

Polynomial Controllers for Optimal Trajectory Matching with Stability Guarantees

by

Alexander Kitaev

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2025

© Alexander Kitaev 2025

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We formulate a trajectory matching problem in which a set of reference trajectories for a plant is given, and a control law that causes the plant's trajectories to be as close as possible to the reference trajectories is desired. These trajectories might be generated by an implicit controller such as a model predictive control (MPC) algorithm or manually chosen by a user. This thesis presents a nonconvex optimization approach for solving the trajectory matching problem that generates explicit polynomial controllers. The value of this approach is that the explicit control laws it generates are simpler to implement, and can be used for stability analysis. Additionally, the method presented in this thesis guarantees local stability of the generated controller by ensuring local contractivity towards the generated trajectories.

This thesis presents several theoretical results that justify the method described here. Firstly, a proof that the local contractivity constraints used to ensure local stability can be expressed as a set of matrix inequalities is presented, which turns an infinite set of constraints into a finite one. Secondly, a theorem that describes how symmetries in the trajectory matching problem correspond to symmetries in its solution is presented and proven, which enables a reduction in the control design problem size and resulting solution.

Finally, this thesis demonstrates the method it describes on two example problems motivated by real-world applications. The first of these is stabilization and disturbance recovery for a single-machine infinite-bus (SMIB) power system, and the second is a lane change manoeuvre for Dubin's vehicle, a simple vehicle model. In each case, the reference trajectories are generated by MPC.

Acknowledgements

In no particular order:

I would like to thank my supervisors, Profs. Michael Fisher and Christopher Nielsen, not only for their invaluable advice, but for their patience and encouragement.

I would like to thank all of my fellow students in the DOCS Group for all the help we offered, all the questions we answered, all the jokes we shared, and all the memories we made.

I would like to thank every member of the FASS Theatre Company at the University of Waterloo for breaking up the monotony of graduate school, helping me build confidence and grow out of my shell, and giving me a space to be myself.

Finally, I would like to thank all my friends for their endless support, for their encouragement and motivation, and for always being there for me.

Dedication

To my siblings, Nadia and Michael.

Table of Contents

Author's Declaration	ii
Abstract	iii
Acknowledgements	iv
Dedication	v
List of Figures	viii
List of Abbreviations	ix
List of Symbols	x
1 Introduction	1
1.1 Motivating Example	3
1.2 Problem Statement	4
1.3 Terminology and Notation	6
1.4 Outline and Contributions	7
2 Literature Review	8
2.1 Model Predictive Control	8
2.1.1 Explicit MPC	9

2.2	System Level Synthesis	10
2.3	System Identification	10
2.4	Inverse Optimal Control	11
3	Theoretical Results	12
3.1	The Static Controller Case	12
3.2	Stability Guarantees	16
3.2.1	Transverse Stability Guarantees	19
3.3	Symmetries of Optimal Controllers	20
4	Numerical Results	26
4.1	Solution Methods	26
4.2	Examples	27
4.2.1	Single-Machine Infinite-Bus Power System	27
4.2.2	Dubin's Vehicle	28
5	Conclusion	34
5.1	Future Work	34
	References	36

List of Figures

1.1	A block diagram of the closed-loop system considered in this work.	5
3.1	The commutative diagram for Definition 3.7.	20
3.2	The commutative diagram for Theorem 3.10	23
4.1	The squared errors over time for each reference trajectory on the SMIB system.	30
4.2	The trajectories generated by the linear and nonlinear controllers for the Dubin's vehicle example for each reference trajectory.	31
4.3	The control inputs generated by the linear and controllers versus time for the Dubin's vehicle example.	32
4.4	The squared position errors for the linear and controllers versus time for the Dubin's vehicle example.	33

List of Abbreviations

HJB Hamilton–Jacobi–Bellman

IOC inverse optimal control

KKT Karush–Kuhn–Tucker

MPC model predictive control

PDIC post-disturbance initial condition

SLS system level synthesis

SMIB single-machine infinite-bus

List of Symbols

$\text{GL}_n(\mathbb{R})$ The group of invertible n -by- n matrices with real entries

Ker Kernel

$\otimes$ Kronecker product

$A \preceq B$ $A - B$ is negative semidefinite

$A \prec 0$ A is negative definite

$A \preceq 0$ A is negative semidefinite

$A \succeq B$ $A - B$ is positive semidefinite

$A \succ 0$ A is positive definite

$A \succeq 0$ A is positive semidefinite

$A_{:j}$ The j -th column of the matrix A

$A_{i:}$ The i -th row of the matrix A

$\mathbb{Z}_{\geq 0}$ The set of nonnegative integers

$\mathbb{S}_n$ The set of symmetric n -by- n matrices

Chapter 1

Introduction

A *discrete-time control system* is an equation of the following form:

$$\mathbf{x}[t + 1] = \mathbf{f}(\mathbf{x}[t], \mathbf{u}[t]), \quad (1.1)$$

where $\mathbf{x}[t] \in \mathbb{R}^n$ is the system's *state vector* at time t , $\mathbf{u}[t] \in \mathbb{R}^m$ is the *control input vector* at time t , and $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is the *state transition function* or *state transition map*. The entire time series $\mathbf{x}[\cdot]$ is called the *state trajectory*, and the time series $\mathbf{u}[\cdot]$ is called the *control input trajectory*. In the context of control engineering, an externally determined control system, which a control engineer cannot change, is called a *plant*.

An interesting problem in control system design is the *trajectory matching problem*: given one or more pairs of corresponding state and control input trajectories, called the *reference trajectories*, how can a control law be found that causes the plant's actual state and control input trajectories to be as close as possible to the desired ones? Trajectory matching has many real-world applications: for example, a high-level path-planning problem might generate a desired trajectory for a robotic manipulator or a mobile robot, and then a controller must be found that causes the system to follow that trajectory.

The trajectory matching problem can be casted an *optimal control problem*, a problem of finding a control scheme that optimizes some objective function, subject to dynamic constraints and possibly constraints on the plant states and inputs. In the trajectory matching problem, the objective function is a measure of the mismatch between the given reference trajectory and the actual state trajectory achieved by the plant with a given control input trajectory. The objective might also penalize high control input magnitudes due to practical implementation concerns.

The trajectory matching problem can be used to complement *model predictive control* (MPC), which is discussed further in Section 2.1. In general, MPC finds the control input trajectory that optimizes some given objective function, but it does not generate an explicit control law. Once MPC has been used to find an optimal control input trajectory, and therefore an optimal state trajectory, the trajectory matching problem can then be posed to find an explicit control law that makes the actual state trajectory as close as possible to the state trajectory found by MPC.

A key difference between this work and others is that this work poses the problem of finding *explicit* control laws; that is, the goal of this work is to find an explicit equation for the control inputs at time t as a function of the state trajectory up to time t and of previous control inputs. This controller might be represented as a dynamical system itself. There are several reasons why an explicit control law might be desired: for example, the computational resources available to the controller might be insufficient to solve an online optimization problem, or an explicit control law might be desired for stability or robustness analysis.

The general trajectory matching problem is complicated by the fact that the space of controllers has infinite dimension. Therefore, a special case of the problem, where the allowable controllers are restricted to a finite-dimensional subspace parametrized by finitely many coordinates, is preferred. The subspace chosen here is the space of polynomial dynamical systems: that is, ones where the state transition function is a multivariable polynomial function. There are several reasons for this choice. On one hand, the space of polynomial functions has many desirable properties: for example, it is closed under addition, subtraction, multiplication, composition, and differentiation. On the other hand, polynomial functions are able to describe a wide range of behaviours: indeed, Taylor's theorem states that an extremely broad class of functions can be approximated arbitrarily well by a polynomial of sufficient degree.

Another restriction is limiting the plant's state transition function to *control-affine* functions: that is, functions of the form

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{a}(\mathbf{x}) + B(\mathbf{x})\mathbf{u}, \quad (1.2)$$

where the function $\mathbf{f}$ is decomposed into polynomial functions $\mathbf{a} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $B : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$. A system with such a state transition function is called a *control-affine system*. The name comes from the fact that for a fixed $\mathbf{x}$, the state transition function is an affine function of the control input vector $\mathbf{u}$. This restriction might seem very limiting, but it turns out to lose almost no generality, since any system can be turned into a control-affine one by adding plant states.

One advantage of explicit controllers is their greater simplicity of implementation compared to approaches that do not generate explicit control laws such as MPC. Another advantage is the greater ease of stability analysis. Finally, a significant advantage of explicit controllers is the ability to impose *stability guarantees* on the optimal control problem: that is, adding constraints to the optimization problem that ensure the closed-loop system is stable in some sense. The sense of stability used in this work is *contractivity*: given a point in a neighbourhood of a reference trajectory, does the closed-loop trajectory starting at this point converge towards that reference trajectory? Such a property is very desirable for the trajectory matching problem, since it ensures asymptotic tracking: that is, that the reference trajectories will still be approximately followed even if a disturbance causes the actual plant state to deviate from the reference trajectory. One of the key results of this work is expressing the contractivity requirement as a system of inequalities that can be incorporated as constraints in an optimal control problem, ensuring that any solution to that problem comes with contractivity guarantees.

This work is concerned only with discrete-time systems; there are two reasons for this. Firstly, the vast majority of controllers are implemented digitally, which means that any continuous-time control design will need to be discretized anyway before it can be implemented in the real world. Secondly, working in discrete time allows the explicit representation of arbitrary trajectories as time series in state space, since a discrete-time trajectory with a finite time horizon is finite, and the optimization problems discussed in this work rely very heavily on such representations.

1.1 Motivating Example

An example of a real-world trajectory matching problem is the disturbance recovery problem for a *single-machine infinite-bus* (SMIB) power system, a simple power system consisting of a single synchronous generator connected to an infinite bus (that is, an ideal voltage source) by a lossless transmission line. Suppose that this system, initially at an equilibrium representing a desired operating point, is disturbed in some way: in the field of power systems, this disturbance is often a step which almost instantaneously places the system's state in a new point known as a *post-disturbance initial condition* (PDIC). System operators often maintain lists of credible worst-case contingencies to design for, such as worst-case step changes in load or generation at various points in the network.

Suppose that a set of PDICs has been obtained by simulating each credible worst-case contingency, and for each such PDIC, an engineer has selected a reference trajectory from that PDIC back to the equilibrium point. Additionally, suppose that an explicit control law

is desired for reliability or simplicity of implementation. The disturbance recovery problem then becomes the problem of finding an explicit control law that causes the SMIB system's trajectory from each PDIC to follow the corresponding reference trajectory back to the equilibrium point. The SMIB system is control-affine, so this problem exactly matches the class of problems considered in this thesis. The SMIB system and this work's application to it will be discussed in more detail in Section 4.2.1.

1.2 Problem Statement

Throughout this thesis, we consider plants of the following form:

$$\mathbf{x}[t+1] = \mathbf{a}(\mathbf{x}[t]) + B(\mathbf{x}[t])\mathbf{u}[t], \quad (1.3)$$

where $\mathbf{x}[t] \in \mathbb{R}^n$ is the *plant state* at time t , $\mathbf{u}[t] \in \mathbb{R}^m$ is the control input at time t , and $\mathbf{a} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $B : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ are polynomial functions. The class of allowable controllers of the following form:

$$\begin{aligned} \mathbf{z}[t+1] &= \mathbf{f}(\mathbf{z}[t], \mathbf{x}[t]) \\ \mathbf{u}[t] &= \mathbf{h}(\mathbf{z}[t], \mathbf{x}[t]), \end{aligned} \quad (1.4)$$

where $\mathbf{z}[t] \in \mathbb{R}^{n_c}$ is the *controller state* at time t and $\mathbf{f} : \mathbb{R}^{n_c} \times \mathbb{R}^n \rightarrow \mathbb{R}^{n_c}$ and $\mathbf{h} : \mathbb{R}^{n_c} \times \mathbb{R}^n \rightarrow \mathbb{R}^m$ are polynomial functions. Throughout this thesis, unless otherwise stated, n refers to the dimension of the plant state space, m refers to the number of control inputs of the plant, and n_c refers to the dimension of the controller state space. The value of n_c is not fixed *a priori*; it is selected by the designer as a hyperparameter. The term *closed-loop system* refers to the system formed by connecting systems (1.3) and (1.4) in feedback, as shown in Figure 1.1:

$$\begin{aligned} \mathbf{x}[t+1] &= \mathbf{a}(\mathbf{x}[t]) + B(\mathbf{x}[t])\mathbf{h}(\mathbf{z}[t], \mathbf{x}[t]) \\ \mathbf{z}[t+1] &= \mathbf{f}(\mathbf{z}[t], \mathbf{x}[t]), \end{aligned} \quad (1.5)$$

and the *closed-loop state* refers to the state of the closed-loop system: that is, the vector $\mathbf{x}_{CL} := [\mathbf{x}^T \ \mathbf{z}^T]^T \in \mathbb{R}^{n+n_c}$. The *closed-loop state transition map*, or simply the *closed-loop map*, refers to the function that sends each $\mathbf{x}_{CL}[t]$ to the corresponding $\mathbf{x}_{CL}[t+1]$.

We are given a plant of the form in Equation (1.3), as well as N pairs of state and control input trajectories $(\mathbf{x}_{\text{ref}}^j[t], \mathbf{u}_{\text{ref}}^j[t])$, $1 \leq j \leq N$, $0 \leq t \leq T$, each of length T ¹. We

¹The requirement that each reference trajectory have the same length is purely for simplicity of implementation; the method described in this thesis generalizes to reference trajectories of different lengths.

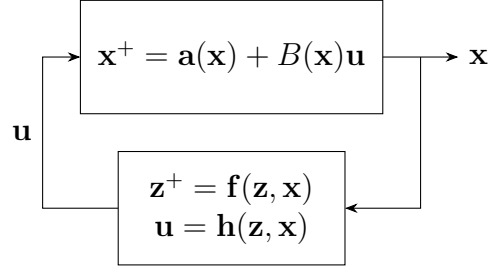


Figure 1.1: A block diagram of the closed-loop system considered in this work.

refer to these given trajectories as the *reference trajectories* or the *desired trajectories*. For each reference trajectory, the *actual trajectory* $(\mathbf{x}^j[t], \mathbf{u}^j[t])$ is the plant's state trajectory when its control inputs are generated by a candidate control law and the initial plant state is the starting point of the corresponding reference trajectory. Suppose that we are interested in controllers of the form in (1.4). We are also given a cost function of the following form:

$$q(\mathbf{f}, \mathbf{h}, \mathbf{x}^j, \mathbf{z}^j) = \sum_{j=1}^N \sum_{t=0}^{T-1} \|\mathbf{x}^j[t] - \mathbf{x}_{\text{ref}}^j[t]\|_Q^2 + \|\mathbf{u}^j[t] - \mathbf{u}_{\text{ref}}^j[t]\|_R^2, \quad (1.6)$$

where $Q \in \mathbb{S}_n$ and $R \in \mathbb{S}_m$ are positive definite. This cost function penalizes deviations between the desired and actual trajectories, as well as differences between the desired and actual control inputs. We seek to solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{f}, \mathbf{h}, \mathbf{x}^j, \mathbf{z}^j} \sum_{j=1}^N \sum_{t=0}^{T-1} \|\mathbf{x}^j[t] - \mathbf{x}_{\text{ref}}^j[t]\|_Q^2 + \|\mathbf{u}^j[t] - \mathbf{u}_{\text{ref}}^j[t]\|_R^2 \\ \text{subject to} \end{aligned} \quad (1.7)$$

$$\begin{aligned} \mathbf{f}(\mathbf{z}^j[t], \mathbf{x}^j[t]) &= \mathbf{z}^j[t+1], \quad 1 \leq j \leq N, \quad 0 \leq t < T \\ \mathbf{a}(\mathbf{x}^j[t]) + B(\mathbf{x}^j[t])\mathbf{h}(\mathbf{z}^j[t], \mathbf{x}^j[t]) &= \mathbf{x}^j[t+1], \quad 1 \leq j \leq N, \quad 0 \leq t < T, \end{aligned}$$

where $Q \in \mathbb{S}_n$ and $R \in \mathbb{S}_m$ are positive definite. Additionally, we seek to solve the above problem with additional constraints that guarantee that the closed-loop map is contracting towards the actual trajectories in the following sense: let $\hat{\mathbf{x}}^j[t]$ be the trajectory realized by the closed-loop system with a candidate controller starting from an initial condition in a neighbourhood of $\mathbf{x}^j[0]$. We seek to ensure that

$$\|\hat{\mathbf{x}}_{CL}^j[t+1] - \mathbf{x}_{CL}^j[t+1]\|_{P[t]} - \|\hat{\mathbf{x}}_{CL}^j[t] - \mathbf{x}_{CL}^j[t]\|_{P[t]} \leq -\|\hat{\mathbf{x}}_{CL}^j[t] - \mathbf{x}_{CL}^j[t]\|_{P'}, \quad 1 \leq j \leq N, \quad 0 \leq t < T \quad (1.8)$$

for some $P[t] \succ 0, P' \succ 0 \in \mathbb{S}_{n+n_c}$, $0 \leq t < T$. In terms of $\mathbf{x}$ and $\mathbf{z}$, the above becomes

$$\begin{aligned} & \begin{bmatrix} \hat{\mathbf{x}}^j[t+1]^T - \mathbf{x}^j[t+1]^T & \hat{\mathbf{z}}^j[t+1]^T - \mathbf{z}^j[t+1]^T \end{bmatrix} P[t] \begin{bmatrix} \hat{\mathbf{x}}^j[t+1] - \mathbf{x}^j[t+1] \\ \hat{\mathbf{z}}^j[t+1] - \mathbf{z}^j[t+1] \end{bmatrix} \\ & - \begin{bmatrix} \hat{\mathbf{x}}^j[t]^T - \mathbf{x}^j[t]^T & \hat{\mathbf{z}}^j[t]^T - \mathbf{z}^j[t]^T \end{bmatrix} P[t] \begin{bmatrix} \hat{\mathbf{x}}^j[t] - \mathbf{x}^j[t] \\ \hat{\mathbf{z}}^j[t] - \mathbf{z}^j[t] \end{bmatrix} \\ & \leq - \begin{bmatrix} \hat{\mathbf{x}}^j[t]^T - \mathbf{x}^j[t]^T & \hat{\mathbf{z}}^j[t]^T - \mathbf{z}^j[t]^T \end{bmatrix} P' \begin{bmatrix} \hat{\mathbf{x}}^j[t] - \mathbf{x}^j[t] \\ \hat{\mathbf{z}}^j[t] - \mathbf{z}^j[t] \end{bmatrix}, \quad 1 \leq j \leq N, \quad 0 \leq t < T. \quad (1.9) \end{aligned}$$

1.3 Terminology and Notation

Throughout this work, scalar quantities are written in lowercase italics (e.g. a), vector quantities are written in lowercase boldface (e.g. $\mathbf{a}$), and matrix quantities are written in uppercase italics (e.g. A). The symbol used for functions is based on their codomain: that is, a function from $\mathbb{R}^n$ to $\mathbb{R}^{n \times m}$ would be represented with an uppercase italic letter.

Much of this work is concerned with multivariate polynomials, so specific notation to simplify working with these polynomials is introduced. For positive integers n and k , the *index set* $I(n, k)$ is defined as

$$I(n, k) := \{\alpha \in \mathbb{Z}_{\geq 0}^n \text{ s.t. } \|\alpha\|_1 \leq k\}. \quad (1.10)$$

That is, $I(n, k)$ is the set of possible powers of each variable in a multivariate polynomial over n variables of degree at most k . Elementary combinatorics shows that the number of elements in $I(n, k)$ is

$$|I(n, k)| = \binom{n+k}{k} = \frac{(n+k)!}{n! k!}. \quad (1.11)$$

For convenience, if $\mathbf{v} \in \mathbb{R}^n$ and $\alpha \in I(n, k)$, define:

$$\mathbf{v}^\alpha := \prod_{i=1}^n v_i^{\alpha_i}, \quad (1.12)$$

where v_i is the i th component of $\mathbf{v}$. This notation allows a multivariate polynomial, e.g. $p: \mathbb{R}^n \rightarrow \mathbb{R}$ of degree at most k , to be written succinctly in terms of its coefficients:

$$p(\mathbf{x}) = \sum_{\alpha \in I(n, k)} p_\alpha \mathbf{x}^\alpha. \quad (1.13)$$

This work uses square brackets (*e.g.* $\mathbf{x}[t]$) to refer to time-varying quantities; the value in the brackets is the number of timesteps since a given starting point. Any quantity written with square brackets should be understood to be time-varying. When multiple reference trajectories are considered, they are differentiated by superscripts: for example, $\mathbf{x}^j[t]$ is the value of the plant state at time t starting at the j th initial condition.

For a symmetric positive definite matrix P and a vector $\mathbf{x} \in \mathbb{R}^n$, the notation $\|\mathbf{x}\|_P$ is defined as $\sqrt{\mathbf{x}^T P \mathbf{x}}$. For a positive semidefinite matrix Q , the matrix $\sqrt{Q}$ is the unique positive semidefinite matrix R such that $R^2 = Q$.

A trajectory $(\mathbf{x}^j[t], \mathbf{u}^j[t])$ is *feasible* for the plant (1.3) if

$$\mathbf{a}(\mathbf{x}^j[t]) + B(\mathbf{x}^j[t])\mathbf{u}^j[t] = \mathbf{x}^j[t+1], \quad 1 \leq j \leq N, 1 \leq t < T. \quad (1.14)$$

1.4 Outline and Contributions

Section 2 of this thesis presents a summary of the literature on topics related to the current work, including model predictive control (MPC), system level synthesis (SLS), system identification (system ID), and inverse optimal control (IOC). Section 3 contains the key theoretical results of the current work. Section 4 describes some concrete demonstrations of the approach described in this work on two example systems, including the specific implementation details used for the numerical examples. Section 5 summarizes the thesis and provides several directions for possible future work.

The key contributions of this thesis are the demonstration of a method for optimal trajectory matching with explicit polynomial controllers, a proof that contractivity requirements for such a problem can be expressed as a set of matrix inequalities, and a proof that if certain symmetries are present in the plant and reference trajectories for the trajectory matching problem, there exists an optimal controller with those same symmetries.

Chapter 2

Literature Review

There exist various methods for solving problems similar to the one considered in this work: that is, identifying an explicit control law that makes the plant's state trajectories and control inputs from various initial points as close as possible to some given reference trajectories. One approach is model predictive control (MPC), which is good for solving the trajectory matching problem with nonlinear systems, and can handle constraints on the states and control inputs well, but does not generate explicit control laws. Another approach is system level synthesis (SLS), which generates explicit control laws based on desired disturbance-to-state and disturbance-to-control-input maps. However, extending SLS to nonlinear plants is difficult. A third approach is a creative application of system identification (system ID) techniques, where the system whose model is to be determined is the controller, and its input and output data are the reference state and control input trajectories, respectively. Still another approach is inverse optimal control (IOP), which takes one or more reference trajectories and finds parameters for the cost function for which the reference trajectories are local optima.

2.1 Model Predictive Control

One control design technique related to the trajectory matching problem is *model predictive control* (MPC), which finds the control input trajectory for a given plant that optimizes a given objective function. If the objective function is a measure of the mismatch between the reference and actual state and control input trajectories, as discussed above, then the MPC problem becomes a trajectory matching problem. MPC is an “online” algorithm: rather than generate an explicit control law ahead of time, at each timestep, it finds the

optimal sequence of control inputs for the next few timesteps, performs some or all of those control inputs, and repeats the cycle with a new optimization problem starting from the new plant state. MPC thus circumvents the difficulty of finding explicit solutions to the Hamilton–Jacobi–Bellman (HJB) equation, which describes the optimal control input to a plant in terms of its current state and the objective function, in all but the simplest cases. MPC lends itself particularly well to constrained control problems, since state and control input constraints can be imposed directly on the optimization problem solved by MPC [1]. For a trajectory matching problem with the plant in Equation (1.3), the MPC problem over T timesteps would take the form

$$\begin{aligned}
& \min_{\mathbf{x}[1:T], \mathbf{u}[1:T]} \sum_{t=1}^{T-1} (\|\mathbf{x}[t] - \mathbf{x}_{\text{ref}}[t]\|_{Q_t}^2 + \|\mathbf{u}[t] - \mathbf{u}_{\text{ref}}[t]\|_R^2) + \|\mathbf{x}[T] - \mathbf{x}_{\text{ref}}[T]\|_{Q_T}^2 \\
& \text{subject to} \\
& \mathbf{x}[0] = \mathbf{x}_0 \\
& \mathbf{a}(\mathbf{x}[t]) + B(\mathbf{x}[t])\mathbf{u}[t] = \mathbf{x}[t+1], \quad 1 \leq t < T \\
& \mathbf{x}[t] \in \mathbb{X}, \quad 1 \leq t < T \\
& \mathbf{u}[t] \in \mathbb{U}, \quad 1 \leq t < T \\
& \mathbf{x}[T] \in \mathbb{Z},
\end{aligned} \tag{2.1}$$

where $\mathbf{x}_0 \in \mathbb{R}^n$ is the initial state, $Q_t \in \mathbb{S}_n$ is the terminal cost matrix, $\mathbb{X} \subseteq \mathbb{R}^n$ and $\mathbb{U} \subseteq \mathbb{R}^m$ are the admissible state and control input sets, respectively, and $\mathbb{Z} \subseteq \mathbb{R}^n$ is the admissible terminal state set.

Drawbacks of MPC, however, include the relatively high computational demands of solving the optimization problem in real time. This is of particular concern for faster plants, where the optimization problem needs to be solved more frequently, and for higher-dimensional and nonlinear plants, where the optimization problems involved are more complex [1]. Additionally, guaranteeing closed-loop stability or robustness with nonlinear MPC can be challenging, and often comes at the cost of optimality [2] [3].

2.1.1 Explicit MPC

Explicit MPC is a variant of MPC where, instead of solving the MPC optimization problem online, the problem is solved *parametrically* in advance: the optimal solution is computed as a function of the current plant state, often using piecewise approximations [4]. A clear advantage of this method is that it provides an explicit control law, which is simpler to implement or to evaluate for stability and robustness. However, a downside of explicit MPC

is the great complexity of parametric representations of solutions to the MPC problem, which makes this method unsuitable for plants with high-dimensional state spaces [5].

2.2 System Level Synthesis

System level synthesis (SLS) is an approach to control design that works with explicit closed-loop disturbance-to-state and disturbance-to-control maps; its key is a parametrization of the space of such maps. For a linear plant with state feedback, *i.e.* $\mathbf{x}[t+1] = A\mathbf{x}[t] + B\mathbf{u}[t] + \delta[t]$, where $\delta[t] \in \mathbb{R}^n$ is the value of an exogenous *disturbance* at time t , suppose that the desired closed-loop transfer functions from $\delta[t]$ to $\mathbf{x}[t]$ and $\mathbf{u}[t]$ are $R(z)$ and $M(z)$, respectively. Then the *SLS equation* has the form

$$\begin{bmatrix} zI - A & -B \end{bmatrix} \begin{bmatrix} R(z) \\ M(z) \end{bmatrix} = I. \quad (2.2)$$

The closed-loop maps $R(z)$ and $M(z)$ are feasible — that is, there exists a controller that makes the closed-loop disturbance-to-state and disturbance-to-control transfer functions equal to $R(z)$ and $M(z)$, respectively — if and only if Equation 2.2 holds. In that case, the transfer function for the feedback controller that realizes these closed-loop maps is given by $K(z) = M(z)R(z)^{-1}$ [6].

SLS can be applied to the trajectory matching problem: the reference state and control signal trajectories can be converted to disturbance-to-state and disturbance-to-control models, respectively, possibly by using an impulse disturbance as the input. This conversion can be done using system identification, or a form of SLS that works with time-varying systems, such as that in [7], can be used.

Work has been done on extending SLS to nonlinear systems, as well as proving robustness and stability guarantees for controllers generated by SLS [7]. However, the method in [7] still relies on linear approximations of the closed-loop maps. Therefore, that method does not apply to more general nonlinear systems, such as the polynomial systems considered in this work.

2.3 System Identification

System identification (system ID) is the determination of a system’s dynamical equations from a collection of input and output data [8]. The problem studied in this work is an

interesting variation of the system ID problem where the system to be identified is not a plant, but a *controller*: the explicit control law that replicates the reference trajectories as well as possible. The input to this system is the reference plant state trajectory, and the output is the control input trajectory.

A major part of the nonlinear system ID problem is selecting the structure for the model [8]. In this thesis, this problem is simplified by considering only polynomial models. However, there are still two “hyperparameters”: the maximal degree of the polynomial functions in the model and the number of controller states. Optimal choices for these hyperparameters might not be obvious *a priori*, and their selection is an interesting problem. Section 3.1 discusses the topic of selecting the controller degree in further detail.

2.4 Inverse Optimal Control

Inverse optimal control (IOC) is an approach that can be applied to the trajectory matching problem. IOC takes several given trajectories, called *demonstrations*, and assumes they are locally optimal trajectories for a cost function with unknown parameters; it then determines the values of those parameters using the fact that the demonstrations are solutions to the Karush–Kuhn–Tucker (KKT) conditions, first-order conditions necessary and sufficient for local optimality under mild conditions, for the cost function [9] [10].

However, IOC does not actually generate control laws; it only generates objective functions, which can then be used by a separate optimal control problem to generate control laws. It is therefore a potentially helpful tool for the trajectory matching problem, but not a solution on its own.

Chapter 3

Theoretical Results

This section of the thesis presents several theoretical results, which are used to justify the methods used in this work. Section 3.1 presents a simple case of the problem studied in this work where only static controllers are considered. Section 3.2 shows that the contractivity requirements for the closed-loop system can be written as a set of matrix inequalities. Section 3.3 proves that certain symmetries present in the given plant and reference trajectories will also be present in optimal controllers, which can be used to reduce the dimension of the problem and design simpler controllers.

3.1 The Static Controller Case

Given the complexity of the problem in (1.7), it makes sense to begin with a simpler problem. The simplification of the problem discussed here is its restriction to static controllers: namely, those of the form $\mathbf{u}[t] = \mathbf{h}(\mathbf{x}[t])$, with no controller state $\mathbf{z}[t]$. With these restrictions, the problem in (1.7) is still nonconvex. However, a slightly different formulation of the problem can be used to obtain results:

$$\arg \min_{\mathbf{h}} \sum_{j=1}^N \sum_{t=0}^{T-1} \|\mathbf{a}(\mathbf{x}_{\text{ref}}^j[t]) + B\mathbf{h}(\mathbf{x}_{\text{ref}}^j[t]) - \mathbf{x}_{\text{ref}}^j[t+1]\|_Q^2. \quad (3.1)$$

One simplification applied in this problem is that B is now a constant, rather than depending on $\mathbf{x}$. The other simplification is that unlike the problem in (1.7), the above problem does not “iterate” the closed-loop map; that is, it compose it with itself repeatedly to generate the actual trajectories. Instead, it simply penalizes the mismatch between the value

of the closed-loop map applied at each point along each reference trajectory and the next point in that reference trajectory. These objective functions are not equivalent, but each is zero if and only if the other is zero, since either being zero implies that the controller matches the reference trajectories *perfectly*.

Despite the similarity between the problems in (1.7) and (3.1), the latter is a far “nicer” optimization problem: it is not only convex, but *quadratic*. This can be seen by expressing $\mathbf{h}(\mathbf{x}_{\text{ref}}^j[t])$ as the sum

$$\mathbf{h}_i(\mathbf{x}) = \sum_{\alpha \in I(n,k)} \mathbf{h}_{i,\alpha} \mathbf{x}^\alpha, \quad 1 \leq i \leq m. \quad (3.2)$$

This sum is linear in the coefficients of $\mathbf{h}$. The problem can be rewritten further in terms of a matrix $X^j \in \mathbb{R}^{(T-1) \times |I(n,k)|}$ and vector $\mathbf{b} \in \mathbb{R}^{n(T-1)}$ for each reference trajectory j :

$$X_{t\alpha}^j = \mathbf{x}^j[t]^\alpha, \quad (3.3)$$

and

$$\mathbf{b}_{(n-1)T+i}^j = \mathbf{x}_{\text{ref},i}^j[t+1] - \mathbf{a}_i(\mathbf{x}_{\text{ref}}^j[t]), \quad 1 \leq i \leq n, \quad 1 \leq j \leq N, \quad 0 \leq t < T. \quad (3.4)$$

Further, define X and $\mathbf{b}$ as the vertical concatenation of each X^j and $\mathbf{b}^j$, respectively. If the coefficients of $\mathbf{h}$ are gathered into an $m|I(n,k)|$ -dimensional vector, then the following simplification can be applied:

$$B\mathbf{h}(\mathbf{x}^j[t]) = (B \otimes X_t^j)\mathbf{h}. \quad (3.5)$$

With all these definitions, the problem in (3.1) can be written as

$$\arg \min_{\mathbf{h}} \|(\sqrt{Q} \otimes I_{N(T-1)})((B \otimes X)\mathbf{h} - \mathbf{b})\|_2^2, \quad (3.6)$$

and the quadratic nature of this problem becomes apparent.

Theorem 3.1. *The optimal solution for $\mathbf{h}$ to the problem in (3.1) is given by*

$$\mathbf{h} = ((B^T Q B)^{-1} B^T Q \otimes (X^T X)^{-1} X^T) \mathbf{b}. \quad (3.7)$$

Proof. As already shown, the problem can be rewritten as

$$\arg \min_{\mathbf{h}} \|(\sqrt{Q} \otimes I_{N(T-1)})((B \otimes X)\mathbf{h} - \mathbf{b})\|_2^2, \quad (3.8)$$

and further as

$$\begin{aligned}
& \arg \min_{\mathbf{h}} \|(\sqrt{Q} \otimes I_{N(T-1)})(B \otimes X)\mathbf{h} - (\sqrt{Q} \otimes I_{N(T-1)})\mathbf{b}\|_2^2 \\
&= \arg \min_{\mathbf{h}} \|(\sqrt{Q}B \otimes X)\mathbf{h} - (\sqrt{Q} \otimes I_{N(T-1)})\mathbf{b}\|_2^2 \\
&\quad (\text{by the mixed-product property of the Kronecker product}) \\
&= \left((\sqrt{Q}B \otimes X)^T (\sqrt{Q}B \otimes X) \right)^{-1} (\sqrt{Q}B \otimes X)^T (\sqrt{Q} \otimes I_{N(T-1)})\mathbf{b} \\
&= \left((B^T \sqrt{Q} \otimes X^T) (\sqrt{Q}B \otimes X) \right)^{-1} (B^T \sqrt{Q} \otimes X^T) (\sqrt{Q} \otimes I_{N(T-1)})\mathbf{b} \\
&\quad (\text{by the transpose property of the Kronecker product}) \\
&= (B^T QB \otimes X^T X)^{-1} (B^T Q \otimes X^T)\mathbf{b} \\
&= ((B^T QB)^{-1} \otimes (X^T X)^{-1}) (B^T Q \otimes X^T)\mathbf{b} \\
&\quad (\text{by the inverse property of the Kronecker product}) \\
&= ((B^T QB)^{-1} B^T Q \otimes (X^T X)^{-1} X^T) \mathbf{b}. \quad \square
\end{aligned}$$

The problem in (3.1) might not seem like a faithful analogue to that in (1.7), since the latter also penalizes differences between the reference and actual control signals. Is this a problem? No; the following theorem shows that penalizing the control signal mismatch does not increase the generality of the problem in (3.1).

Theorem 3.2. *Consider the following optimization problem:*

$$\arg \min_{\mathbf{h}} \sum_{t=0}^{T-1} \|\mathbf{a}(\mathbf{x}[t]) + B\mathbf{h}(\mathbf{x}[t]) - \mathbf{x}[t+1]\|_{Q'}^2 + \|\mathbf{h}(\mathbf{x}[t]) - \mathbf{u}[t]\|_R^2, \quad (3.9)$$

where $(\mathbf{x}[t], \mathbf{u}[t])$ is a given feasible trajectory. Suppose further that B has full column rank. Then the above optimization problem is equivalent to that in (3.1) with $Q = Q' + B(B^T B)^{-1}R(B^T B)^{-1}B^T$.

Proof. By assumption, $\mathbf{a}(\mathbf{x}[t]) + B\mathbf{u}[t] = \mathbf{x}[t+1]$ for all t . Since B has full column rank, there exist no other such $\mathbf{u}[t]$. Let $\tilde{Q} = B(B^T B)^{-1}R(B^T B)^{-1}B^T$, so that $B^T \tilde{Q} B = R$.

This way, the problem can be rewritten as

$$\begin{aligned}
& \arg \min_{\mathbf{h}} \sum_{t=0}^{T-1} \|\mathbf{a}(\mathbf{x}[t]) + B\mathbf{h}(\mathbf{x}[t]) - (\mathbf{a}(\mathbf{x}[t]) + B\mathbf{u}[t])\|_{Q'}^2 + \|\mathbf{h}(\mathbf{x}[t]) - \mathbf{u}[t]\|_R^2 \\
&= \arg \min_{\mathbf{h}} \sum_{t=0}^{T-1} \|\mathbf{a}(\mathbf{x}[t]) + B\mathbf{h}(\mathbf{x}[t]) - (\mathbf{a}(\mathbf{x}[t]) + B\mathbf{u}[t])\|_{Q'}^2 + \|B\mathbf{h}(\mathbf{x}[t]) - B\mathbf{u}[t]\|_{\tilde{Q}}^2 \\
&= \arg \min_{\mathbf{h}} \sum_{t=0}^{T-1} \|\mathbf{a}(\mathbf{x}[t]) + B\mathbf{h}(\mathbf{x}[t]) - (\mathbf{a}(\mathbf{x}[t]) + B\mathbf{u}[t])\|_{Q'+\tilde{Q}}^2 \\
&= \arg \min_{\mathbf{h}} \sum_{t=0}^{T-1} \|\mathbf{a}(\mathbf{x}[t]) + B\mathbf{h}(\mathbf{x}[t]) - \mathbf{x}[t+1]\|_{Q'+\tilde{Q}}^2. \quad \square
\end{aligned}$$

The assumption that B has full column rank is a very mild assumption: in fact, it can be guaranteed in practice by designing a controller that “ignores” redundant control inputs.

A natural further question is the effect of the degree of the controller on its optimal cost. It seems natural that increasing the controller degree decreases the mismatch between the desired and reference trajectories, and it certainly cannot *increase*, but how rapidly does it decrease, and until what point?

The following conjecture, if true, allows a stronger version of Theorem 3.4 to be proven. The matrix in question is a multivariable generalization of a Vandermonde matrix; if $n = 1$, this conjecture is already known to be true.

Conjecture 3.3. *Let $X \in \mathbb{R}^{(T-1) \times |I(n,k)|}$ be defined as*

$$X_{t,\alpha} = \mathbf{x}[t]^\alpha, \quad 0 \leq t < T, \quad \alpha \in I(n,k). \quad (3.10)$$

If the vectors $\mathbf{x}[t]$ are not all multiples of one another, then X has full rank.

Theorem 3.4. *For the optimization problem in (3.1), increasing the degree of the controller cannot increase the optimal value of the objective function. Moreover, if Conjecture 3.3 is true, the optimal cost of the optimization problem is*

$$\left\| \left((\sqrt{Q}B(B^TQB)^{-1}B^TQ - \sqrt{Q}) \otimes I_{N(T-1)} \right) \mathbf{b} \right\|_2, \quad (3.11)$$

which is achieved when $|I(n,k)| \geq N(T-1)$.

Proof. The optimal cost of the optimization problem in (3.1) is

$$\begin{aligned}
& \left\| (\sqrt{Q}B \otimes X) ((B^TQB)^{-1}B^TQ \otimes (X^TX)^{-1}X^T) \mathbf{b} - (\sqrt{Q} \otimes I_{N(T-1)}) \mathbf{b} \right\|_2^2 \\
&= \left\| \left(\sqrt{Q}B(B^TQB)^{-1}B^TQ \otimes X(X^TX)^{-1}X^T \right) \mathbf{b} - (\sqrt{Q} \otimes I_{N(T-1)}) \mathbf{b} \right\|_2^2 \\
&= \| (C - I) \mathbf{d} \|_2^2,
\end{aligned} \tag{3.12}$$

where $C = \sqrt{Q}B(B^TQB)^{-1}B^T\sqrt{Q} \otimes X(X^TX)^{-1}X^T$ and $\mathbf{d} = (\sqrt{Q} \otimes I_{T-1})\mathbf{b}$. Now, increasing the controller degree increases the number of columns of X without changing any columns already there, so the rank of X cannot decrease, and thus, neither can the rank of $X(X^TX)^{-1}X^T$. Further, the Kronecker product of two projection matrices is itself a projection matrix, since each of its eigenvalues is 0 or 1, and the eigenvalues of a Kronecker product of two matrices are each pairwise product of the eigenvalues of each respective matrix. Thus, increasing the rank of X necessarily increases the multiplicity of 1 as an eigenvalue of C , and therefore increases the dimension of $\text{Ker}(C - I)$. Further, if Conjecture 3.3 is true, increasing the controller degree increases the rank of X until it has full row rank, which is true when $|I(n, k)| \geq N(T - 1)$, at which point $X(X^TX)^{-1}X^T = I_{N(T-1)}$, and

$$\begin{aligned}
\| (C - I) \mathbf{d} \|_2^2 &= \left\| \left(\sqrt{Q}B(B^TQB)^{-1}B^TQ \otimes I_{N(T-1)} \right) \mathbf{b} - (\sqrt{Q} \otimes I_{N(T-1)}) \mathbf{b} \right\|_2^2 \\
&= \left\| \left((\sqrt{Q}B(B^TQB)^{-1}B^TQ - \sqrt{Q}) \otimes I_{N(T-1)} \right) \mathbf{b} \right\|_2^2. \quad \square
\end{aligned}$$

3.2 Stability Guarantees

A notion of stability for nonlinear systems is *contractivity*, which, informally, refers to a property of a function that sends points “closer together” with respect to some metric. Formally, if X is a metric space with metric d , a function $f : X \rightarrow X$ is γ -contracting on $Y \subseteq X$ for some $\gamma \in [0, 1)$ if

$$d(f(x_1), f(x_2)) \leq \gamma d(x_1, x_2) \quad \forall x_1, x_2 \in Y. \tag{3.13}$$

In this work, the metric space is typically the closed-loop state space equipped with the metric induced by a quadratic norm.

Contractivity of a function on a set implies exponential stability of that function on that set, which makes it a desirable property for a closed-loop state transition map. However,

contractivity on a set is often difficult to verify or enforce. In the context of the trajectory matching problem, a natural approach is instead to enforce contractivity at each point on each closed-loop trajectory: that is,

$$d(\mathbf{x}_{CL}^j[t+1], \hat{\mathbf{x}}_{CL}^j[t+1]) \leq \gamma d(\mathbf{x}_{CL}^j[t], \hat{\mathbf{x}}_{CL}^j[t]), \quad 0 \leq t < T, \quad 1 \leq j \leq N, \quad (3.14)$$

where $\hat{\mathbf{x}}_{CL}^j[t]$ is the closed-loop trajectory starting at an arbitrary point $\hat{\mathbf{x}}_{CL}^j[0]$ in a neighbourhood of $\mathbf{x}_{CL}^j[0]$. If d is the metric induced by a quadratic norm $\|\cdot\|_P^2$, then the contractivity requirement has the form

$$\|\hat{\mathbf{x}}_{CL}^j[t+1] - \mathbf{x}_{CL}^j[t+1]\|_P^2 \leq \gamma \|\hat{\mathbf{x}}_{CL}^j[t] - \mathbf{x}_{CL}^j[t]\|_P^2, \quad 0 \leq t < T, \quad 1 \leq j \leq N. \quad (3.15)$$

P is allowed to be time-varying to help the optimization problem converge; Theorem 3.5 does not require P to be the same at all timesteps. This requirement still cannot be directly imposed as a constraint for the optimization problem, since there are infinitely many possible values for $\hat{\mathbf{x}}_{CL}^j[0]$. However, the properties of the nonlinear closed-loop map can be related to the properties of its linearization in a neighbourhood of the point at which the map is linearized. This approach is used in Theorem 3.5, which shows that the following set of matrix inequalities is sufficient to guarantee contractivity:

$$J^{jT}[t]P[t]J^j[t] - P[t] \preceq -\gamma I, \quad 0 \leq t < T, \quad 1 \leq j \leq N, \quad (3.16)$$

where

$$J^j[t] := \frac{\partial \mathbf{x}_{CL}^j[t+1]}{\partial \mathbf{x}_{CL}^j[t]}. \quad (3.17)$$

The proof of Theorem 3.5 relies on two key insights: firstly, that the difference between the closed-loop map and its linearization can be bounded in a neighbourhood around the point at which the closed-loop map is linearized, and secondly, that the continuity of the closed-loop map ensures that the preimage of such a neighbourhood of each point in a trajectory under the closed-loop map is itself a neighbourhood of the previous point.

Theorem 3.5. *Let $\mathbf{k} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be continuously differentiable, and let $\mathbf{x}[0], \mathbf{x}[1], \dots, \mathbf{x}[T] \in \mathbb{R}^n$ be such that $\mathbf{x}[t+1] = \mathbf{k}(\mathbf{x}[t])$ for all t , $1 \leq t \leq T$. Let*

$$J[t] := \left. \frac{\partial \mathbf{k}}{\partial \mathbf{x}} \right|_{\mathbf{x}[t]}. \quad (3.18)$$

Suppose there exist $P[t] \in \mathbb{S}_n$, $0 \leq t < T$ and $\gamma \in (0, 1]$ such that $P[t] \succ 0$ and $J[t]^T P[t] J[t] - P[t] \preceq -\gamma I$ for all t . Then for any $\epsilon < \gamma$, there exists a neighbourhood N of $\mathbf{x}_0$ such that for all $\hat{\mathbf{x}}[0] \in N$ and for all t , $1 \leq t \leq T$,

$$\|\hat{\mathbf{x}}[t+1] - \mathbf{x}[t+1]\|_{P[t]}^2 - \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_{P[t]}^2 \leq -\epsilon \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2^2, \quad (3.19)$$

where $\hat{\mathbf{x}}_{t+1} = \mathbf{k}(\hat{\mathbf{x}}_t)$ for all t .

Proof. For all $\mathbf{x} \in \mathbb{R}^n$ and all $t, 1 \leq t \leq T$, we have $\mathbf{x}^T J[t]^T P[t] J[t] \mathbf{x} - \mathbf{x}^T P[t] \mathbf{x} \leq -\gamma \|\mathbf{x}[t]\|_2^2$. In particular,

$$\begin{aligned} (\hat{\mathbf{x}}[t]^T - \mathbf{x}[t]^T) J[t]^T P[t] J[t] (\hat{\mathbf{x}}[t] - \mathbf{x}[t]) - (\hat{\mathbf{x}}[t]^T - \mathbf{x}[t]^T) P[t] (\hat{\mathbf{x}}[t] - \mathbf{x}[t]) \\ \leq -\gamma \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2^2. \end{aligned} \quad (3.20)$$

However, since $\mathbf{k}$ is continuously differentiable, for every $\epsilon < \gamma$ and for every $t, 0 \leq t < T$, there exists a neighbourhood $N[t]$ of $\mathbf{x}[t]$ such that for all $\hat{\mathbf{x}}[t] \in N[t]$,

$$\|\mathbf{k}(\hat{\mathbf{x}}[t]) - \mathbf{k}(\mathbf{x}[t]) - J[t](\hat{\mathbf{x}}[t] - \mathbf{x}[t])\|_2 \leq \frac{1}{2} \lambda_{\max}(P[t])^{-1}(\gamma - \epsilon) \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2, \quad (3.21)$$

or equivalently, $\hat{\mathbf{x}}[t+1] - \mathbf{x}[t+1] = J[t](\hat{\mathbf{x}}[t] - \mathbf{x}[t]) + \mathbf{v}$, where $\|P[t]\mathbf{v}\|_2 \leq \frac{1}{2}(\gamma - \epsilon) \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2$. Substituting:

$$\begin{aligned} (\hat{\mathbf{x}}[t+1]^T - \mathbf{x}[t+1]^T - \mathbf{v}^T) P[t] (\hat{\mathbf{x}}[t+1] - \mathbf{x}[t+1] - \mathbf{v}) \\ - (\hat{\mathbf{x}}[t]^T - \mathbf{x}[t]^T) P[t] (\hat{\mathbf{x}}[t] - \mathbf{x}[t]) \leq -\gamma \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2^2 \\ (\hat{\mathbf{x}}[t+1]^T - \mathbf{x}[t+1]^T) P[t] (\hat{\mathbf{x}}[t+1] - \mathbf{x}[t+1]) \\ - 2\mathbf{v}^T P[t] (\hat{\mathbf{x}}[t] - \mathbf{x}[t]) + \|\mathbf{v}\|_{P[t]}^2 \\ - (\hat{\mathbf{x}}[t]^T - \mathbf{x}[t]^T) P[t] (\hat{\mathbf{x}}[t] - \mathbf{x}[t]) \leq -\gamma \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2^2 \\ \|\hat{\mathbf{x}}[t+1] - \mathbf{x}[t+1]\|_{P[t]}^2 - \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_{P[t]}^2 \leq -\gamma \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2^2 \\ + 2\mathbf{v}^T P[t] (\hat{\mathbf{x}}[t] - \mathbf{x}[t]). \end{aligned} \quad (3.22)$$

By the Cauchy-Schwarz Inequality:

$$\begin{aligned} \|\hat{\mathbf{x}}[t+1] - \mathbf{x}[t+1]\|_{P[t]}^2 - \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_{P[t]}^2 \\ \leq -\gamma \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2^2 + 2\|P[t]\mathbf{v}\|_2 \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2 \\ \leq -\gamma \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2^2 + 2 \left(\frac{1}{2}(\gamma - \epsilon) \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2 \right) \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2 \\ = -\epsilon \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2^2. \end{aligned} \quad (3.23)$$

Therefore, $\|\hat{\mathbf{x}}[t+1] - \mathbf{x}[t+1]\|_{P[t]}^2 - \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_{P[t]}^2 \leq -\epsilon \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2^2$ for any $\hat{\mathbf{x}}[t] \in N[t]$. Now, since $\mathbf{k}$ is continuous, the preimage of any $N[t+1]$ contains a neighbourhood $\mathbf{k}^{-1}(N[t+1])$ of $\mathbf{x}[t]$. Let $N^*[T-2] := N[T-2] \cap \mathbf{k}^{-1}(N[T-1])$, itself a neighbourhood of $\mathbf{x}[T-2]$. For any $\hat{\mathbf{x}}[T-2] \in N^*[T-2]$, $\|\hat{\mathbf{x}}[T-1] - \mathbf{x}[T-1]\|_{P[T-2]}^2 - \|\hat{\mathbf{x}}[T-2] - \mathbf{x}[T-2]\|_{P[T-2]}^2 \leq -\epsilon \|\hat{\mathbf{x}}[T-2] - \mathbf{x}[T-2]\|_2^2$, and since $\hat{\mathbf{x}}[T-1] \in N[T-1]$, the corresponding property also holds for $\hat{\mathbf{x}}[T-1]$. For $t < T-2$, let $N^*[t] := N[t] \cap \mathbf{k}^{-1}(N^*[t+1])$. By induction, for any $\hat{\mathbf{x}}[t] \in N^*[t]$, $\|\hat{\mathbf{x}}[t+1] - \mathbf{x}[t+1]\|_{P[t]}^2 - \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_{P[t]}^2 \leq -\epsilon \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2^2$ for all $t, 1 \leq i < T$. Setting $N = N^*[0]$ completes the proof. $\square$

3.2.1 Transverse Stability Guarantees

It may happen that a system cannot be made contractive in the sense of Equation (3.14). An example of a system for which this happens is Dubin's vehicle, which is discussed in more detail in Section 4.2.2. The Jacobian of Dubin's vehicle has an eigenvalue of 1 at each point in state-space, meaning that no control law can make the system satisfy Equation (3.14). This fixed eigenvalue comes from the vehicle's fixed speed.

A solution to this problem is to stabilize the system in only *some* directions: that is, to enforce

$$d(A^j[t]\mathbf{x}_{CL}^j[t+1], A^j[t]\hat{\mathbf{x}}_{CL}^j[t+1]) \leq \gamma d(A^j[t]\mathbf{x}_{CL}^j[t], A^j[t]\hat{\mathbf{x}}_{CL}^j[t]), \quad 0 \leq t < T, \quad 1 \leq j \leq N, \quad (3.24)$$

where each $A^j[t] \in \mathbb{R}^{n \times n_s}$ has full row rank for each j and t , and where n_s is the number of directions to be stabilized. The above equation guarantees that the closed-loop map sends the *images* of the points in question under $A^j[t]$ closer together. Effectively, this form of the contractivity requirement replaces the matrix norm $\mathbf{x} \mapsto \mathbf{x}^T P^j[t] \mathbf{x}$ with a matrix seminorm $\mathbf{x} \mapsto \mathbf{x}^T A^{jT}[t] P^j[t] A^j[t] \mathbf{x}$. For example, in the case of Dubin's vehicle, the eigenvector corresponding to the eigenvalue of 1 is the vehicle's current heading, so the rows of $A^j[t]$ could be a basis for the orthogonal complement of the span of the vehicle's heading at the point $\mathbf{x}^j[t]$.

Just as before, the constraints in Equation (3.24) are impossible to impose directly, since there are infinitely many possible values for each $\hat{\mathbf{x}}_{CL}^j[0]$. However, just as before, it is possible to guarantee that these constraints are satisfied by imposing finitely many matrix inequality constraints.

Theorem 3.6. *Let $\mathbf{k}$ and $J[t]$ be defined as in the statement of Theorem 3.5, and for a fixed $n_s < n$, let $A[t] \in \mathbb{R}^{n \times n_s}$ have full row rank for each t . Suppose there exist $P \in \mathbb{S}_n$ and $\gamma \in (0, 1]$ such that $P \succ 0$ and $J[t]^T A[t]^T P A[t] J[t] - P \preceq -\gamma I$ for all t . Then for any $\epsilon < \gamma$, there exists a neighbourhood N of $\mathbf{x}_0$ such that for all $\hat{\mathbf{x}}[0] \in N$ and for all t , $1 \leq t \leq T$,*

$$\|A[t]\hat{\mathbf{x}}[t+1] - A[t]\mathbf{x}[t+1]\|_P^2 - \|A[t]\hat{\mathbf{x}}[t] - A[t]\mathbf{x}[t]\|_P^2 \leq -\epsilon \|\hat{\mathbf{x}}[t] - \mathbf{x}[t]\|_2^2, \quad (3.25)$$

where $\hat{\mathbf{x}}_{t+1} = \mathbf{k}(\hat{\mathbf{x}}_t)$ for all t .

Proof. The proof is the same as that of Theorem 3.5, but replacing each occurrence of $P[t]$ with $A[t]^T P[t] A[t]$. $\square$

3.3 Symmetries of Optimal Controllers

The optimization problem in (1.7) is very high-dimensional, which is partially due to the large number of coefficients in high-order multivariate polynomials. For example, a third-degree polynomial in four variables has 35 coefficients, while a fifth-degree polynomial in ten variables has 3003. One way of reducing the number of decision variables in the optimization problem is to use the symmetries of the plant to design simpler controllers. For example, when the plant's $\mathbf{a}$ and B functions are odd and the set of reference trajectories includes the negative of each of its trajectories, it seems natural that the optimal controller should be odd, or that there should be *an* odd controller in the set of optimal controllers. Theorem 3.10 formalizes and generalizes this idea.

We describe the symmetries of dynamical systems in terms of the symmetry groups of their state and control signal spaces, which are matrix groups. The following definitions demonstrate this.

Definition 3.7. Let H be a subgroup of $\text{GL}_n(\mathbb{R}) \times \text{GL}_m(\mathbb{R})$. The plant in (1.3) is defined to be *symmetric under H* if for each $(N, M) \in H$,

$$\mathbf{a}(N\mathbf{x}) + B(N\mathbf{x})M\mathbf{u} = N(\mathbf{a}(\mathbf{x}) + B(\mathbf{x})\mathbf{u}). \quad (3.26)$$

In other words, the plant is symmetric under H if the diagram in Figure 3.1 commutes for each $(N, M) \in H$.

$$\begin{array}{ccccc} \mathbb{R}^n \times \mathbb{R}^m & \xrightarrow{B(\mathbf{x})\mathbf{u}} & \mathbb{R}^n & \xrightarrow{\mathbf{a}} & \mathbb{R}^n \\ \downarrow (N, M) & & \downarrow N & & \downarrow N \\ \mathbb{R}^n \times \mathbb{R}^m & \xrightarrow{B(\mathbf{x})\mathbf{u}} & \mathbb{R}^n & \xrightarrow{\mathbf{a}} & \mathbb{R}^n \end{array}$$

Figure 3.1: The commutative diagram for Definition 3.7.

For example, if the plant is “odd”, in the sense that $\mathbf{a}(\mathbf{x})$ is an odd function and $B(\mathbf{x})$ is an even function, H is the group $\{(I_n, I_m), (-I_n, -I_m)\}$.

Definition 3.8. Let H be a subgroup of $\text{GL}_n(\mathbb{R}) \times \text{GL}_m(\mathbb{R})$. Writing each element of H as (N, M) , define its left action on a set of reference trajectories as

$$(N, M) \cdot (\mathbf{x}_{\text{ref}}^j[t], \mathbf{u}_{\text{ref}}^j[t]) = (N\mathbf{x}_{\text{ref}}^j[t], M\mathbf{u}_{\text{ref}}^j[t]), \quad 0 \leq t \leq T. \quad (3.27)$$

The set of reference trajectories is defined to be *closed under H* if this group action is well-defined: that is, if for every j , $1 \leq j \leq N$ and for every $(N, M) \in H$, there exists a j' , $1 \leq j' \leq N$ such that $\mathbf{x}_{\text{ref}}^j[t] = \mathbf{x}_{\text{ref}}^{j'}[t]$ and $\mathbf{u}_{\text{ref}}^j[t] = \mathbf{u}_{\text{ref}}^{j'}[t]$ for all t .

A natural informal question to ask is whether or not, given a set of points with certain symmetries, it is possible to fit a polynomial to interpolate those points that has the same symmetries. The following lemma formalizes this question and answers it in the positive.

Lemma 3.9. *Let H be a finite subgroup of $\text{GL}_n(\mathbb{R}) \times \text{GL}_m(\mathbb{R})$, and suppose that there exists a set $(\mathbf{x}_i, \mathbf{y}_i) \subset \mathbb{R}^n \times \mathbb{R}^m$ of size A such that no two $\mathbf{x}_i$ are equal, and for every $(N, M) \in H$ and for every i , $1 \leq i \leq A$, there exists a j , $1 \leq j \leq A$ such that $N\mathbf{x}_i = \mathbf{x}_j$ and $M\mathbf{y}_i = \mathbf{y}_j$. Then there exists a polynomial function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ such that for every i , $1 \leq i \leq A$, $\mathbf{f}(\mathbf{x}_i) = \mathbf{y}_i$, and for every $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{f}(N\mathbf{x}) = M\mathbf{f}(\mathbf{x})$.*

Proof. Since no two $\mathbf{x}_i$ are equal, there exists a polynomial function $\hat{\mathbf{f}} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ such that for every i , $1 \leq i \leq A$, $\hat{\mathbf{f}}(\mathbf{x}_i) = \mathbf{y}_i$ [11]. Define $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ as

$$\mathbf{f}(\mathbf{x}) = \frac{1}{|H|} \sum_{(N,M) \in H} M^{-1} \hat{\mathbf{f}}(N\mathbf{x}). \quad (3.28)$$

To show that $\mathbf{f}(N'\mathbf{x}) = M'\mathbf{f}(\mathbf{x})$ for every $(N', M') \in H$, write

$$\begin{aligned} \mathbf{f}(N'\mathbf{x}) &= \frac{1}{|H|} \sum_{(N,M) \in H} M^{-1} \hat{\mathbf{f}}(NN'\mathbf{x}) \\ &= \frac{1}{|H|} \sum_{(N,M) \in H} M' M'^{-1} M^{-1} \hat{\mathbf{f}}(NN'N'^{-1}N'\mathbf{x}) \\ &= M' \frac{1}{|H|} \sum_{(N,M) \in H} M'^{-1} M^{-1} \hat{\mathbf{f}}(NN'\mathbf{x}) \\ &= M' \frac{1}{|H|} \sum_{(N,M) \in H} M^{-1} \hat{\mathbf{f}}(N\mathbf{x}) \\ &= M' \mathbf{f}(\mathbf{x}). \end{aligned} \quad (3.29)$$

The above is justified since $\{(NN', MM') \mid (N, M) \in H\} = H$ for any $(N', M') \in H$.

To show that $\mathbf{f}(\mathbf{x}_i) = \mathbf{y}_i$ for every i , $1 \leq i \leq A$, write

$$\begin{aligned}
\mathbf{f}(\mathbf{x}_i) &= \frac{1}{|H|} \sum_{(N,M) \in H} M^{-1} \hat{\mathbf{f}}(N\mathbf{x}_i) \\
&= \frac{1}{|H|} \sum_{(N,M) \in H} M^{-1} \hat{\mathbf{f}}(\mathbf{x}_{j(N,M,i)}) \\
&= \frac{1}{|H|} \sum_{(N,M) \in H} M^{-1} \mathbf{y}_{j(N,M,i)} \\
&= \frac{1}{|H|} \sum_{(N,M) \in H} \mathbf{y}_i \\
&= \mathbf{y}_i,
\end{aligned} \tag{3.30}$$

where $j(N, M, i)$ is such that $N\mathbf{x}_i = \mathbf{x}_{j(N,M,i)}$ and $M\mathbf{y}_i = \mathbf{y}_{j(N,M,i)}$; such a $j(N, M, i)$ is guaranteed to exist by the assumptions. $\square$

The conditions of Lemma 3.9 effectively require the set $(\mathbf{x}_i, \mathbf{y}_i)$ to be “closed” under H , and informally, the lemma shows that the points $(\mathbf{x}_i, \mathbf{y}_i)$ can be interpolated by a polynomial function “symmetric” under H . The result of Lemma 3.9 is used in the proof of Theorem 3.10.

Theorem 3.10, the main result of this section, states that if the plant and the set of reference trajectories are symmetric under a group H , as defined above, and if there is any globally optimal controller at all for the trajectory matching problem, then there exists an optimal controller that is also symmetric under H in a similar way. The proof of 3.10 shows how any optimal controller can be turned into one symmetric under H . The key insight of the proof is dividing the set of reference trajectories into its orbits under the group action by H and picking the “best” reference trajectory from each orbit, that is, the reference trajectory best matched by the corresponding actual trajectory. We can then gather the set of points of each chosen trajectory and “stitch” them together through polynomial interpolation in a symmetric way using Lemma 3.9; this interpolated controller can be shown to do no worse than the original controller.

Theorem 3.10. *Let H be a finite subgroup of $\text{GL}_n(\mathbb{R}) \times \text{GL}_m(\mathbb{R})$, and let $\phi : H \rightarrow \text{GL}_{n_c}(\mathbb{R})$ be a group homomorphism. For the problem in Equation (1.7), suppose that the plant is symmetric under H and the set of reference trajectories is closed under H . Suppose further that $N^T Q N = Q$ and $M^T R M = R$ for every $(N, M) \in H$. Additionally, suppose that each point along each trajectory is distinct: that is, if there exist t, t' and j, j' such that*

$$\begin{array}{ccccc}
\mathbb{R}^{n_c} & \xleftarrow{\mathbf{f}} & \mathbb{R}^{n_c} \times \mathbb{R}^n & \xrightarrow{\mathbf{h}} & \mathbb{R}^m \\
\downarrow \phi(N,M) & & \downarrow (\phi(N,M), N) & & \downarrow M \\
\mathbb{R}^n & \xleftarrow{\mathbf{f}} & \mathbb{R}^{n_c} \times \mathbb{R}^n & \xrightarrow{\mathbf{h}} & \mathbb{R}^m
\end{array}$$

Figure 3.2: The commutative diagram for Theorem 3.10

$\mathbf{x}_{ref}^j[t] = \mathbf{x}_{ref}^{j'}[t']$, then $t = t'$ and $j = j'$. Then if any optimal solution to (1.7) exists, there exists at least one optimal solution to (1.7) that satisfies the following conditions for every $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^{n_c}$, and $(N, M) \in H$:

$$\begin{aligned}
\mathbf{f}(\phi(N, M)\mathbf{z}, N\mathbf{x}) &= \phi(N, M)\mathbf{f}(\mathbf{z}, \mathbf{x}) \\
\mathbf{h}(\phi(N, M)\mathbf{z}, N\mathbf{x}) &= M\mathbf{h}(\mathbf{z}, \mathbf{x}),
\end{aligned} \tag{3.31}$$

or, in other words, the diagram in Figure 3.2 commutes for every $(N, M) \in H$.

Proof. Let $(\hat{\mathbf{f}}, \hat{\mathbf{h}})$ be an optimal solution to (1.7), and let $\hat{\mathbf{x}}^j[t]$, $\hat{\mathbf{u}}^j[t]$, and $\hat{\mathbf{z}}^j[t]$ be the plant state, control signal, and controller state trajectories realized by that controller from initial condition j . Partition the set of reference trajectories into its orbits under the action of H ; let there be A such orbits, and denote them as $O_1, O_2, \dots, O_A$. From each orbit O_i , $1 \leq i \leq A$, select the reference trajectory j_i that minimizes

$$\sum_{t=0}^{T-1} \|\mathbf{x}^{j_i}[t] - \mathbf{x}_{ref}^{j_i}[t]\|_Q^2 + \|\mathbf{u}^{j_i}[t] - \mathbf{u}_{ref}^{j_i}[t]\|_R^2. \tag{3.32}$$

Let $\mathbf{f}^* : \mathbb{R}^{n_c} \times \mathbb{R}^n \rightarrow \mathbb{R}^{n_c}$ be a polynomial function that satisfies (3.31) such that

$$\mathbf{f}^*(\phi(N, M)\hat{\mathbf{z}}^{j_i}[t], N\hat{\mathbf{x}}^{j_i}[t]) = \phi(N, M)\hat{\mathbf{z}}^{j_i}[t+1], \quad (N, M) \in H, \quad 1 \leq i \leq A, \quad 0 \leq t < T, \tag{3.33}$$

and let $\mathbf{h}^* : \mathbb{R}^{n_c} \times \mathbb{R}^n \rightarrow \mathbb{R}^m$ be a polynomial function that satisfies (3.31) such that

$$\mathbf{h}^*(\phi(N, M)\hat{\mathbf{z}}^{j_i}[t], N\hat{\mathbf{x}}^{j_i}[t]) = M\hat{\mathbf{u}}^{j_i}[t], \quad (N, M) \in H, \quad 1 \leq i \leq A, \quad 0 \leq t < T. \tag{3.34}$$

Since each point along each trajectory is distinct, $\mathbf{f}^*$ and $\mathbf{h}^*$ are well-defined by Lemma 3.9: for $\mathbf{f}^*$, the role of H in Lemma 3.9 is played by $\phi(H) \times \{N\} \times \phi(H)$, and for $\mathbf{h}^*$, it is played by $\phi(H) \times \{N\} \times \{M\}$, where $\{N\} := \{N \mid (N, M) \in H\}$ and $\{M\} := \{M \mid (N, M) \in H\}$.

Let $\mathbf{x}^{*j}[t]$, $\mathbf{u}^{*j}[t]$, and $\mathbf{z}^{*j}[t]$ be the plant state, control signal, and controller state trajectories realized by $(\mathbf{f}^*, \mathbf{h}^*)$ from initial condition j , respectively. We now show that

$(\mathbf{f}^*, \mathbf{h}^*, \mathbf{x}^{*j}[t], \mathbf{z}^{*j}[t])$ is an optimal solution to (1.7); to do so, we first prove the following claim.

Claim: For every i , $1 \leq i \leq A$, for every t , $0 \leq t < T$, and for every $(N, M) \in H$, there exists j' , $1 \leq j' \leq N$ such that

$$\begin{aligned}\phi(N, M)\mathbf{z}^{*j_i}[t] &= \mathbf{z}^{*j'}[t] \\ N\mathbf{x}^{*j_i}[t] &= \mathbf{x}^{*j'}[t].\end{aligned}\tag{3.35}$$

The claim is shown by induction on t ; the base case follows directly from the fact that the set of $(\mathbf{x}^j[0], \mathbf{u}^j[0])$ is closed under H and that $\mathbf{z}^j[0] = \mathbf{0} \forall j$. To show the inductive step, write

$$\begin{aligned}\phi(N, M)\mathbf{z}^{*j_i}[t+1] &= \phi(N, M)\mathbf{f}^*(\mathbf{z}^{*j_i}[t], \mathbf{x}^{*j_i}[t]) \\ &= \mathbf{f}^*(\phi(N, M)\mathbf{z}^{*j_i}[t], N\mathbf{x}^{*j_i}[t]) \\ &= \mathbf{f}^*(\mathbf{z}^{*j'}[t], \mathbf{x}^{*j'}[t]) \\ &= \mathbf{z}^{*j'}[t+1],\end{aligned}\tag{3.36}$$

and

$$\begin{aligned}N\mathbf{x}^{*j_i}[t+1] &= N(\mathbf{a}(\mathbf{x}^{*j_i}[t]) + B(\mathbf{x}^{*j_i}[t])\mathbf{h}^*(\mathbf{z}^{*j_i}[t], \mathbf{x}^{*j_i}[t])) \\ &= \mathbf{a}(N\mathbf{x}^{*j_i}[t]) + B(N\mathbf{x}^{*j_i}[t])M\mathbf{h}^*(\mathbf{z}^{*j_i}[t], \mathbf{x}^{*j_i}[t]) \\ &= \mathbf{a}(N\mathbf{x}^{*j_i}[t]) + B(N\mathbf{x}^{*j_i}[t])\mathbf{h}^*(\phi(N, M)\mathbf{z}^{*j_i}[t], N\mathbf{x}^{*j_i}[t]) \\ &= \mathbf{a}(\mathbf{x}^{*j'}[t]) + B(\mathbf{x}^{*j'}[t])\mathbf{h}^*(\mathbf{z}^{*j'}[t], \mathbf{x}^{*j'}[t]) \\ &= \mathbf{x}^{*j'}[t+1],\end{aligned}\tag{3.37}$$

completing the proof of the claim.

To show that $(\mathbf{f}^*, \mathbf{h}^*, \mathbf{x}^*, \mathbf{z}^*)$ is an optimal solution, observe that the value of the objec-

tive function in (1.7) associated with $\mathbf{f}^*$ and $\mathbf{h}^*$ is

$$\begin{aligned}
& \sum_{j=1}^N \sum_{t=0}^{T-1} \|\mathbf{x}^{*j}[t] - \mathbf{x}_{\text{ref}}^j[t]\|_Q^2 + \|\mathbf{u}^{*j}[t] - \mathbf{u}_{\text{ref}}^j[t]\|_R^2 \\
&= \sum_{i=1}^A \sum_{j \in O_i} \sum_{t=0}^{T-1} \|\mathbf{x}^{*j}[t] - \mathbf{x}_{\text{ref}}^j[t]\|_Q^2 + \|\mathbf{u}^{*j}[t] - \mathbf{u}_{\text{ref}}^j[t]\|_R^2 \\
&\quad (\text{since the orbits } O_i \text{ partition the set of reference trajectories}) \\
&= \sum_{i=1}^A \sum_{(N,M) \in H} \sum_{t=0}^{T-1} \|N\mathbf{x}^{*j_i}[t] - N\mathbf{x}_{\text{ref}}^{j_i}[t]\|_Q^2 + \|M\mathbf{u}^{*j_i}[t] - M\mathbf{u}_{\text{ref}}^{j_i}[t]\|_R^2 \\
&\quad (\text{using the claim}) \\
&= \sum_{i=1}^A \sum_{(N,M) \in H} \sum_{t=0}^{T-1} \|\mathbf{x}^{*j_i}[t] - \mathbf{x}_{\text{ref}}^{j_i}[t]\|_Q^2 + \|\mathbf{u}^{*j_i}[t] - \mathbf{u}_{\text{ref}}^{j_i}[t]\|_R^2 \tag{3.38} \\
&\quad (\text{since } N^T Q N = Q \text{ and } M^T R M = R \text{ for every } (N, M) \in H) \\
&= \sum_{i=1}^A \sum_{j \in O_i} \sum_{t=0}^{T-1} \|\hat{\mathbf{x}}^{j_i}[t] - \mathbf{x}_{\text{ref}}^{j_i}[t]\|_Q^2 + \|\hat{\mathbf{u}}^{j_i}[t] - \mathbf{u}_{\text{ref}}^{j_i}[t]\|_R^2 \\
&\quad (\text{since } \mathbf{f}^* \text{ was chosen to “agree” with } \hat{\mathbf{f}} \text{ on each } j_i, \text{ and } \mathbf{h}^* \text{ with } \hat{\mathbf{h}}) \\
&\leq \sum_{i=1}^A \sum_{j \in O_i} \sum_{t=0}^{T-1} \|\hat{\mathbf{x}}^j[t] - \mathbf{x}_{\text{ref}}^j[t]\|_Q^2 + \|\hat{\mathbf{u}}^j[t] - \mathbf{u}_{\text{ref}}^j[t]\|_R^2 \\
&\quad (\text{since the } j_i \text{ were selected from each } O_i \text{ to minimize the above}) \\
&= \sum_{j=1}^N \sum_{t=0}^{T-1} \|\hat{\mathbf{x}}^j[t] - \mathbf{x}_{\text{ref}}^j[t]\|_Q^2 + \|\hat{\mathbf{u}}^j[t] - \mathbf{u}_{\text{ref}}^j[t]\|_R^2,
\end{aligned}$$

which is the value of the objective function associated with $\hat{\mathbf{f}}$ and $\hat{\mathbf{h}}$. However, $(\hat{\mathbf{f}}, \hat{\mathbf{h}}, \hat{\mathbf{x}}, \hat{\mathbf{z}})$ is an optimal solution. Therefore, the above inequality must be an equality, and $(\mathbf{f}^*, \mathbf{h}^*, \mathbf{x}^*, \mathbf{z}^*)$ is an optimal solution as well. $\square$

Remark. The assumptions that $N^T Q N = Q$ and $M^T R M = R$ are easy to satisfy. For example, in the case where N and M are $-I_n$ and $-I_m$, respectively, the assumptions are true for any Q and R .

Chapter 4

Numerical Results

4.1 Solution Methods

The optimization problem in Equation (1.7) is very nonlinear and nonconvex, even without the stability constraints; the nonconvexity comes from the products and powers of $\mathbf{x}^j[t]$ and $\mathbf{z}^j[t]$ in the constraints. Due to this nonconvexity, all optimization problems were solved using MATLAB's `fmincon` function, which is sufficiently general to express the optimization problems in this work and offers multiple optimization algorithms. Since `fmincon` cannot work with matrix inequalities directly, positive semidefiniteness constraints were first rewritten as positive definiteness constraints and then converted to simple inequalities using Sylvester's criterion, which states that a matrix is positive definite if and only if each of its leading principal minors is positive.

The optimization problem was observed to be very sensitive to initial conditions; when the initial point was infeasible, the optimization almost universally failed to converge to a feasible point. To overcome this problem and find a good initial condition for the optimization problem, a “secondary” optimization problem was first solved, and its optimal value was converted into an initial point for the full “primary” optimization problem. This secondary optimization problem has the following form:

$$\begin{aligned} \min_{F_x, F_z, H_x, H_z, \hat{\mathbf{z}}[t]} & \sum_{j=1}^N \sum_{t=1}^T \|H_x \mathbf{x}_{\text{ref}}^j[t] + H_z \hat{\mathbf{z}}^j[t] - \mathbf{u}_{\text{ref}}^j[t]\|_2^2 \\ & \text{subject to} \\ & F_x \mathbf{x}_{\text{ref}}^j[t] + F_z \hat{\mathbf{z}}^j[t] - \hat{\mathbf{z}}^j[t+1] = 0, \quad 1 \leq j \leq N, \quad 0 \leq t < T. \end{aligned} \tag{4.1}$$

This problem is similar to that in Equation (3.1), but with a linear dynamic controller instead of a nonlinear static one: in fact, $F_x \mathbf{x} + F_z \mathbf{z}$ can be viewed as a linearization of $\mathbf{f}(\mathbf{x}, \mathbf{z})$, and likewise for $H_x \mathbf{x} + H_z \mathbf{z}$.

Once a solution to this problem was found, it was converted to an initial condition for the primary optimization problem by setting $\mathbf{f}(\mathbf{z}, \mathbf{x}) = F_z \mathbf{z} + F_x \mathbf{x}$ and $\mathbf{h}(\mathbf{z}, \mathbf{x}) = H_z \mathbf{z} + H_x \mathbf{x}$. The initial conditions for $\mathbf{x}^j[t]$ and $\mathbf{z}^j[t]$ were found by computing the closed-loop trajectory achieved by this controller from the initial point of each given trajectory, and the initial $P^j[t]$ were found by solving the discrete Lyapunov equation at each point of each trajectory with the Jacobian of the closed-loop dynamic function at that point.

4.2 Examples

The approach described here was demonstrated on several example plants. In each case, the continuous-time equations were first discretized using the forward Euler approach, and then a four-term (cubic) Taylor series approximation was used to represent each plant as a polynomial model.

4.2.1 Single-Machine Infinite-Bus Power System

The first example plant on which this method was demonstrated was the single-machine infinite-bus (SMIB) power system, which models a single synchronous generator connected to an infinite bus through a lossless transmission line. With a simplifying choice of coordinates, the SMIB system can be described by the following differential equations:

$$\begin{aligned}\dot{\mathbf{x}}_1 &= \mathbf{x}_2 \\ \dot{\mathbf{x}}_2 &= -\sin \mathbf{x}_1 - c\mathbf{x}_2 + u,\end{aligned}\tag{4.2}$$

where $\mathbf{x} \in \mathbb{R}^2$ is the state vector, $c \geq 0$ is a damping constant, and u is a term related to the mechanical power supplied by the generator [12]. These equations are identical to those describing a damped driven pendulum. This is a control-affine system which, after replacing the sine term with a Taylor series approximation, is suitable for the approach described here. The polynomial approximation used for the SMIB system's dynamics is

$$\begin{aligned}\dot{\mathbf{x}}_1 &= \mathbf{x}_2 \\ \dot{\mathbf{x}}_2 &= -\mathbf{x}_1 + \frac{1}{6}\mathbf{x}_1^3 - c\mathbf{x}_2 + u.\end{aligned}\tag{4.3}$$

A simple first test for this approach was to show that it could match trajectories that approach the unstable equilibrium of the SMIB system while stabilizing this unstable equilibrium. To accomplish this, MPC, using MATLAB’s `nlmpc` class, was used to generate several trajectories that approach the unstable equilibrium; the cost function supplied to MPC was a simple quadratic cost. These trajectories were then used as the reference trajectories for the trajectory matching problem. The transition function for both the original SMIB dynamics and their polynomial approximation are odd functions, so we can apply Theorem 3.10 and consider only odd controllers. In this case, H is simply $\{(I_n, I_m), (-I_n, -I_m)\}$, and ϕ sends the only non-identity element of H to $-I_{nc}$. The reason that we do not need the “inverses” of each trajectory — that is, the reason that the set of reference trajectories does not need to be closed under H — is that the plant and controller symmetries will ensure that the actual trajectories for the inverses of the initial conditions are simply the inverses of the original actual trajectories.

The approach described in this thesis achieved excellent results on the SMIB system. The performance of the nonlinear controller found by the new approach was evaluated by comparing it against the optimal linear controller found by the secondary optimization problem in Equation (4.1). The nonlinear controller found by the optimization achieved a mismatch between the reference and actual trajectories approximately 11.6 times smaller than of the linear controller, and the nonlinear controller satisfied the contractivity guarantees at each point along each trajectory. Figure 4.1 shows the squared errors between the reference and actual trajectories over time for the linear and nonlinear controllers.

4.2.2 Dubin’s Vehicle

Another class of systems suitable for this approach is vehicle systems, which frequently have very nonlinear dynamics. Contractivity guarantees are also very relevant to vehicles: for instance, consider a mobile robot placed in a starting area by a human operator and tasked to follow a predetermined path. The operator might not place the robot in exactly the same location each time, so a contractivity guarantee ensures that even if the robot starts a small distance from the starting location, its trajectory will still converge to the reference path.

A simple representative of the class of vehicle systems is Dubin’s vehicle, which is described by the following differential equations:

$$\begin{aligned}\dot{\mathbf{x}}_1 &= v \cos \mathbf{x}_3 \\ \dot{\mathbf{x}}_2 &= v \sin \mathbf{x}_3 \\ \dot{\mathbf{x}}_3 &= u,\end{aligned}\tag{4.4}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are the vehicle's position in two-dimensional space, $\mathbf{x}_3$ is its heading, $v > 0$ is its speed, and u is the steering input [13]. Dubin's vehicle is effectively a unicycle with a fixed speed.

A problem arises when applying the contractivity constraints in Equation (3.16): the Jacobian matrix for Dubin's vehicle has an eigenvalue of 1 for all values of $\mathbf{x}$, which means that constraints of the form in (3.16) are impossible to satisfy. To avoid this, stability was only enforced in the *transverse* directions: that is, those orthogonal to the vehicle's current heading. This works because the fixed eigenvalue of 1 arises from the vehicle's fixed speed, and its eigenvector is simply the heading of Dubin's vehicle, $[v \cos \mathbf{x}_3 \ v \sin \mathbf{x}_3 \ u]^T$. A basis for the orthogonal complement of the corresponding eigenspace of the Jacobian can be gathered into the following matrix:

$$A = \begin{bmatrix} -v \sin \mathbf{x}_3 & u \cos \mathbf{x}_3 \\ v \cos \mathbf{x}_3 & u \sin \mathbf{x}_3 \\ 0 & -v \end{bmatrix}. \quad (4.5)$$

The stability constraints used for Dubin's vehicle then take the modified form:

$$J^j{}^T[t] \begin{bmatrix} A[t] & 0 \\ 0 & I_{n_c} \end{bmatrix} P \begin{bmatrix} A[t]^T & 0 \\ 0 & I_{n_c} \end{bmatrix} J^j[t] - P \preceq -\gamma I, \ 1 \leq j \leq N, \ 1 \leq t < T. \quad (4.6)$$

Effectively, these constraints ensure that the closed-loop state transition function is contracting in every direction except that parallel to the vehicle's heading.

The test for Dubin's vehicle was a lane change problem, which is directly motivated by real-world autonomous vehicle applications. MPC, again using MATLAB's `nlmpc` class, was used to generate two reference trajectories starting from different initial conditions. The objective function given to MPC was to maintain a Y coordinate of 0 units for the first half of the timesteps, and a Y coordinate of 0.005 units for the second half. This represents an autonomous vehicle performing a lane change manoeuvre. The approach described in this thesis also achieved excellent results on Dubin's vehicle, reducing the mismatch between the reference and actual trajectories from the linear to the nonlinear controller by a factor of roughly 130. Figure 4.2 shows the trajectories generated by the initial linear controller and the full nonlinear controller for each starting point, as well as the reference trajectories. Figure 4.3 shows the control inputs generated by each controller, as well as the reference control input trajectory. Figure 4.4 shows the squared error between the reference and actual positions over time for the linear and nonlinear controllers for each trajectory. In each case, the comparison is the linear controller found by the secondary optimization problem in Equation (4.1).

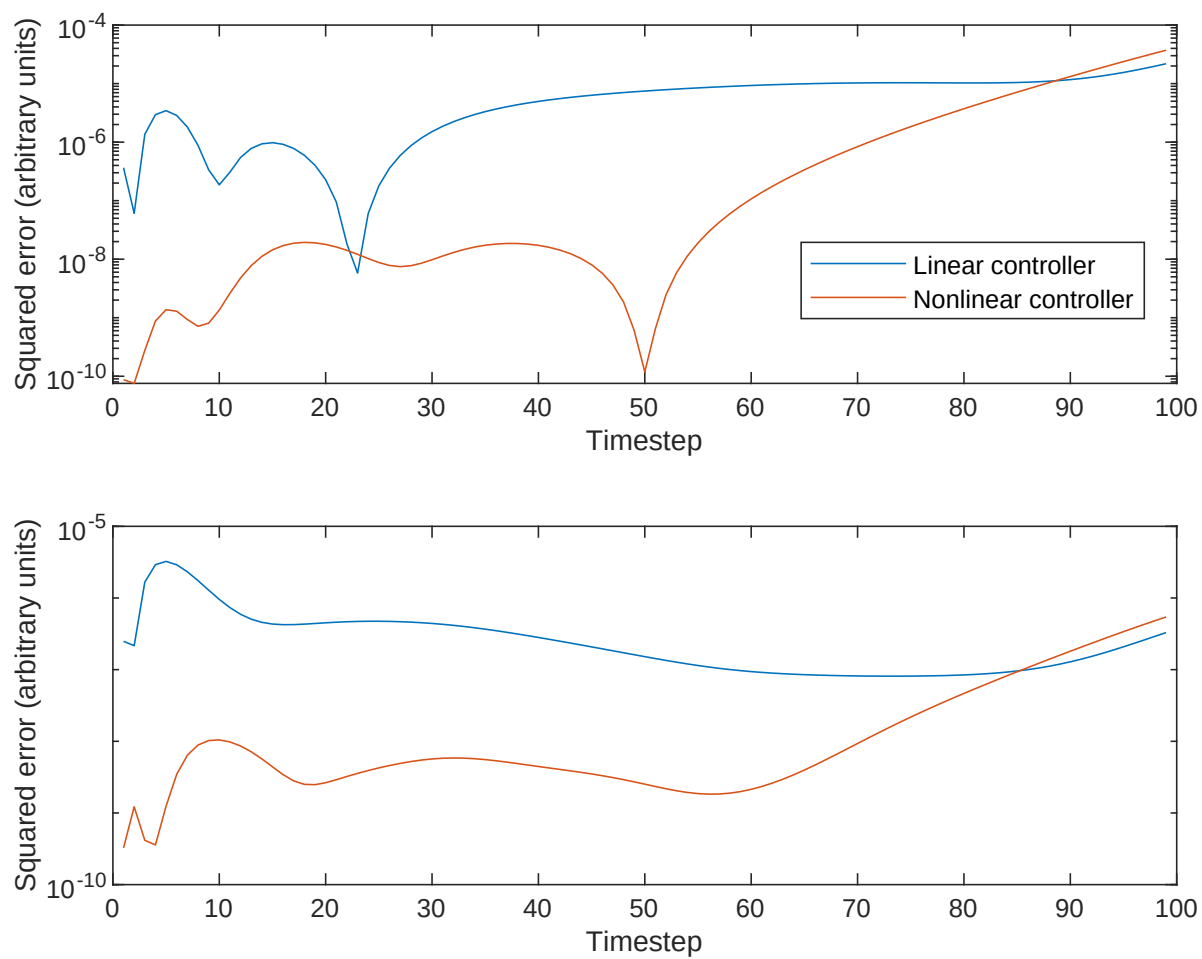


Figure 4.1: The squared errors over time for each reference trajectory on the SMIB system.

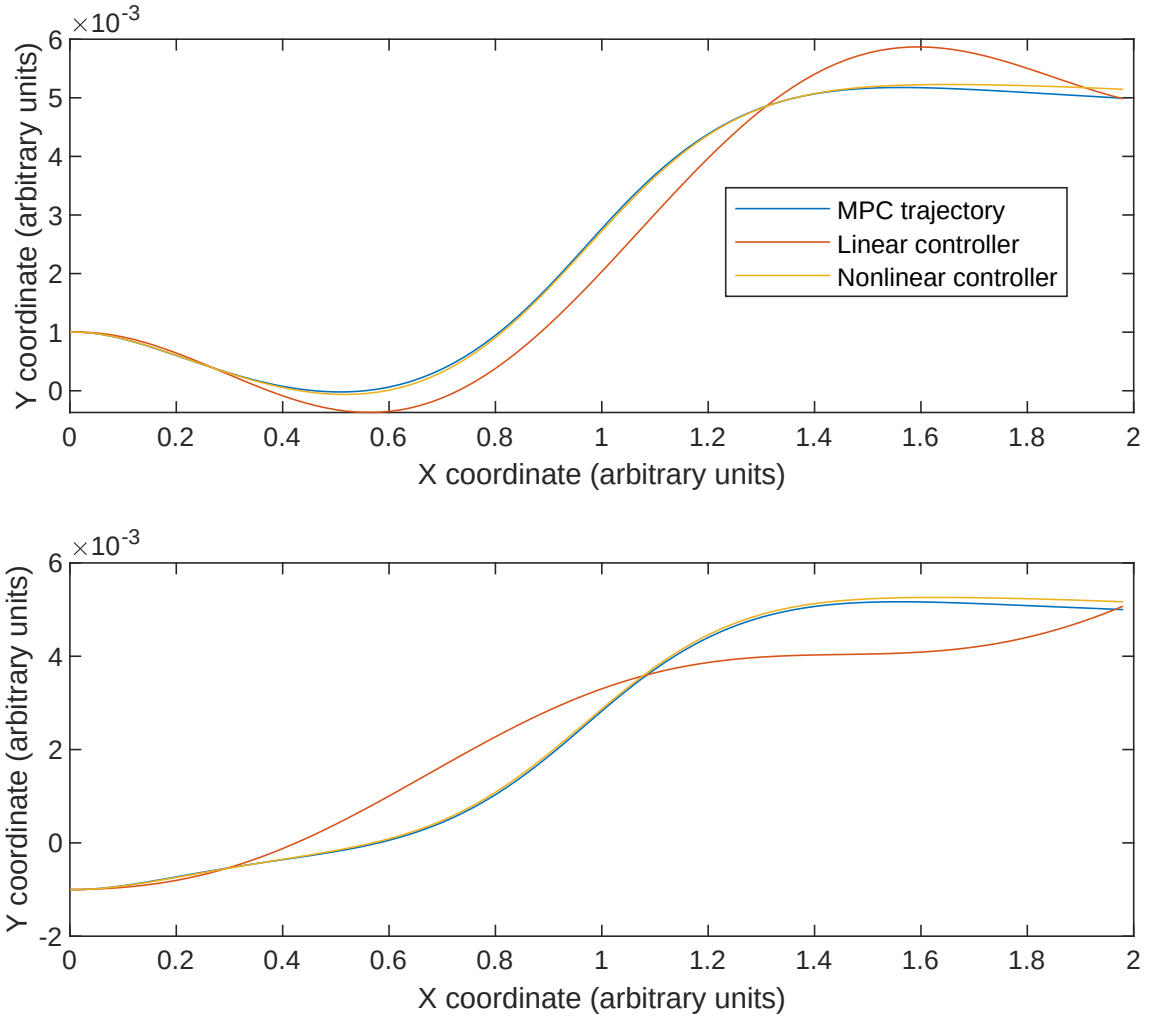


Figure 4.2: The trajectories generated by the linear and nonlinear controllers for the Dubin's vehicle example for each reference trajectory.

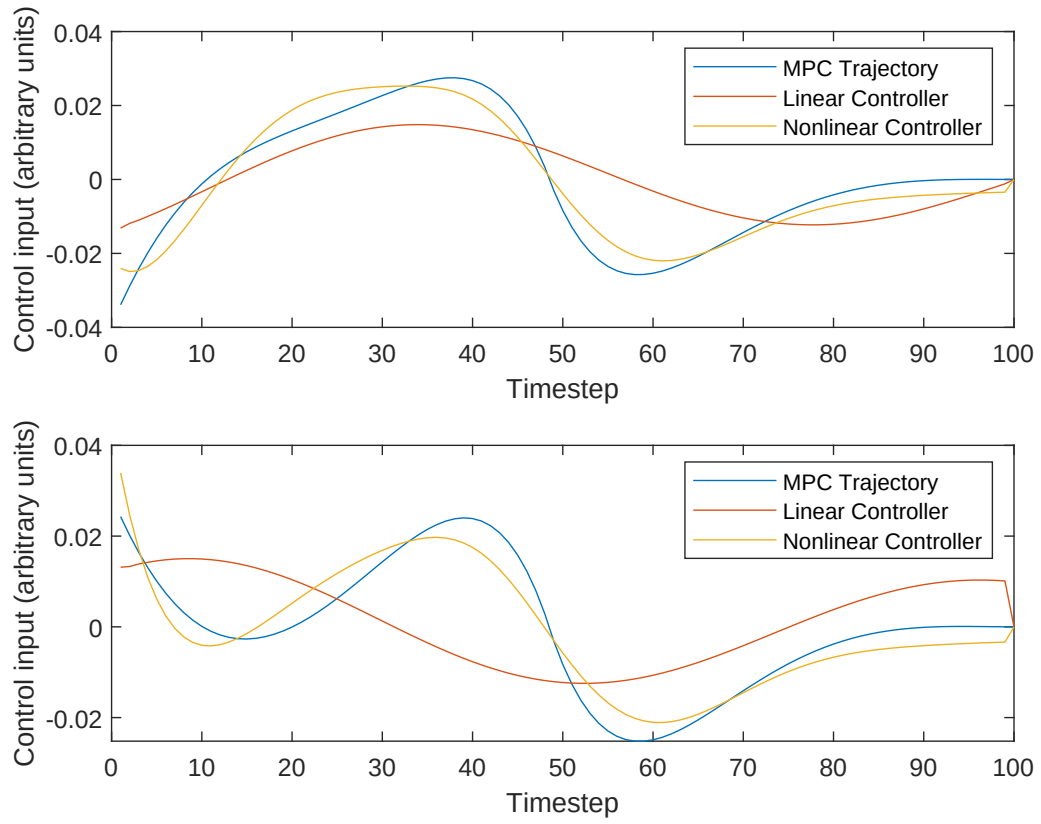


Figure 4.3: The control inputs generated by the linear and controllers versus time for the Dubin's vehicle example.

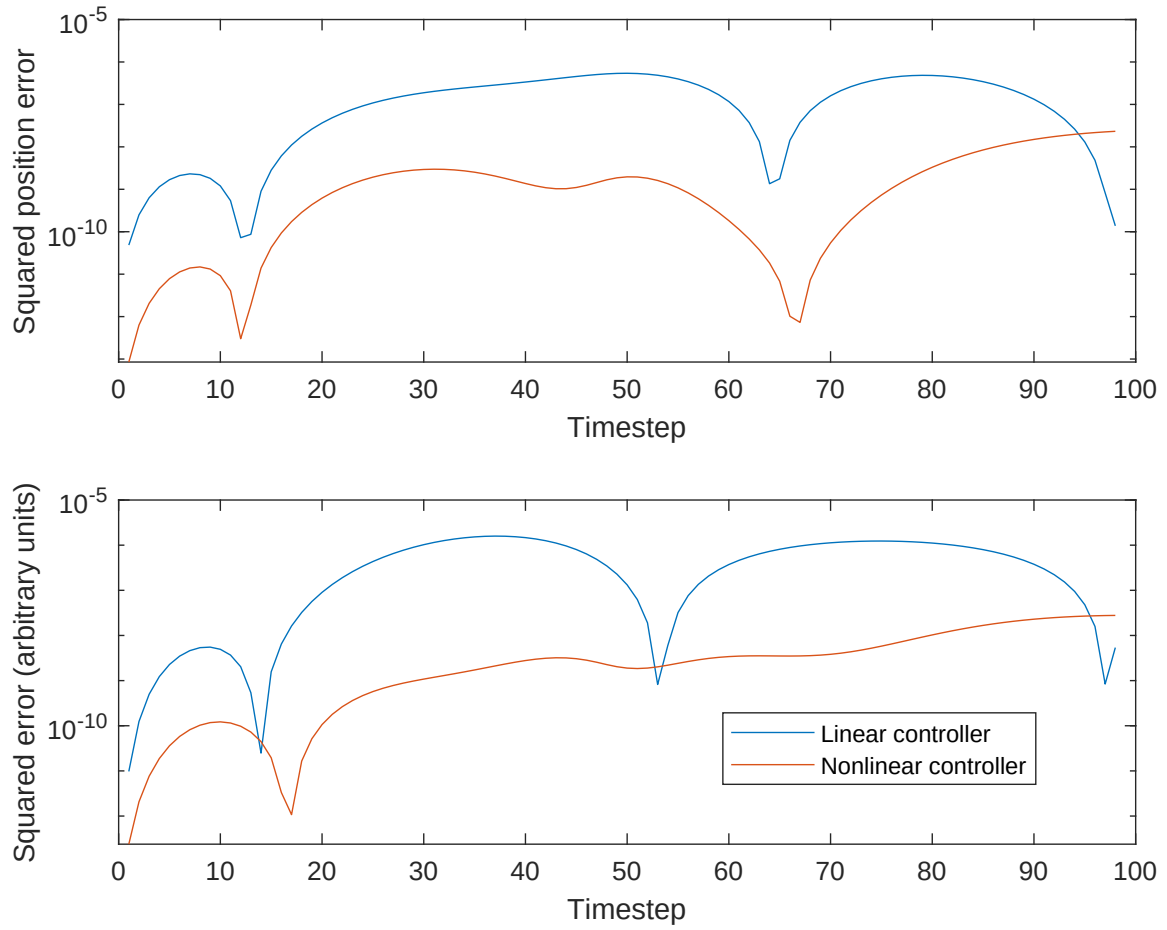


Figure 4.4: The squared position errors for the linear and controllers versus time for the Dubin's vehicle example.

Chapter 5

Conclusion

This thesis has described, justified, demonstrated, and evaluated a new approach for solving the trajectory-matching problem with explicit polynomial controllers in a way that guarantees contractivity. Chapter 2 summarized the literature related to this problem and identified a clear gap, which this thesis seeks to close. Chapter 3 presented two key theoretical results that support and justify the methods used here: Section 3.2 introduced and proved a way to represent contractivity constraints as a set of matrix inequalities, while Section 3.3 demonstrated an elegant way to take advantage of symmetries in the problem to reduce the dimensionality of the controller search space. Chapter 4 described the demonstration of this approach on two example problems directly motivated by real-world applications. Section 4.1 described the practical considerations and details involved in using this method in practice, while Section 4.2 demonstrated its effectiveness on two different problems: control of a single-machine infinite-bus (SMIB) system and a lane change manoeuvre for Dubin’s vehicle.

5.1 Future Work

A potential area for future work is the development of optimization algorithms tailored to the optimization problem described here. All optimization problems in this work were solved using an off-the-shelf general nonlinear optimization solver, which was slow and frequently failed to converge. A custom optimization solver might help alleviate these problems.

Another possible direction is the investigation of parametrized family of controllers

other than polynomial controllers. While polynomial functions have many desirable properties, they have a significant drawback: all nonzero polynomial functions are unbounded. This fact complicates stability analysis, and it means that global stability results are very difficult to prove in general.

A further topic for future work is the extension of the methods described in this thesis to plants without full state feedback. All the work in this thesis assumes that the controller has full access to the plant state, which is often not true in practice. An observer used in combination with the method described in this thesis might be sufficient to solve this problem, but the proof of contractivity described in Section 3.2 would need to be extended to observer-based controllers, and the proof of Theorem 3.5 would need revisions as well.

References

- [1] J. H. Lee, “Model predictive control: Review of the three decades of development,” *International Journal of Control, Automation, and Systems*, vol. 9, no. 3, pp. 415–424, 2011. [9](#)
- [2] P. Scokaert, D. Mayne, and J. Rawlings, “Suboptimal model predictive control (feasibility implies stability),” *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 648–654, 1999. [9](#)
- [3] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, “Review on model predictive control: an engineering perspective,” *International Journal of Advanced Manufacturing Technology*, vol. 117, pp. 1327–1349, 2021. [9](#)
- [4] T. Besselmann, J. Lofberg, and M. Morari, “Explicit MPC for LPV systems: Stability and optimality,” *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2322–2332, 2012. [9](#)
- [5] M. M. Nicotra and E. Garone, “The explicit reference governor: A general framework for the closed-form control of constrained nonlinear systems,” *IEEE Control Systems Magazine*, vol. 38, no. 4, pp. 89–107, 2018. [10](#)
- [6] J. C. Doyle, N. Matni, Y.-S. Wang, J. Anderson, and S. Low, “System level synthesis: A tutorial,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2856–2867, 2017. [10](#)
- [7] D. Ho, “A system level approach to discrete-time nonlinear systems,” in *2020 American Control Conference (ACC)*, pp. 1625–1630, 2020. [10](#)
- [8] L. Ljung, “Perspectives on system identification,” *Annual Reviews in Control*, vol. 34, no. 1, pp. 1–12, 2010. [10](#), [11](#)

- [9] T. C. Y. Chan, R. Mahmood, and I. Y. Zhu, “Inverse optimization: Theory and applications,” *Operations Research*, vol. 73, no. 2, pp. 1046–1074, 2025. [11](#)
- [10] G. Chou, N. Ozay, and D. Berenson, “Learning constraints from locally-optimal demonstrations under cost function uncertainty,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3682–3690, 2020. [11](#)
- [11] T. Sauer and Y. Xu, “On multivariate Lagrange interpolation,” *Mathematics of Computation*, vol. 64, no. 211, pp. 1147–1170, 1995. [21](#)
- [12] X. Wang, Y. Chen, G. Han, and C. Song, “Nonlinear dynamic analysis of a single-machine infinite-bus power system,” *Applied Mathematical Modelling*, vol. 39, no. 10, pp. 2951–2961, 2015. [27](#)
- [13] R. P. Anderson and D. Milutinović, “A stochastic approach to Dubins vehicle tracking problems,” *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2801–2806, 2014. [29](#)