof: To demonstrate asymptotic stability of $\tilde{x}_k$, we must show both stability (Lyapunov stability) and convergence. We refer to the results of Halanay [23] for properties of Lyapunov functions for x_k. We begin by noting that Halanay does not include f continuous as a sufficient condition. However, the sufficient conditions of Halanay follow from our continuity assumption.

Convergence: From Halanay, there exist $r > 0$ and a Lyapunov function $V(x)$ for the unperturbed system satisfying

$$a(||x||) \leq V(x) \leq b(||x||)$$
$$\Delta V(x) = V(f(x)) - V(x) \leq -c(||x||)$$

for $||x|| < r$. The functions a, b and c are class-K functions [60], that is, they are continuous and strictly increasing with $a(0) = b(0) = c(0) = 0$. Halanay proves that $V(x)$ is Lipschitz continuous, and so it can be shown that

$$\begin{aligned} \Delta V(\tilde{x}_k) = V(f(\tilde{x}_k) + e_k) - V(\tilde{x}_k) &\leq -c(||\tilde{x}_k||) + K||e_k|| && (4.43) \\ &\leq -c(||\tilde{x}_k||) + K\rho\lambda^k && (4.44) \end{aligned}$$

In which K is the Lipschitz constant. This last inequality arises from the exponential convergence of e_k in which $\rho \geq 0$, $0 < \lambda < 1$ and $||e_k|| \leq \rho\lambda^k$. Using $\tilde{x}_{k+1} = f(\tilde{x}_k) + e_k$ and summing the above inequality N times gives

$$V(\tilde{x}_{k+N}) - V(\tilde{x}_k) \leq -\sum_{j=k}^{k+N} c(||\tilde{x}_j||) + K\rho\frac{\lambda^k}{1-\lambda}$$

or equivalently

$$\begin{aligned} \sum_{j=k}^{k+N} c(||\tilde{x}_j||) &\leq V(\tilde{x}_k) - V(\tilde{x}_{k+N}) + K\rho\frac{\lambda^k}{1-\lambda} && (4.45) \\ &\leq V(\tilde{x}_k) + K\rho\frac{\lambda^k}{1-\lambda} && (4.46) \end{aligned}$$

The right-hand side of the last inequality above is not a function of N; therefore, the sequence of partial sums is bounded above by a constant. This means that $c(||\tilde{x}_k||) \to 0$ as $k \to \infty$. From the properties of c, this implies $||\tilde{x}_k|| \to 0$ and convergence is proven.

Stability: Identifying $R(t_k, x)$ with e_k and $\delta_2 = \rho$ the result follows directly from Section 3 of Halanay [23].

4.5 Stability of Combined Observer/Controller

In this section, we apply the above result to the controller of Rawlings and Muske with unknown initial condition. Taking a linear, time-invariant model of the form

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k && (4.47) \\ y_k &= Cx_k && (4.48) \end{aligned}$$

The pair (A, C) is assumed detectable and the state is estimated with a stable linear observer.

Muske and Rawlings showed that an asymptotically stabilizing state feedback controller that incorporates linear state and input constraints could be constructed as the solution to a quadratic program. The controller is implemented as a model predictive controller in which the quadratic program is posed and solved at each time step. Since the control at time k depends on the state we can write Equation 4.47 as

$$x_{k+1} = Ax_k + Bu^* (x_k) \tag{4.49}$$

in which $u^*(x_k)$ represents the feedback control law obtained using from the Rawlings-Muske MPC formulation. In the absence of constraints, the feedback control law, $u^* (x_k)$, would be a linear function of x_k determined through linear-quadratic-Gaussian optimal control theory. With constraints, it becomes a nonlinear function for which we no longer have a closed-form expression. As previously noted [46], since the MPC control law, $u^* (\cdot)$, is given through the solution of a quadratic program, it is Lipschitz continuous in its argument [22] for fixed k_1.

We are interested in the effect of an observer on the stability of the system of Equation 4.49. We consider here the special case of an unknown initial condition with a stable linear observer. Because the control input is known exactly, the nonlinearities in u^* have no bearing on the stability properties of the observer, and the reconstruction error ϵ_k is exponentially stable. If we take x_k and $\hat{x}_k$ as the true and reconstructed state, respectively, the state of the system evolves in accordance with

$$\begin{aligned} x_{k+1} &= Ax_k + Bu^* (\hat{x}_k) && (4.50)\\ &= Ax_k + Bu^* (x_k + \epsilon_k) && (4.51)\\ &= Ax_k + B \left[u^* (x_k + \epsilon_k) - u^*(x_k) + u^* (x_k)\right] && (4.52)\\ &= Ax_k + Bu^*(x_k) + B \left[u^* (x_k + \epsilon_k) - u^*(x_k)\right] && (4.53) \end{aligned}$$

Identifying $f(x) = Ax + Bu^*(x)$, we see that the system evolves as an asymptotically stable system that is subject to the additive disturbance

$$B \left[u^* (x_k + \epsilon_k) - u^*(x_k)\right]$$

at each time step. Since the reconstruction error ϵ_k is exponentially stable and $u^*(\cdot)$ satisfies a Lipschitz condition, the disturbance satisfies the following inequality and therefore is also exponentially stable.

$$||B \left[u^* (x_k + \epsilon_k) - u^*(x_k)\right] || \leq ||B|| \, K_c ||\epsilon_k|| \leq ||B|| \, K_c \rho \lambda^k$$

From the above inequality, we see that the combined observer/controller satisfies the sufficient conditions of Theorem 16 and the resulting system is asymptotically stable.

Chapter 5

Model Predictive Control of Stochastic Systems

Model Predictive Control uses deterministic models to predict the effect of control actions. Implementation of MPC with *stochastic* models, especially for nonlinear or constrained systems is not clearly defined. Bertsekas [4] provides a discussion of various control strategies that involve differences in how feedback and computation of expected values are used in the controller formulation. Dreyfus and Law [12, 13] cite an example that clearly demonstrates the suboptimality batch optimization schemes that depend on the current state value (i.e. MPC), when compared to optimization through dynamic programming, which provides *the* optimal feedback controller. In Section 5.1, I present a modified version that specifically highlights the differences between dynamic programming and model predictive control approaches to the stochastic control problem. In Section 5.2, I examine a more familiar problem, one with a linear system containing stochastic coefficients.

For the general dynamic system

$$x_{k+1} = f(x_k, u_k, w_k) \tag{5.1}$$

with state x_k, control u_k and stochastic input w_k, we seek a controller to minimize a criterion

$$\phi = \mathop{E}_{w}\left[\sum_{i=0}^{N-1} L\left(z_i, v_i\right) + \phi_0\left(z_N\right) \;\middle|\; x_k \right] \tag{5.2}$$

in which $z_0 = x_k$ and $z_{k+1} = f(z_k, v_k)$. The expectation is conditioned on x_k. This section does not address the issue of nonlinear state estimation, so I assume that full state measurement is available. (See the paper of Muske and Rawlings [55] for a discussion of nonlinear state estimation.) I also assume that distributional parameters are known for the stochastic input w_k so that we can calculate any expectations needed for ϕ.

The minimizing control may be obtained through dynamic programming (DP) which can be shown to provide the minimizing controller for multistage optimization problems of this kind [4]. Except for special cases, dynamic programming does *not* represent a computationally feasible control method because of the "curse of dimensionality" cited by Bellman [1] and others.

Bertsekas [4] discusses three other (suboptimal) control schemes for stochastic multistage problems, none of which share the dimensionality problem that afflicts dynamic programming. Each is optimization-based and requires a measured initial state, but treats process measurements and the stochastic nature of w_k differently. The three methods are as follows:

- *Certainty Equivalence Control* is motivated by the certainty equivalence principle arising from LQG control theory. In this technique, all random variables in the problem are replaced by their expected values. A controller is then obtained through the solution of the resulting deterministic optimal control problem, which can be computed without resorting to dynamic programming.

- *Open Loop Feedback Control* incorporates the effect of future stochastic input variables on the objective function by computing the expectations arising in Equation 5.2. Although the effect of *past* random variables is incorporated at each stage by adjusting the state to conform to the measured value, this method does not consider the effect of future measurements on the optimization criterion. It is this method that is identified with MPC.

- *Open Loop Control without Feedback* does not account for the effect of future state observations and does not update the current state to account for current state measurements. Only one optimal control profile is computed which is not changed subsequently based on new state measurements.

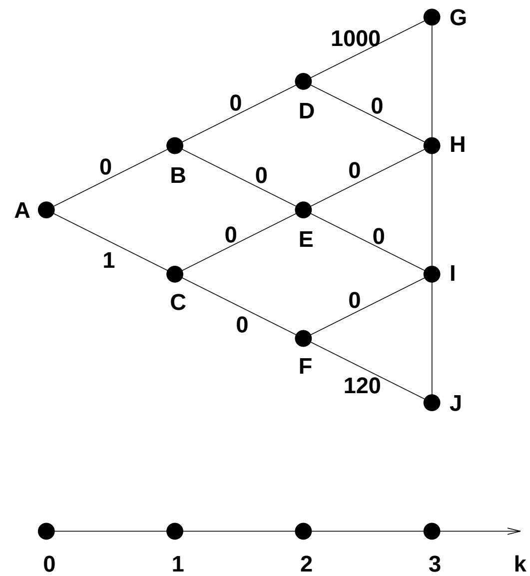

Figure 5.1: Modified Dreyfus Optimal Path Problem

5.1 The Dreyfus Problem

In this section, I present a simple stochastic control problem to illustrate the difference between MPC and optimal feedback control. The example presented here is based on an example from Dreyfus [12]. The data from the original example have been changed in order to illustrate better the points to be discussed.

Consider the control problem illustrated in Figure 5.1. The problem indicated in Figure 5.1 is an optimal path problem in which we seek a path from A to any of the the four right-hand points G, H, I, or J that has minimum cost. The cost of each transition is indicated. The control action at each point is either up (U) or down (D). For the deterministic problem, we have complete assurance that the state trajectory will proceed exactly according to the control input: if U is a control decision, the state moves up to the adjacent vertex and similarly a control action D always causes the state to move down.

For this simple deterministic control problem, there are several control sequences that will result in zero cost, for example {UDU}. Clearly, it makes no difference at all in the total cost if we implement a feedback scheme; once the initial open-loop sequence is chosen, no changes will result from any new state information because the state can be perfectly predicted in advance.

If we allow the possibility that disturbances enter the system, then state

sequence	det. cost	stoch. cost
DDD	121	67.0
DDU	1	64.5
DUD	1	64.6
DUU	1	146.9
UDD	**0**	**64.0**
UDU	0	146.5
UUD	0	146.5
UUU	1000	473.2

Table 5.1: Open-loop cost from state A.

feedback becomes important. Consider the case now in which the effect of control action is random: if U is called for, then there is a 75 percent chance of actually implementing U and 25 percent chance of implementing D and similarly for the case when we specify D. This defines a stochastic control problem in which state feedback will play a role and the difference between dynamic programming and model predictive control becomes apparent.

Model predictive control is implemented for this problem as follows: We can enumerate all possible open-loop controls from point A as indicated in Table 5.1. The deterministic cost is given by assuming that the probability of making each transition in accordance with the control is 1. The stochastic cost is obtained from the expectation of the costs for each specified control trajectory. The expectation uses all the available information in the problem including the transition probabilities, the current state and knowledge of all the transition costs. For example, consider the control sequence {UDU}. Since there are only three non-zero costs, we can directly compute the expected value of this control sequence by the following:

$$\text{Cost}(\mathsf{UDU}) = (0.75)(0.25)(.75)(1000) +$$

$$(0.25)(1) + (0.25)(0.75)(0.25)(120) = 146.5$$

Costs for the remaining control sequences are indicated in Table 5.1, which indicate that the best possible cost using this control strategy is {UDD}.

To implement feedback in this method, we inject the first control of the series {UDD} and observe the state of the system after the transition. Based

state	MPC		DP	
	input	cost	input	cost
A	**U**	**49.0**	**D**	**22.0**
B	D	62.5	D	62.5
C	U	7.5	U	7.5
D	D	250	D	250
E	U*	0	U*	0
F	U	30	U	30

* or D, non-unique solution

Table 5.2: Feedback control action and closed-loop cost for MPC and optimal control.

on this new initial state, we compute another open-loop optimal control input, this time containing two moves, that will result in lowest expected cost. This process is repeated until the final desired state on the right-hand side of Figure 5.1 is reached. The computed control moves at each state A through F are indicated in Table 5.2 along the cost from each point to the end of the problem.

The optimal feedback policy is determined via dynamic programming. Beginning with the points on the right-hand side of Figure 5.1, the optimal policy is computed from each point, and an optimal cost is associated with each point. As the algorithm progresses from the right-hand points to point A, optimal feedback laws are computed that consider the fact that control moves from future states will also be optimal.

Comparing the dynamic programming and MPC results in Table 5.2, we see that both methods predict the same control from each point. The exception is at point A, where the cost of dynamic programming is less than half that of model predictive control. This substantial difference was *not* seen in the examples of Dreyfus and Law [12, 13]. In their examples, the difference was less than one percent between dynamic programming and model predictive control. Based on their result, it might be argued that MPC is alway close to DP in performance. Our example conclusively demonstrates that this is not the case.

5.2 Linear System with Random Coefficients

The Dreyfus and Law example was an optimal path problem. It could be claimed that the example problem chosen [12, 13, 56] is unrepresentative of any real problem of practical interest. In this section, I turn to a somewhat more familiar scalar linear dynamic system. The stochastic character arises in the random coefficient of the input variable, in which the statistics of the random coefficient are known.

We seek to regulate the following stochastic dynamic system to the origin:

$$x_{k+1} = x_k + w_k u_k \tag{5.3}$$

in which x_k, w_k and u_k are scalar real variables with known initial condition x_0. The following description of the independent random input w_k is available:

$$E\left(w_k\right) = 2p - 1 \tag{5.4}$$

$$E\left(w_k w_j\right) = \left\{ \begin{array}{ll} 1 & k = j \\ (2p-1)^2 & k \neq j \end{array} \right] \tag{5.5}$$

in which $p \in [0, 1]$. Although these statistics do not specify a unique distribution, one possibility is a discrete one in which w_k takes values 1 or -1 with probabilities indicated below:

$$P\left(w_k = 1\right) = p \tag{5.6}$$

$$P\left(w_k = -1\right) = 1 - p \tag{5.7}$$

A discrete distribution with the specified statistics is illustrated in Figure 5.2.

Another possible distribution is that of a normally distributed random variable with mean $2p - 1$ and variance $1 - (2p-1)^2$. The corresponding normal distribution is shown in Figure 5.3. In both figures, $p = 0.2$.

The controller is designed to minimize the criterion

$$\underset{w_0, w_1, \ldots w_{N-1}}{E} \left[\sum_{k=0}^{N-1} \left(x_k^2 + u_k^2 \right) + x_N^2 \,\middle|\, x_0 \right] \tag{5.8}$$

As indicated above, the expectation is taken with respect to the random inputs $\{w_0, w_1, \ldots w_{N-1}\}$, conditioned on x_0. (As indicated by the usage above, I use x_k and u_k in place of z_i or v_i. In this case, there is little room for confusion, so I have chosen to retain the x_k-u_k notation in this section.)

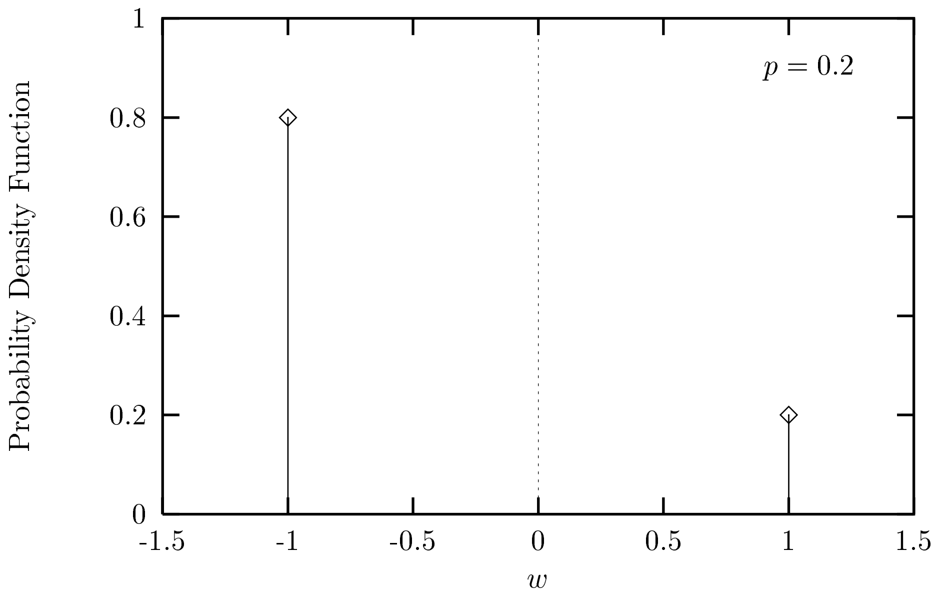

Figure 5.2: Discrete Distribution with Specified Statistics

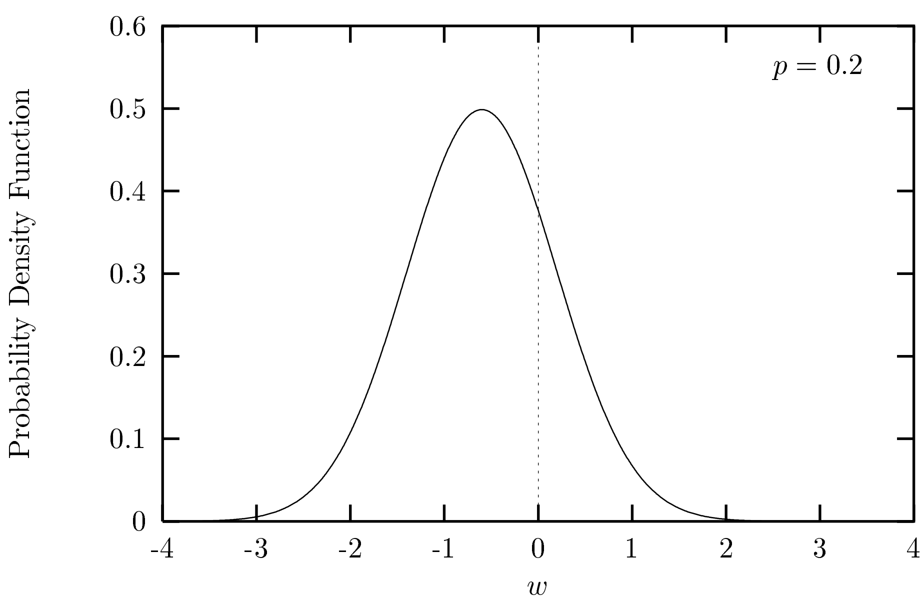

Figure 5.3: Normal Distribution with Specified Statistics

5.2.1 Dynamic Programming Solution

The DP algorithm requires the solution of a recursive series of subproblems, proceeding backward from N to 1:

$$J_N^*(x) = \min_u \left[\underset{w}{E} \left(x^2 + u^2 + J_{N-1}^* \left[f(x,u) \right] \Big| x \right) \right]$$

in which $J_N^*(\cdot)$ is the optimal N-stage objective as a function of the state. Since the dynamic programming objective function is *not* equivalent to the objective function for the MPC, I retain the notation J_N^* to distinguish them. Taking $J_0^*(x) = x^2$, we obtain the following from Equations 5.3 through 5.5 and the linearity of the expectation operator.

$$J_1^*(x) = \min_u \left[x^2 + u^2 + \left(x^2 + 2x(2p-1)u + u^2 \right) \right]$$

Unconstrained minimization of the above expression gives

$$u_1^*(x) = -\frac{\lambda}{2} x \tag{5.9}$$

in which $\lambda = 2p-1$. The notation $u_N^*(x)$ will be used in this section to indicate the feedback law arising from an N-stage dynamic programming problem.

If the control law of Equation 5.9 is substituted into the expression for J_1^*, we obtain

$$J_1^*(x) = \frac{4-\lambda^2}{2} x^2$$

Continuing in the same manner reveals that $J_N^*(x)$ is proportional to x^2. If we let

$$J_N^*(x_{k+1}) = c_N x^2 \tag{5.10}$$

then the DP algorithm can be represented by the following recursive equation:

$$J_N^*(x) = \min_u E \left[x^2 + u^2 + c_{N-1} \left(f(x,u) \right)^2 \Big| x \right]$$

Substituting Equations 5.3, 5.4 and 5.5 into the above and performing the indicated minimization gives the following feedback control law:

$$u_N^*(x) = -\frac{c_{N-1}}{1+c_{N-1}} \lambda x \tag{5.11}$$

To obtain a recursive equation for c_k, we substitute Equation 5.11 into the expression for J_N^* and compare the result to Equation 5.10 to obtain

$$\begin{aligned} J_N^*(x) &= \frac{\left[1 - c_{N-1}(1-\lambda)\right]\left[1 + c_{N-1}(1+\lambda)\right]}{1+c_{N-1}} x^2 && (5.12) \\ &= c_N x^2 && (5.13) \end{aligned}$$

Therefore, the complete dynamic programming solution becomes

$$
\begin{aligned}
c_0 &= 1 && (5.14)\\
c_N &= \frac{\left[1 + c_{k+1}\left(1-\lambda\right)\right]\left[1 + c_{N-1}\left(1+\lambda\right)\right]}{1 + c_{N-1}} && (5.15)\\
u_N^*\left(x\right) &= -\frac{c_{N-1}}{1 + c_{N-1}}\lambda x && (5.16)
\end{aligned}
$$

This derivation produces a recursive solution equivalent to that of Bertsekas [4] in which he discusses linear systems with stochastic systems matrices.

5.2.2 Model Predictive Control Solution

In contrast to dynamic programming, the MPC approach does not solve a series of subproblems. Rather, the problem is posed over the complete horizon and mathematical programming techniques are applied to solve for a complete open loop trajectory over the prediction horizon. Nonlinearities in the process model, non-quadratic penalty functions and constraints introduce complexities that usually require numerical solutions using nonlinear optimization computer codes. However, due to the structure of this example, the solution to the optimization problem is available as the solution to an N-dimensional linear algebra problem.

The one-stage MPC controller has exactly the same solution as the dynamic programming result. The differences first appear for $N \geq 2$. For $N = 2$, the objective is given by

$$
\min_{u_0,u_1}\left[\underset{w_0,w_1}{E}\left(x_0^2 + u_0^2 + x_1^2 + u_1^2 + x_2^2\right)\right]
$$

Using Equation 5.3, performing the necessary expansions and inserting the expectations of Equations 5.4 and 5.5 gives

$$
\phi_2 = 3x_0^2 + 4\left(2p-1\right)x_0u_0 + 3u_0^2 + 2\left(2p-1\right)x_0u_1 + 2\left(2p-1\right)^2 u_0u_1 + u_1^2
$$

in which ϕ_2 represents the two-stage objective function. Since the minimization is unconstrained, the optimal values of u_0 and u_1 can be obtained through differentiating with respect to each, setting the result to zero and solving the resulting system of equations.

$$
\begin{aligned}
\partial\phi/\partial u_0 &= 6u_0 + 2(2p-1)^2u_1 + 4(2p-1)x_0 &= 0\\
\partial\phi/\partial u_1 &= 2(2p-1)^2u_0 + 4u_1 + 2(2p-1)x_0 &= 0
\end{aligned}
$$

Using matrix notation and taking $\lambda = 2p - 1$, these equations become

$$\begin{bmatrix} 3 & \lambda^2 \\ \lambda^2 & 2 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix} = - \begin{bmatrix} 2 \\ 1 \end{bmatrix} \lambda x_0 \tag{5.17}$$

The solution of this system of equation is given by

$$u_0 = - \frac{\lambda(4 - \lambda^2)}{6 - \lambda^4} x_0 \tag{5.18}$$

$$u_1 = - \frac{\lambda(3 - 2\lambda^2)}{6 - \lambda^4} x_0 \tag{5.19}$$

A key distinction between the results of model predictive control and dynamic programming is illustrated in Equations 5.18 and 5.19 in the way that the computed control moves depend on the state. In model predictive control, *all* the control moves depend on the *initial* state x_0. In dynamic programming, each control move depends on the corresponding state, that is, u_k depends on x_k.

The linear algebra problem of Equation 5.17 can be generalized to N stages and takes the following form:

$$\begin{bmatrix} N & (N-1)\lambda^2 & (N-2)\lambda^2 & & 3\lambda^2 & 2\lambda^2 & \lambda^2 \\ (N-1)\lambda^2 & N & (N-2)\lambda^2 & \cdots & 3\lambda^2 & 2\lambda^2 & \lambda^2 \\ (N-2)\lambda^2 & (N-2)\lambda^2 & N-1 & & 3\lambda^2 & 2\lambda^2 & \lambda^2 \\ & \vdots & & \ddots & & \vdots & \\ 3\lambda^2 & 3\lambda^2 & 3\lambda^2 & & 4 & 2\lambda^2 & \lambda^2 \\ 2\lambda^2 & 2\lambda^2 & 2\lambda^2 & \cdots & 2\lambda^2 & 3 & \lambda^2 \\ \lambda^2 & \lambda^2 & \lambda^2 & & \lambda^2 & \lambda^2 & 2 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-3} \\ u_{N-3} \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} N \\ N-1 \\ N-2 \\ \vdots \\ 3 \\ 2 \\ 1 \end{bmatrix} \lambda x_0 \tag{5.20}$$

Only the first control input u_0 is desired and we are interested in the limiting behavior of u_0 as $N \to \infty$. It would therefore be convenient to derive a recursive form for u_0. A recursive solution is possible, the details of which are omitted for brevity. The matrix on the left-hand side of Equation 5.20 can be tridiagonalized and individual equations of the complete system extracted to provide a three-term recursion for u_k. It is convenient to define $v_k = u_{k-N}$ and rewrite the recursion relation in terms of (v_{k+1}/v_k). (For convenience in this context, I depart from my previous stipulation that v_k be used to represent open-loop optimal control in MPC.) Additional algebraic manipulations provide the following:

$$\frac{v_2}{v_1} = - \frac{(4 - \lambda^2)}{(2\lambda^2 - 3)} \tag{5.21}$$

and for $k \in [2, N-1]$

$$\left(\frac{v_{k+1}}{v_k}\right) = \frac{\left[(2k-1)\,\lambda^2 - 2\,(k+1)\right] + \left[k - (k-1)\,\lambda^2\right]\left(\frac{v_{k-1}}{vk}\right)}{(k+1)\,\lambda^2 - (k+2)} \tag{5.22}$$

The first control input u_0 is provided by the following:

$$v_N = u_0 = \frac{-\,\lambda x_0 \left(\frac{v_N}{v_{N-1}}\right)}{\left[N+1-(N-1)\,\lambda^2\right]\left(\frac{v_N}{v_{N-1}}\right) + \left[(N-1)\,\lambda^2 - N\right]} \tag{5.23}$$

5.2.3 Other Control Strategies

Model predictive control and dynamic programming are not the only possible ways that we could design a controller for this system. For comparison, I consider two other possibilities: open loop optimal control and certainty equivalence, considered here briefly.

Open Loop Optimal Control. This method solves for an open loop optimal controller over a horizon of N stages, just as in model predictive control. Unlike MPC, the state is not updated as new data become available. Instead, we simply inject the open-loop optimal control moves calculated in the initial optimal control problem.

Because the solutions depend on a fixed value of N to compute the open loop control, this method is inherently suited only to problems with fixed final time.

Certainty Equivalence Control. This is termed certainty equivalence because of its similarity to the results of linear-quadratic-Gaussian (LQG) observer/controller methods. Rather than solving an optimization using the expectation of a stochastic objective function as in Equation 5.8, we pose a strictly deterministic objective and in place of the stochastic model equation, we insert the expected values of all stochastic variables. The state is updated as new measurements become available.

5.2.4 Comparison of Control Strategies

To compare performance of the various control strategies, the final time is fixed at $N = 20$. I examine the feedback control laws for the initial state and the expected value of the objective function over the entire horizon.

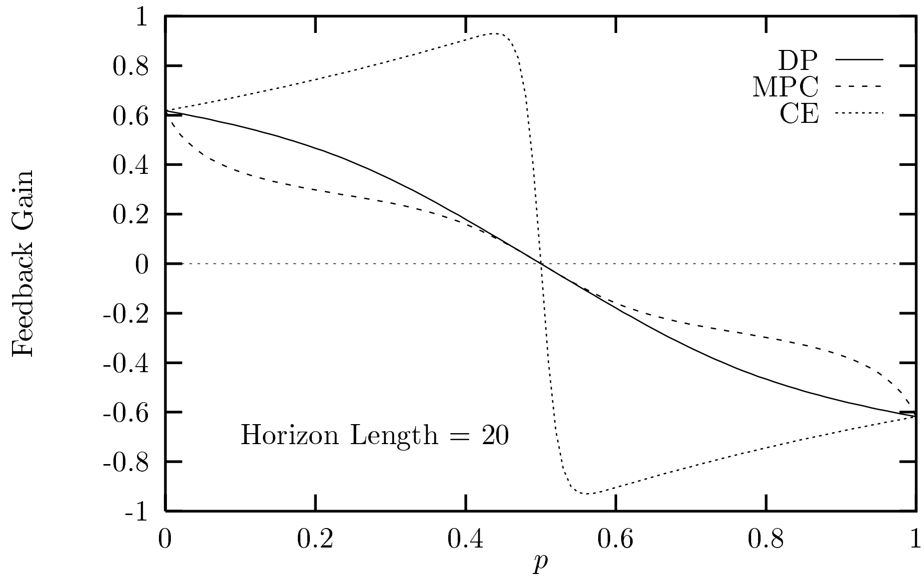

Figure 5.4: Feedback Gain for Various Control Strategies

The feedback control law at the initial time is illustrated in Figure 5.4 as a function of the distributional parameter p. Since the controls are all linear functions of the state, they are presented here as feedback gains.

Figure 5.4 shows that the three methods being compared yield the same result for $p = 0$, $p = 1/2$ and $p = 1$. For the two endpoint cases, the problem degenerates into a non-stochastic problem. All three methods give the same result in these strictly deterministic cases. For the case $p = 1/2$, all three method demand that the control be identically zero. For the "certainty equivalence" method this corresponds to an uncontrollable system. Any control action can only add to the cost without affecting the state of the system. For the other methods, $p = 1/2$ corresponds to a control gain which can have either sign with equal probability. The best control action in this case is also to set the control to zero. For $p = 1/2$, the zero control action eliminates the effect of the stochastic variable w_k and all three methods coincide, as reflected in the costs shown in Figures 5.5 and 5.6. Open loop optimal control is not shown in Figure 5.4 since the first control input is the same for both OLO control and MPC.

Figure 5.5 shows the costs associated with the various control strategies. Clearly, the certainty equivalence controller performs far worse than the other three methods on $p \in [0.25, 0.75]$. Figure 5.6 show the same results omitting the certainty equivalence method. Figures 5.5 and 5.6 illustrate that dynamic

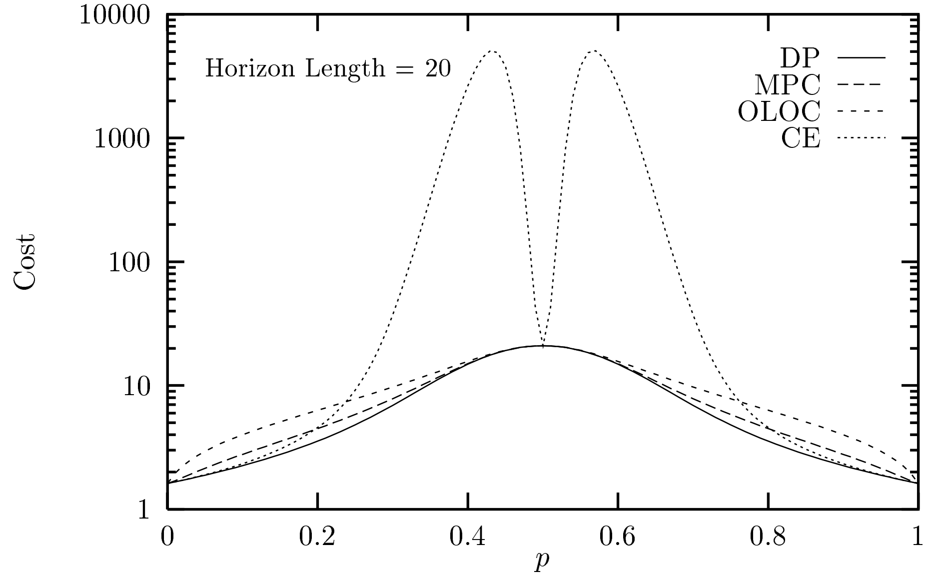

Figure 5.5: Expected Costs for Various Control Strategies

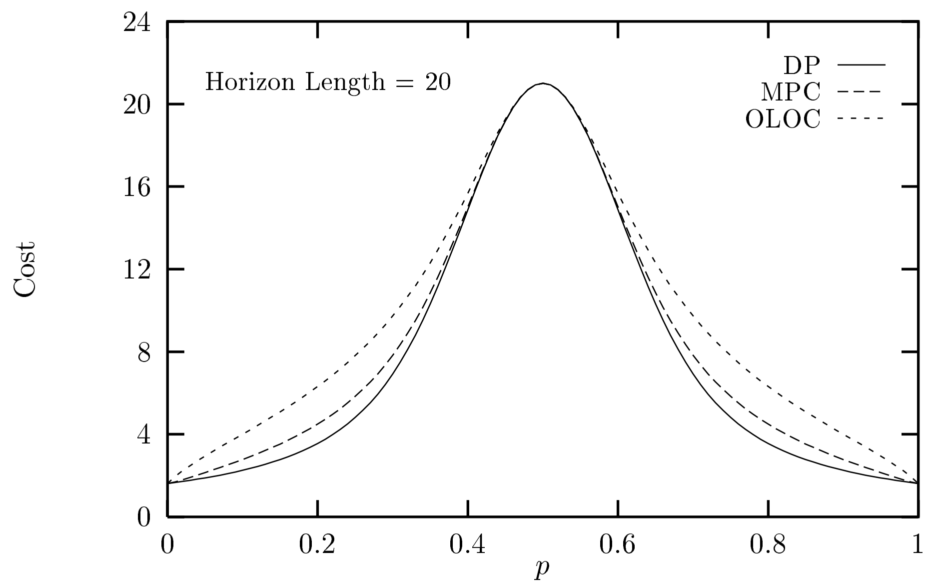

Figure 5.6: Expected Costs for Various Control Strategies

programming yields the lowest possible cost for stochastic feedback control systems as predicted by theory.

5.2.5 Infinite Horizon Implementation

In process control applications, only batch processes are naturally defined with finite horizons. In this section, I consider the application of model predictive control with an infinite horizon and compare the MPC performance to a dynamic programming-based controller.

An infinite number of control moves from an ordinary mathematical programming problem such as in Equation 5.8. However, since only the first move of the optimal sequence is implemented, it is possible to use the recursion of Equations 5.21 through 5.23 to examine the behavior of the infinite horizon model predictive controller and compare to the infinite horizon dynamic programming controller. When I refer to the infinite horizon controller in both of these cases, I am considering the controller given in the limit as $N \to \infty$ by Equations 5.21 through 5.23 and 5.14 through 5.16 for the model predictive and dynamic programming controllers, respectively.

Model Predictive Controller. To consider the limiting behavior of the recursion represented by Equation 5.21, we find that different behavior is observed for $p = 0$ or $p = 1$ than for p in the open interval $(0, 1)$. Consider first the cases of $p = 0$ or $p = 1$. For these cases, Equations 5.21 and 5.22 become

$$\frac{v_2}{v_1} = 3 \tag{5.24}$$

$$\left(\frac{v_{k+1}}{v_k}\right) = 3 - \left(\frac{v_{k-1}}{v_k}\right) \tag{5.25}$$

Analysis of Equation 5.25 show that, for (v_2/v_1) in $\left(\frac{3-\sqrt{5}}{2}, \infty\right)$, the ratio (v_{k+1}/v_k) converges to $\frac{3+\sqrt{5}}{2}$, or approximately 2.618. With $p = 1$, Equation 5.23 provides an infinite horizon gain of 0.618. When $p \in (0, 1)$, (v_{k+1}/v_k) converges to 1. From Equation 5.23, this gives a feedback gain that is identically 0 in the limit as $k \to \infty$ for all $p \in (0, 1)$.

Dynamic Programming Controller. Equations 5.14 and 5.15 do not provide a convergent sequence for c_k; however, the sequence $c_k / (1 + c_k)$ of Equation 5.11 used to compute the dynamic programming controller *does* converges rapidly to produce the feedback gain shown in Figure 5.7.

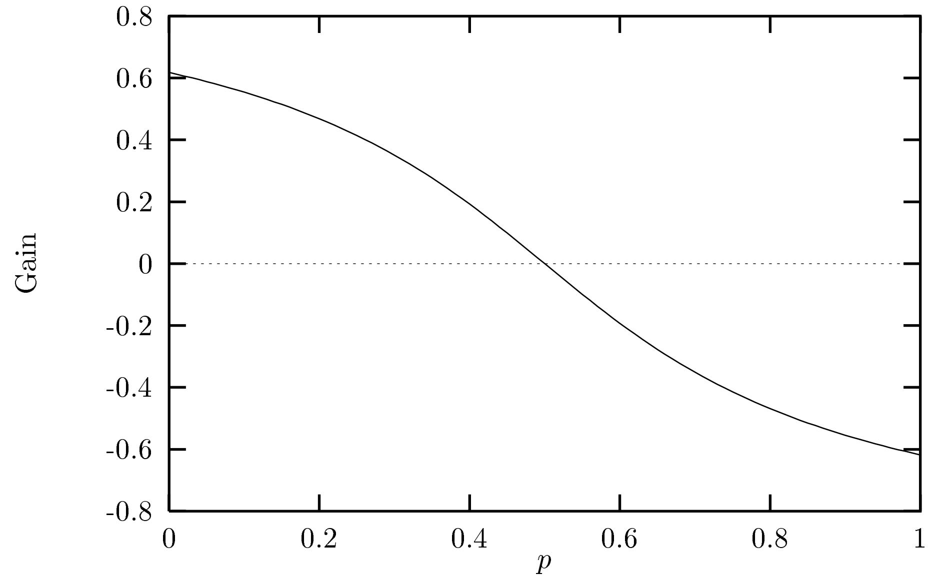

Figure 5.7: Infinite Horizon Feedback Gain from Dynamic Programming

Costs of Infinite Horizon Strategies. It is not possible to compute the infinite horizon cost function directly because it is unbounded for almost every p. Consider instead the *average* cost per stage resulting from the candidate control schemes. Once again, the costs of dynamic programming and model predictive control coincide for the three cases $p \in \{0, 0.5, 1\}$. The cost in both cases is a quadratic function of the initial state, $\phi_\infty = c\, x_0^2$, in which c is given by

$$c = \begin{cases} \left.\begin{array}{ll} 1 & p = \frac{1}{2} \\ 0 & \text{otherwise} \end{array}\right\} & \text{DP} \\ \\ \left.\begin{array}{ll} 0 & p = \{0, 1\} \\ 1 & \text{otherwise} \end{array}\right\} & \text{MPC} \end{cases}$$

5.3 Some Observations

In Sections 4.4–4.5, I demonstrated that for linear plants, the nominal stability properties of MPC with state feedback are sufficient to provide asymptotic stability in the presence of disturbances due to imperfect state knowledge. Converse Lyapunov theorems and results on stability under perturbations, such as Theorem 16, provide powerful tools to address the effects of other kinds of disturbances, such as those that arise from the interconnection of more general nonlinear estimators and regulators. Further research in this direction would seem warranted.

In Sections 5.2–5.2.5, the best possible control performance was obtained for a stochastic process example using the complete dynamic programming solution, which is available in feedback form as an algebraic equation. The second best was provided by open-loop optimal feedback control, which was identified with model predictive control. Ordinary open-loop optimal control without feedback performed worse than either of the previous two methods. This ordering is consistent with previously published results [4]. The certainty equivalence controller showed no clear ordering with respect to the other methods considered, but demonstrated performance that was orders of magnitude worse over some parameter ranges.

Based on the feasibility of computation, and the relative ordering of the possible control schemes, it seems clear that model predictive control for stochastic systems is a competitive approach for finite horizon control problems (i.e. batch processes). For infinite horizon control problems (i.e. continuous processes), the picture is less clear. I had hoped that by extending the horizon of the MPC scheme, it might be possible to approximate the infinite horizon behavior of the dynamic programming feedback controller. Keerthi and Gilbert [31] show that that these costs converge for the class of nonlinear, deterministic problems which they considered. Unfortunately, this is not the case for the stochastic problem presented here. The examples illustrate that it is not obvious how best to implement model predictive control on stochastic systems. This open question remains a topic of current research.

In the optimal path problem of Dreyfus and Law, the difference between the dynamic programming solution (called "optimal feedback" by those authors) and dynamic programming was less than 1 percent. Despite this small difference, they harshly characterized the model predictive approach, going so far as to state that "[t]here is little to recommend this scheme since it uses as much information as the true feedback solution. We presented it here to protect the reader from mistakenly believing that this is an alternative way of computing the true optimal feedback solution [13]."

This suggests that what we consider today to be a practical approach for many problems was rejected based on theoretical concerns. Yet despite this rejection, model predictive methods were embraced by industrial practitioners, while dynamic programming approaches have rarely been adopted. In [56], several possible reasons were offered as to why this has occurred, including educational and training barriers in understanding the dynamic programming approach, the difficulty in identifying stochastic models and the artificiality of the cost function.

The model predictive controller, on the other hand, requires only the solution of an open-loop optimization problem, a problem that has become practicable on-line with the decrease in computing cost and the development of efficient and reliable optimization codes. Even if not optimal, MPC can provide a stable feedback controller for a large class of linear or nonlinear systems, and it can address the practical issue of constraints on the inputs. The cost function, although artificial, can be connected to tuning parameters that are under the control of human operators for monitoring and on-line tuning. It is fair to say that although MPC may not have the aesthetic appeal of optimal control, it does address many practical issues that are still out of reach with dynamic programming and other methods.

Chapter 6

Conclusions

This work has focused on four main areas in the study of model predictive control: sufficient conditions for nominal stability with general nonlinear systems, improvements to the Rawlings-Muske approach for linear systems with constraints, application of theory of dynamic programming to analyze model predictive control and the application of MPC to stochastic or perturbed systems.

Stability of nonlinear systems controlled by MPC was shown to depend on the non-negativity and continuity of the MPC objective function. The objective acts as a Lyapunov function for the state trajectory of the controlled system. For nominal systems, a decreasing objective function can be obtained with a final state stability constraint using a prediction horizon or with an infinite prediction horizon. Through an example, I demonstrated that finite horizon problems do not necessarily converge to the infinite horizon problem, although other researchers [29, 28, 30, 31] have provided stronger sufficient conditions for convergence, that restrict the class of systems being considered.

Since continuity of the MPC objective is so important for stability, I examined in detail the behavior of a system controlled via MPC that demands a discontinuous controller for stability. This example provides strong insight into how the final state stability constraint can produce a discontinuous controller or MPC objective function. I provided a sufficient condition derived from the Implicit Function Theorem that assures continuity of the objective function and thus stability.

The MPC method of Rawlings and Muske [53, 54, 57] is maturing as an implementable method and has attracted interest from industrial practitioners. Their method requires the satisfaction of linear state constraints for an infinite number of states. Rawlings and Muske were able to show that constraints beyond a certain time index could not be active, but their result was state dependent. The work reported in this dissertation incorporates a result of Gilbert and Tan [19] that collapses the infinite set of constraints into an equivalent finite set that is state-independent.

Although this work resolves some important implementation issues, it also raises a concern that the controller may violate some sufficient conditions for stability in the presence of disturbances. This issue is not yet resolved but is the object of ongoing research [52] at the University of Texas.

I showed that important results from Linear-Quadratic (LQ) optimal control can be derived as a special case of the more general theory using dynamic programming. Using arguments involving monotonicity of the model predictive control objective, I extended an LQ stability result to the general nonlinear case. This provided a sufficient condition for stability that is a generalization of a well-known result for the Riccati difference equation of LQ optimal control. This work is still in the early stages and may yield significant future results.

Finally, I considered the application of model predictive control to problems involving stochastic and perturbed systems. One key result showed that a combined observer/controller is asymptotically stable if the controller is asymptotically stable and Lipschitz continuous and the observer is exponentially stable. This illustrated a "pseudo-separation principle" in action for nonlinear systems in which stability properties for the nominal controller and observer can be derived separately then combined to yield a stable combination.

For stochastic systems, the model predictive control method based on batch optimization was shown to be inferior to one designed using dynamic programming. Since current MPC methods depend on batch optimization, this indicates that MPC is suboptimal when compared to dynamic programming, which can be shown to provide *the* minimizing controller. This result, which has been in the literature since the 1960's with little notice by researchers or practitioners, was illustrated by two examples.

Despite the suboptimality of MPC when compared to full dynamic programming solutions, I continue to advocate that controllers be designed based on deterministic models. The stochastic nature of physical systems can then be analyzed in combination with linear or nonlinear observers by resorting to the pseudo-separation principle.

6.1 Future Work

Of the topics discussed in this dissertation, several provide clear avenues for future study:

- Separation Principle for Nonlinear Systems: Section 4.4 provides sufficient conditions for asymptotic stability of a combined observer/controller when the observer is exponentially stable and the nominal controller is asymptotically stable. It is unclear whether a similar result will hold for an observer that is only asymptotically stable rather than exponentially stable. It is of significant practical interest, since nonlinear observers are being developed that are not necessarily exponentially stable [43, 44, 48, 49, 52]. This case should be proven or a counterexample should be provided.

- Recurrent Stochastic Input: The observer/controller of Section 4.4 was posed as a deterministic problem in which the purpose of the observer was only to reconstruct an unknown initial state. More development work is needed to characterize the behavior of combined observer/controller pairs in the presence of recurrent disturbances. A significant contribution in that regard would simply be to define a problem in a stochastic framework that is relevant to control practice. The work of LaSalle [38] and Kushner [34] provide discussion of stability concepts, in terms of bounded disturbances, stability in probability and other definitions. The work of Kushner especially emphasize stochastic Lyapunov functions. It is not clear which of these address significant questions in control practice that should be addressed.

- Analysis of MPC through Dynamic Programming: This work demonstrated that dynamic programming can be used to analyze the MPC algorithm. A key result was that if the inequality

$$J \geq T(J) \tag{6.1}$$

  could be solved, then the controller $u(x) = \arg\ T^N(J)(x)$ would be stabilizing for all N. This follows from the monotonicity property of the operator T. An investigation of the solutions to the inequality of Equation 6.1 has a potentially large impact on the design of nonlinear model predictive controllers.

- Final State Constraint for Stability: The monotonicity property of T has important consequences when $J \leq T(J)$ or $J \geq T(J)$. It may be difficult to find initial J that satisfy these inequalities. What about the case where $T(J)$ an J satisfy no such ordering? If sufficient conditions could be found such that $T^N(J) \geq T^{N+1}(J)$ for some N despite having no ordering for J and $T(J)$, this would provide a sufficient condition for stability of and N-stage MPC control with final state penalty $J(z_N)$. For the linear-quadratic case, there is a wealth of results concerning initial conditions of the Riccati Difference Equation; generalizations for nonlinear systems using T would have immediate practical benefit.

- Improvements to Rawlings-Muske Controller: Experiments indicate that the discontinuity of the controller discussed in Section 4.3 may have little impact on the overall stability of a combined observer/controller, despite not satisfying the sufficient conditions of Theorem 16. More analysis is needed. It may be that a detailed analysis using stochastic systems theory would indicate that the zero measure of the regions where discontinuities exist negates their influence on the overall stability of the combined system.

- Stochastic MPC Design: The emphasis in this work has been to design MPC controllers using deterministic models. The results of Section 5 indicate that this approach produces suboptimal control. The literature on dynamic programming is devoted principally to the solution of stochastic optimal control problems. As shown in this dissertation, it is a rich trove for possible MPC applications. A broader overview of the existing literature in this area is needed to fill a large gap that exists between theoreticians and control practitioners in this area.

- Un-decoupled Observer/Controller: For simplicity of analysis, this work briefly discussed and analyzed a special case observer/controller pair that provide a pseudo-separation principle. Are there cases where no such separation is possible? If so, can a simultaneous observer/controller can be designed that jointly solves the observer and controller as one optimization? The idea would be that measurements are provided as input to an observer/controller algorithm that provides a controller, perhaps without the direct identification of a state variable.

Appendix A

Berge Concept of Semi-continuity

The discussion of Fiacco [17] concerning continuity of objective functions and solutions sets is taken from Berge [2]. The relevant theorems are presented in terms of the semi-continuity of point-to-set maps. Berge's semi-continuity definitions *do not* yield the familiar definitions discussed in elementary calculus classes, which greatly confounded some of my early efforts to understand the behavior of the Hermes example (Section 3.1). Since Fiacco refers to Berge for proofs of his continuity theorems and Berge's book is no longer in print, I am providing a basic discussion of the Berge semi-continuity concept here for reference.

The following is taken from Chapter VI of Berge [2]:

> "Let Γ be a mapping of a topological space X into a topological space Y and let x_0 be a point of X. We say the Γ is **lower semi-continuous at** x_0 if for each open set G meeting Γx_0, there is a neighbourhood $U(x_0)$ such that
>
> $$x \in U(x_0) \Rightarrow \Gamma x \cap G \neq \emptyset$$
>
> We say that Γ is **upper semi-continuous at** x_0 if for each open set G containing Γx_0 there exists a neighbourhood $U(x_0)$ such that
>
> $$x \in U(x_0) \Rightarrow \Gamma x \subset \cap G$$
>
> We say that the mapping Γ is **continuous at** x_0 if it is both lower and upper semi-continuous at x_0.

> If Γ is a single-valued mapping, the definition given above for lower semi-continuity coincides with the ordinary definition of continuity; the same is true for upper semi-continuity."

In the final quoted paragraph, Berge makes clear the distinction between semi-continuity concepts for point-to-set maps and single-valued functions. This distinction is not highlighted by Fiacco and is a trap for the unwary seeking to apply the results contained in his book.

Appendix B

Constraint Reduction Algorithm

Given a quadratic program of the form

$$\min \ \frac{1}{2}v^T R v + r^T v \tag{B.1}$$

$$\begin{array}{rrcl} \text{Subject to:} & Av & \leq & a \\ & Bv & = & b \\ & v & \in & \Re^n \end{array}$$

we seek to eliminate constraints that are redundant, meaning those rows of A and B which, when considered separately, are implied by the other constraints. We assume in the above that the constraints are consistent, meaning that there exists a feasible point to satisfy all of the constraints.

B.1 Inequality Constraints

Let n_b be the dimension of vector b and take $\{G\}_i$ to be the ith row of G. An algorithm which eliminates redundant constraints is given by the following:

1. Set $k = n_b$.

2. Solve the linear programming problem

$$J_k^* = \min \ \{B\}_k^T \, v \tag{B.2}$$

$$\begin{array}{rrcl} \text{Subject to:} & Av & \leq & a \\ & \tilde{B}_k v & = & \tilde{b}_k \\ & v & \in & \Re^n \end{array}$$

in which $\tilde{B}_k$ and $\tilde{b}_k$ represent the corresponding matrices or vectors with the kth row deleted.

3. If $J_k^* \leq \{b\}_k$, then reduce the constraint set by deleting the kth row of B and b.

4. If $k = 1$, stop.

5. Take $k = k + 1$ and return to Step 2.

B.2 Equality Constraints

The equality constraints present a simpler problem. Since the equality constraints are consistent, we merely delete rows of A individually and check the rank of the resulting $\tilde{A}_k$. If $\text{rank}\left(\tilde{A}_k\right) = \text{rank}\,(A_k)$, then the kth row of A and a can be deleted from the constraint set.

Bibliography

[1] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, New Jersey, 1957.

[2] C. Berge, *Topological Spaces, including a treatment of Multi-Valued Functions, Vector Spaces and Convexity*, Oliver & Boyd, Edinburgh and London, 1963.

[3] D. P. Bertsekas, *Monotone mappings with applications in dynamic programming*, SIAM J. Cont. Opt., 15 (1977), pp. 438–464.

[4] ——, *Dynamic Programming*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[5] D. P. Bertsekas and S. E. Shreve, *Stochastic Optimal Control: The Discrete Time Case*, Academic Press, New York, 1978.

[6] R. R. Bitmead, M. Gevers, and V. Wertz, *Adaptive Optimal Control, The Thinking Man's GPC*, Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

[7] R. R. Bitmead, M. R. Gevers, I. R. Petersen, and R. J. Kaye, *Monotonicity and stabilizability properties of solutions of the Riccati difference equation: Propositions, lemmas, theorems, fallacious conjectures and counterexamples*, Sys. Cont. Let., 5 (1985), pp. 309–315.

[8] R. W. Brockett, *Asymptotic stability and feedback stabilization*, in Differential Geometric Control Theory, R. W. Brockett, R. S. Millman, and H. J. Sussmann, eds., Birkhauser, 1983, pp. 181–191.

[9] C. C. Chen and L. Shaw, *On receding horizon control*, Automatica, 16 (1982), pp. 349–352.

[10] J.-M. Coron, *A necessary condition for feedback stabilization*, Sys. Cont. Let., 14 (1990), pp. 227–232.

[11] C. E. de Souza, M. R. Gevers, and G. C. Goodwin, *Riccati equation in optimal filtering of nonstabilizable systems having singular state transition matrices*, IEEE Trans. Auto. Cont., 31 (1986), pp. 831–838.

[12] S. E. Dreyfus, *Dynamic Programming and the Calculus of Variations*, Academic Press, New York, 1965.

[13] S. E. Dreyfus and A. M. Law, *The Art and Theory of Dynamic Programming*, Academic Press, New York, 1977.

[14] J. W. Eaton, *Finite Horizon Predictive Control of Chemical Processes*, PhD thesis, University of Texas at Austin, 1992.

[15] J. W. Eaton, *Octave—a high-level interactive language for numerical computations*, Tech. Rep. CCSR-93-003, The University of Texas Center for Control and Systems Research, Austin, Texas 78712, 1993.

[16] T. F. Edgar and D. M. Himmelblau, *Optimization of Chemical Processes*, McGraw-Hill, New York, 1988.

[17] A. V. Fiacco, *Introduction to Sensitivity Analysis in Nonlinear Programming*, Academic Press, New York, 1983.

[18] C. E. García, D. M. Prett, and M. Morari, *Model predictive control: Theory and practice—a survey*, Automatica, 25 (1989), pp. 335–348.

[19] E. G. Gilbert and K. T. Tan, *Linear systems with state and control constraints: The theory and application of maximal output admissible sets*, IEEE Trans. Auto. Cont., 36 (1991), pp. 1008–1020.

[20] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's guide for SOL/NPSOL (Version 4.0): A Fortran package for nonlinear programming, technical report SOL 86-2*, tech. rep., Systems Optimization Laboratory, Department of Operations Research, Stanford University, 1986.

[21] G. H. Golub and C. F. V. Loan, *Matrix Computations*, Johns Hopkins, Baltimore, 1989.

[22] W. W. Hager, *Lipschitz continuity for constrained processes*, SIAM J. Cont. Opt., 17 (1979), pp. 321–338.

[23] A. Halanay, *Quelques questions de la théorie de la stabilité pour les systèmes aux différences finies*, Arch. Rational Mech. Anal., 12 (1963), pp. 150–154.

[24] H. Hermes, *Asymptotically stabilizing feedback controls and the nonlinear regulator problem*, SIAM J. Cont. Opt., 29 (1991), pp. 185–196.

[25] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[26] R. E. Kalman, *Contributions to the theory of optimal control*, Bull. Soc. Math. Mex., 5 (1960), pp. 102–119.

[27] R. E. Kalman and J. E. Bertram, *Control system analysis and design via the "Second method" of Lyapunov, Part II: Discrete–time systems*, ASME J. Basic Engr., (1960), pp. 394–400.

[28] S. S. Keerthi, *Optimal Feedback Control of Discrete-Time Systems with State-Control Constraints and General Cost Functions*, PhD thesis, University of Michigan, 1986.

[29] S. S. Keerthi and E. G. Gilbert, *An existence theorem for discrete-time infinite-horizon optimal control problems*, IEEE Trans. Auto. Cont., 30 (1985), pp. 907–909.

[30] ——, *Moving-horizon approximations for a general class of optimal nonlinear infinite-horizon discrete-time systems*, in Proceedings of the 20th Annual Conference on Information Sciences and Systems, Princeton, NJ, 1986, Princeton University, pp. 301–306.

[31] ——, *Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations*, J. Optim. Theory Appl., 57 (1988), pp. 265–293.

[32] D. L. Kleinman, *An easy way to stabilize a linear constant system*, IEEE Trans. Auto. Cont., 15 (1970), p. 692.

[33] D. L. Kleinman, *Stabilizing a discrete, constant, linear system with application to iterative methods for solving the riccati equation*, IEEE Trans. Auto. Cont., 19 (1974), pp. 252–254.

[34] H. J. Kushner, *Stochastic Stability and Control*, vol. 33 of Mathematics in Science and Engineering, Academic Press, New York, 1967.

[35] W. H. Kwon, A. M. Bruckstein, and T. Kailath, *Stabilizing state-feedback design via the moving horizon method*, Int. J. Control, 37 (1983), pp. 631–643.

[36] W. H. Kwon and A. E. Pearson, *A modified quadratic cost problem and feedback stabilization of a linear system*, IEEE Trans. Auto. Cont., 22 (1977), pp. 838–842.

[37] ——, *On feedback stabilization of time-varying discrete linear systems*, IEEE Trans. Auto. Cont., 23 (1978), pp. 479–481.

[38] J. P. LaSalle, *The stability of dynamical systems*, in Regional Conference Series in Applied Mathematics #25, SIAM, 1976.

[39] D. G. Luenberger, *Optimization by Vector Space Methods*, John Wiley and Sons, Inc., New York, 1969.

[40] D. Q. Mayne and H. Michalska, *An implementable receding horizon controller for stabilization of nonlinear systems*, in Proceedings of the 29th Conference on Decision and Control, December 1990, pp. 3396–3397.

[41] ——, *Receding horizon control of nonlinear systems*, IEEE Trans. Auto. Cont., 35 (1990), pp. 814–824.

[42] ——, *Model predictive control of nonlinear systems*, in Proceedings of the 1991 American Control Conference, June 1991, pp. 2343–2348.

[43] D. Q. Mayne and H. Michalska, *Moving horizon, observer-based control*, in Proc. 31st IEEE Conference on Decision and Control, Tulsa, Oklahoma, 1992.

[44] ——, *Moving horizon, observer-based control*, in Abstracts, Washington University-NSF Workshop on Nonlinear Control Systems, Washington University, May 1992.

[45] E. S. Meadows, M. A. Henson, J. W. Eaton, and J. B. Rawlings, *Receding horizon control and discontinuous state feedback stabilization*. Accepted for publication in *Int. J. Cont.*, 1994.

[46] E. S. Meadows, K. R. Muske, and J. B. Rawlings, *Constrained state estimation and discontinuous feedback in model predictive control*, in Proceedings of the 1993 European Control Conference, European Automatic Control Council, June 1993, pp. 2308–2312.

[47] H. Michalska and D. Q. Mayne, *Receding horizon control of nonlinear systems without differentiability of the optimal value function*, Sys. Cont. Let., 16 (1991), pp. 123–130.

[48] H. Michalska and D. Q. Mayne, *Moving horizon observers*, in Proc. IFAC Symposium on Nonlinear Control Systems Design, Bordeaux, France, 1992, pp. 576–581.

[49] ——, *Moving horizon observers and observer-based control*, Tech. Rep. UCD-EECS-SCR-92/8, College of Engineering, University of California, Davis, September 1992.

[50] H. Michalska and D. Q. Mayne, *Robust receding horizon control of constrained nonlinear systems*, IEEE Trans. Auto. Cont., 38 (1993), pp. 1623–1633.

[51] ——, *Moving horizon observers and observer-based control.* Accepted for publication in *IEEE Trans. Auto. Cont.*, 1994.

[52] K. R. Muske, *Linear Model Predictive Control of Chemical Processes*, PhD thesis, The University of Texas at Austin, 1994.

[53] K. R. Muske and J. B. Rawlings, *Linear model predictive control of unstable processes*, J. Proc. Cont., 3 (1993), pp. 85–96.

[54] ——, *Model predictive control with linear models*, AIChE J., 39 (1993), pp. 262–287.

[55] ——, *Nonlinear moving horizon state estimation*, in NATO Advanced Study Institute on Methods of Model Based Control Proceedings, R. Berber, ed., Antalya, Turkey, July 1994, Kluwer Academic Publishers.

[56] J. B. Rawlings, E. S. Meadows, and K. R. Muske, *Nonlinear model predictive control: A tutorial and survey*, in ADCHEM '94 Proceedings, Kyoto, Japan, 1994.

[57] J. B. Rawlings and K. R. Muske, *Stability of constrained receding horizon control*, IEEE Trans. Auto. Cont., 38 (1993), pp. 1512–1516.

[58] E. D. Sontag, *Mathematical Control Theory*, Springer-Verlag, New York, 1990.

[59] Y. A. Thomas, *Linear quadratic optimal estimation and control with receding horizon*, Electron. Lett., 11 (1975), pp. 19–21.

[60] M. Vidyasagar, *Nonlinear Systems Analysis*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 2nd ed., 1993.

Vita

Edward Scott Meadows, Jr. was born in Tulsa, Oklahoma on May 18, 1962, the first son of Edward S. Meadows and Juana Maria Brown Meadows. The first decade of his life was spent in Oklahoma, Florida, South Carolina, and Germany as an Air Force dependent. Returning to Oklahoma in 1972, the family settled in Ardmore in 1974, where he graduated from Ardmore High School in 1980. He entered the University of Oklahoma in the Fall of 1980 and received the degree of Bachelor of Science in Chemical Engineering in 1984.

He was commissioned as a Naval Officer in 1984 in Oklahoma City and was assigned to the U.S. Navy's Division of Naval Reactors in Washington, D.C. Following his first tour at Naval Reactors, he left active duty in 1988 to return to graduate school at the University of Texas at Austin. He remained affiliated with the Naval Reserve and commanded two different Naval Reserve Units while studying in Austin. He received his Master of Science degree from the University of Texas in 1993 .

Permanent address: 4159 Steck Avenue, #133
Austin TX 78759-8515

This dissertation was typeset by the author using the LaTeX document preparation system.