Machine Learning in Feedback Systems: Provable Methods for Safe and Robust Autonomy

Niyousha Rahimi

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2024

Reading Committee:
Mehran Mesbahi, Chair
Behçet Açıkmeşe
Karen Leung
Amir Taghvaei

Program Authorized to Offer Degree:
Aeronautics and Astronautics Engineering

University of Washington

Abstract

Machine Learning in Feedback Systems:
Provable Methods for Safe and Robust Autonomy

Niyousha Rahimi

Chair of the Supervisory Committee:
Mehran Mesbahi
William E. Boeing Department of Aeronautics and Astronautics

This dissertation explores the integration of machine learning into feedback control systems, addressing key challenges in the realm of control theory with a focus on autonomous navigation. Modern advances in sensing technologies and computational methods have enabled remarkable advancements in data-guided control. However, the reliability of machine learning, particularly deep learning, in safety-critical applications remains limited due to its inadequate handling of uncertainty. Furthermore, traditional methods in control theory impose limitations when the system is operating in complex environments with unknown uncertainties. This research seeks to bridge this gap by combining robust and optimal control techniques with machine learning to ensure reliable automated system behavior.

Part I of the dissertation establishes a theoretical framework for the data-driven design of optimal controllers inspired by autonomous physical systems. It investigates the online regulation of both linear and nonlinear systems that are possibly unstable and partially unknown. A significant contribution of this part is the introduction of the concept of "regularizability," which characterizes the extent by which a system can be regulated in finite time, offering a new perspective on system behavior compared to traditional stabilizability and controllability. This theoretical exploration challenges conventional understandings and provides novel insights into finite time regulation versus asymptotic behavior.

Part II addresses the practical application of deep learning algorithms in processing high-dimensional data and generating a spectrum of outputs in automatic feedback control. The inherent challenge in this context is modeling uncertainties in the output, especially when these trained neural networks are employed as perception modules within control loops for autonomous navigation. To mitigate this, the dissertation introduces a novel approach utilizing a perception map as an approximate inverse. This perception-control loop demonstrates commendable attributes, provided that the controller is robustly designed to accommodate for the perception errors. The novelty of this part lies in developing methods to ensure robustness against state-dependent perception errors, thus contributing to more reliable machine learning applications in feedback control systems.

Table of Contents

List of Figures

Glossary

DGR: Data-Guided Regulation

F-DGR: Fast Data-Guided Regulation

DG-RAN: Data-Guided Regulator for Adaptive Nonlinear Control

SVD: Singular Value Decomposition

PBH: Popov-Belevitch-Hautus

LQR: Linear Quadratic Regulator

PI: Policy Iteration

RLS: Recursive Least Squares

SYSID: System Identification

PE: Persistently Exciting

LMI: Linear Matrix Inequalities

LTI: Linear Time-Invariant

MPC: Model Predictive Control

E-ISS: Exponentially Input-to-State Stable

UUB: Uniformly Ultimately Bounded

UCCM: Unmatched Control Contraction Metric

UCLF: Unmatched Control Lyapunov function

RL: Reinforcement Learning

CLF: Control Lyapunov Function

DMI: Differential Matrix Inequality

CNN: Convolutional Neural Network

DNN: Deep Neural Network

LEO: Low Earth Orbit

HCW: Hill-Clohessy-Wiltshire

MRP: Modified Rodriguez parameters

RCVS: Robust Controller Synthesis Method for Vision-based Spacecraft

Acknowledgments

I would like to express my profound gratitude to the following individuals who have played a significant role in shaping my academic and personal journey:

First and foremost, I would like to express my deepest gratitude to my Ph.D. advisor, Professor Mehran Mesbahi, for his unwavering support and invaluable guidance throughout my doctoral journey. His encouragement and profound insights have been instrumental in my academic and professional development. I am immensely thankful for the countless technical lessons in graph theory, optimization, optimal control theory, and learning in control that he has imparted to me. Professor Mesbahi's mentorship has not only facilitated my academic achievements but has also significantly contributed to my personal growth. His influence has extended far beyond the academic realm, and for that, I am truly grateful.

My heartfelt thanks go to my M.Sc. advisor, Ashis Banerjee, for his encouragement and support in the early years of my graduate studies. His invaluable guidance and insights have given me the push I needed to pursue my interest in this field.

I am indebted to Behçet Açıkmeşe, Maryam Fazel, Keren Leung, and Amir Taghvaei for their valuable contributions as committee members. My deepest gratitude goes to Professor Açıkmeşe for his insightful discussions and collaborations on various aspects of my research, including robust and optimal control, which greatly improved the outcomes of my dissertation. I would also like to extend special thanks to Professor Fazel, from whom I had the privilege to learn convex optimization. I feel truly fortunate to have been mentored by and learned from such an exceptional figure and role model. Her influence will resonate with me for years to come.

My sincere thanks go to Steve Brunton and Selim Tuncel for their insightful lectures

during my theoretical and mathematical education at the UW Department of Mechanical Engineering and the UW Department of Mathematics. Steve's remarkably organized lectures on topics including differential equations, dynamical systems, and data-driven control have been instrumental in deepening my understanding of the subject. I am deeply grateful to Selim for his joyful lectures, which introduced me to the beauty of real analysis and rekindled my love for mathematics.

I would like to acknowledge and express my gratitude to my collaborators, Shahriar Talebi, Siavash Alemzadeh, Aditya Deole, Taewan Kim, and Spencer Kraisler, for the amazing opportunities and fruitful teamwork we have shared. In particular, I extend special thanks to Amir Rahmani, Saptarshi Bandyopadhyay, and Jasper Corleis for their exciting research discussions and fruitful collaborations. I would also like to thank the current and former RAIN members: Mengyuan Wang, Shiva Shakeri, Josh Holder, Beniamino Pozzan, Jingjing Bu, Mathias Hudoba de Badyn, Dillon Foight, Taylor Reynolds, and Bijan Barzgaran. The time spent with all of you discussing various research questions has created a wonderful research environment, and I am immensely grateful for your contributions. Working with all of you has been an absolute pleasure.

Dedication

To my Mother, and my family.

Those I lost and those I gained along this journey.

Chapter 1

Introduction

Feedback synthesis has recently been approached from the perspective of data-guided control. This trend has been encouraged by modern sensing technologies and efficient computational methods. Machine learning is a promising tool for processing complex information, but it remains unreliable for control and decision-making, particularly in safety-critical systems. In general, deep learning does not consider uncertainty representation in regression settings, and deep learning classification models often give normalized score vectors, which do not necessarily capture model uncertainty. To this end, the control community seeks to combine the best of learning with robust/optimal control for autonomous navigation. Toward the goal of ensuring reliable behavior, this thesis takes steps toward understanding the trade-offs and limitations that arise in feedback settings.

1.1 Statement of Contributions

This dissertation seeks to make contributions to the realm of control theory with a specific focus on the applications of machine learning for automatic feedback control.

In Part I, we draw inspiration from autonomous physical systems to lay a theoretical groundwork for the data-driven design of optimal controllers. We examine the online regulation of both linear and nonlinear systems that are possibly unstable and partially unknown. Herein, we harness the power of prior knowledge about the system for online regulation. The distinguishing contribution of this part lies in the introduction of the concept of "regularizability." This novel term characterizes the extent to which a system, linear or nonlinear, can be regulated in finite time, offering a stark contrast to its asymptotic behavior, which is typically characterized by stabilizability or controllability. Our explorations challenge the conventional

understanding of system behavior, offering new perspectives on finite time regulation versus asymptotic behavior.

Transitioning into Part II, we explore the potential of deep learning algorithms in processing high-dimensional data and generating a spectrum of outputs when applied to automatic feedback control. Notwithstanding their remarkable potential, the inherent challenge lies in modeling the uncertainties in the output. This issue becomes particularly salient when these trained neural networks are employed as a perception module within the control loop for autonomous navigation—a setting that necessitates stringent safety assurances. In response, we introduce an innovative approach utilizing a perception map, which functions as an approximate inverse. Our investigations illustrate that this perception-control loop boasts commendable attributes, granted the controller is robustly designed to accommodate for perception errors. The novelty of this part lies in the development of a method for ensuring robustness against state-dependent perception errors, contributing to more reliable machine learning applications in the field of automatic feedback control.

1.2 Outline of Dissertation

This dissertation is organized as follows. First, Chapter 2 provides some background information that will serve as a common point of reference for each subsequent chapter. We introduce important assumptions and give formal statements of the tools from optimal and robust control theory that form the core building blocks of our proposed methods. Next, Chapter 3 provides a thorough overview of theoretical advances in using learning methods in feedback control systems. We study and derive characteristics of model-based vs model-free control problems. Applications of high-dimensional data as a perception module are further studied in the subsequent part of this section.

In Part I, we attempt to build a theoretical foundation for the data-driven design of optimal controllers. Chapters 4 and 5 are devoted to proving safety and performance guarantees for learning-based controllers applied to *unstable* systems with partially-known dynamics. Particularly, in Chapter 4, we focus on the online regulation of unstable, partially unknown

linear time-invariant (LTI) systems with no a priori assumptions on the initial controller. Our method can serve as a remedy to the impractical assumption of initializing from stabilizing controllers that is often used in almost every work in the literature on iterative data-driven control.

Further, in Chapter 5, we investigate the efficacy of a data-driven feedback controller in achieving finite-time regulation of system states within the feasible state space of a complex nonlinear dynamical system. The study specifically focuses on the controller's performance in the presence of unknown time-varying uncertainties, which pose challenges to system regulation. By analyzing the controller's effectiveness, we aim to provide insights into its ability to effectively handle uncertainties and achieve the desired regulatory objectives within a finite time horizon.

In Part II, we examine the problem of using perceptual information in feedback control loops. We focus on the practical scenario where the underlying dynamics of a system are well understood, and it is instead the interaction with a perceptual sensor that is the limiting factor.

Chapter 6 proposes a novel approach involving the design of a virtual sensor, which includes learning a perception map and a state-dependent bound on estimation errors. The proposed approach explicitly models the error of the perception process as a function of relative states for a target tracking problem. By incorporating the Uniformly Ultimately Bounded (UUB) concept and leveraging the invariant funnel synthesis technique, we define feasible trajectories in state-and-control spaces, which account for state-dependent output measurement noise. This enables the effective incorporation of the error model and its upper bounds. The ultimate contribution of this section is the development of a composite robust control synthesis. This synthesis generates control signals directly from imagery data, effectively handling uncertainties arising from both the nonlinear model and the noisy observations.

The conclusion of the work presented in this dissertation and the future directions are laid out in Part III.

1.3 Included Publications

Some of the ideas and figures have appeared previously in the following publications:

[2] N. Rahimi and M. Mesbahi, "Data Guided Regulator for Adaptive Nonlinear Control," in AIAA SCITECH 2024 Forum, p. 0090, 2024.

[3] N. Rahimi, S. Talebi, A. Deole, M. Mesbahi, S. Bandyopadhyay, and A. Rahmani, "Robust controller synthesis for vision-based spacecraft guidance and control," in AIAA SCITECH 2022 Forum, p. 2213, 2022.

[4] S. Talebi, S. Alemzadeh, N. Rahimi, and M. Mesbahi, "On regularizability and its application to online control of unstable lti systems," IEEE Transactions on Automatic Control, 2021.

[5] S. Talebi, S. Alemzadeh, N. Rahimi, and M. Mesbahi, "Online regulation of unstable linear systems from a single trajectory," in 59th IEEE Conference on Decision and Control (CDC), pp. 4784–4789, 2020.

[6] J. Becktor, W. Seto, A. Deole, S. Bandyopadhyay, N. Rahimi, s. Talebi, M. Mesbahi, and A. Rahmani, "Robust vision-based multi-spacecraft guidance navigation control using cnn-based pose estimation," in 2022 IEEE Aerospace Conference, pp. –, IEEE, 2022

Chapter 2

Background

This chapter summarizes the preliminaries and notations used throughout the dissertation pertaining to linear algebra, optimal and nonlinear control, and funnel synthesis.

2.1 Linear Algebra

We denote the fields of real and complex numbers by $\mathbb{R}$ and $\mathbb{C}$, respectively. Real $n \times m$ matrices are denoted by $\mathbb{R}^{n\times m}$, and $\mathbb{S}$ refers to the set of symmetric matrices. A column vector with n elements is referred to as $\boldsymbol{v} \in \mathbb{R}^n$, where $\boldsymbol{v}_i$ represents the i^{th} element in $\boldsymbol{v}$. The square matrix $N \in \mathbb{S}^{n\times n}$ is symmetric if $N^\intercal = N$, where $N^\intercal$ denotes the transpose of the matrix N.

The $n \times 1$ vector of all ones is denoted by $\mathbb{1}$. The unit vector $\boldsymbol{e}_i$ is a column vector with identity at its i^{th} entry and zero elsewhere. The $n \times n$ identity matrix is denoted by I_n. The diag(.) indicates a diagonal matrix constructed by elements of its argument in the same order starting from the upper-left corner. For a real symmetric matrix L, we say that $L \succ 0$ when L is positive-definite (PD) and $L \succeq 0$ for the positive-semidefinite (PSD) case. A matrix is positive-definite if and only if all of its leading principle minors are positive. The i^{th} eigenvalue and spectral radius of M are denoted by $\lambda_i(M)$ and $\rho(M)$ and M is Schur stable if $\rho(M) < 1$. The algebraic multiplicity of an eigenvalue λ is denoted by $m(\lambda)$; λ is called simple if $m(\lambda) = 1$. The range and null space of a real matrix $M \in \mathbb{R}^{n\times m}$ are denoted by $\mathcal{R}(M) \subseteq \mathbb{R}^n$ and $\mathcal{N}(M) \subseteq \mathbb{R}^m$, respectively, the dimension of $\mathcal{R}(M)$ is designated by rank(M), and its transpose by $M^\intercal$. We define $\mathcal{R}(M)$ to be A-invariant if there exists C such that $AM = MC$. The dimension of a vector space is denoted by **dim**. The span of a set of vectors over the complex field is denoted by span$\{.\}$. The singular value decomposition of a matrix $M \in \mathbb{R}^{n\times m}$

is the factorization $M = U\Sigma V^\intercal$, where the unitary matrices $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{m\times m}$ consist of the left and right "singular" vectors of M, and $\Sigma \in \mathbb{R}^{n\times m}$ is the diagonal matrix of singular values in descending order. The reduced order matrices U_r, V_r can be obtained by truncating the factored matrices U and V in the SVD to the first r columns, where $r = \text{rank}(M)$. The thin SVD of M is then the factorization $M = U_r\Sigma_r V_r^\intercal$, where $\Sigma_r \in \mathbb{R}^{r\times r}$ is now nonsingular. From SVD, one can also construct the Moore-Penrose generalized inverse —*pseudoinverse* for short— of M as $M^\dagger = V\Sigma^\dagger U^\intercal$, in which $\Sigma^\dagger$ is obtained from Σ by first replacing each nonzero singular value with its inverse (zero singular values remain intact) followed by a transpose. The matrix A is (complex) diagonalizable if there exist a diagonal matrix $\Lambda \in \mathbb{C}^{n\times n}$ and a nonsingular matrix $U \in \mathbb{C}^{n\times n}$ such that $A = U\Lambda U^{-1}$. In this case, Λ consists of the eigenvalues of A with columns of U as the corresponding eigenvectors. We say that $\boldsymbol{x}_0$ *excites k modes of a matrix A* if $\boldsymbol{x}_0$ is contained in the (complex-)span of k eigenvectors of A, but not in the span of any $k-1$ eigenvectors; we refer to those k eigenvectors (for which $\boldsymbol{x}_0$ is in the span of) as the corresponding *excited modes*.

The orthogonal projection of a vector $\boldsymbol{v}$ on a linear subspace S is denoted by $\Pi_S(\boldsymbol{v})$.[1] When the columns of a matrix $U \in \mathbb{R}^{n\times k}$ form an orthonormal basis for the subspace S, then $\Pi_S = UU^\intercal$. The Euclidean norm of a vector $\boldsymbol{x} \in \mathbb{R}^n$ is denoted by $\|\boldsymbol{x}\| = (\boldsymbol{x}^\intercal\boldsymbol{x})^{1/2}$, and the dual norm to $\|\boldsymbol{x}\|$ is denoted by $\|\boldsymbol{x}\|_* := \sup_{\|\boldsymbol{u}\|=1}\langle \boldsymbol{x}, \boldsymbol{u}\rangle$. Also, the 1-norm is defined as $\|x\|_1 = \sum_{i=1}^n |x_i|$. For a matrix M, its operator norm is denoted by $\|M\| = \sup\{\|M\boldsymbol{u}\| : \|\boldsymbol{u}\| = 1\}$. By $\mathcal{B}_2^r$, we refer to the r-dimensional Euclidean ball of unit radius. An r-dimensional multi-index α is an r-tuple of the form $(\alpha_1, \alpha_2, \cdots, \alpha_r)$ with all non-negative integers α_i, where the sum of its elements is denoted by $|\alpha| = \sum_{i=1}^r \alpha_i$; $\alpha \in \{0,1\}^r$ signifies that each $\alpha_i \in \{0,1\}$ for $i = 1, \ldots, r$.

A function f is convex if $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$ for all $\theta \in (0,1)$ and for all x, y in its convex domain, and g is a subgradient of f at point z if $f(y) \geq f(z) + g^\intercal(y-z)$ for all y. If f is convex and differentiable, the gradient of $f, \nabla f(x)$, is also a subgradient of f

[1] We will be working with finite dimensional vector spaces, and as such, all subspaces are closed.

at x. The set of all subgradients of f at x is called subdifferential and denoted by $\partial f(x)$.

2.2 Optimal Control Problems

The equations of motion for a physical system are assumed to be of the form

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t), \boldsymbol{p}), \quad t \in [t_0, t_f]\,, \tag{2.1}$$

where $t_0, t_f \in \mathbb{R}_+$ represent the initial and final times respectively, $\boldsymbol{x}(\cdot) : \mathbb{R} \to \mathcal{X} \subset \mathbb{R}^{n_x}$ is the state, $\boldsymbol{u}(\cdot) : \mathbb{R} \to \mathcal{U} \subset \mathbb{R}^{n_u}$ is the control, $\boldsymbol{w}(\cdot) : \mathbb{R} \to \mathbb{R}^{n_w}$ is the disturbance, and $\boldsymbol{p} \in \mathcal{P} \subset \mathbb{R}^{n_p}$ is the parameter vector. When the state or control vectors are referred to, we mean $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$, respectively. The sets $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{P}$ represent the sets of admissible state, control, and parameter vectors.

Assumption 2.1. *We assume the function $\boldsymbol{u}(\cdot)$ is piecewise continuous with respect to time, and the set $\mathcal{U}$ is compact. Further, we assume the function f is continuous with respect to $\boldsymbol{u}(t)$ and $\boldsymbol{p}$.*

In control theory, an optimal control problem is a problem in which we are looking for a control (or manipulation) strategy for a given system that optimizes (i.e., maximizes or minimizes) a certain performance measure, also known as a cost functional or objective function, subject to certain constraints. The objective is to find the control law or policy that will guide the system from its initial state to a desired final state in an optimal way. The cost is described by using a functional - a scalar-valued function over a space of functions - and is assumed to have the general form

$$J(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}) = M\left(\boldsymbol{x}\left(t_0\right), \boldsymbol{x}\left(t_f\right), \boldsymbol{p}\right) + \int_{t_0}^{t_f} L(\boldsymbol{x}(\zeta), \boldsymbol{u}(\zeta), \boldsymbol{p}) \mathrm{d}\zeta \tag{2.2}$$

where $M : \mathcal{X}_0 \times \mathcal{X}_f \to \mathbb{R}$ is the terminal cost and $L : \mathcal{X} \times \mathcal{U} \times \mathcal{P} \to \mathbb{R}$ is the running cost. Explicit dependence on the state is present in the cost function J both because of the dependence of M on the initial state vector and for notation consistency (i.e., as a matter of taste). The act of solving an optimal control problem is equivalent to simply selecting the solution with the lowest or highest such cost.

An optimal control problem with a cost function of the form (2.2) is called a *Bolza problem*. When $L \equiv 0$, the problem is called a *Mayer problem*, and when $M \equiv 0$, the problem is called a *Lagrange problem*. No form is more general than another, but different forms can be more convenient at different times.

Having stated in words what an optimal control problem is, some mathematical notation is now adopted to formalize the statement. Optimal control problems can be stated so as to minimize or maximize the cost, and we will see examples of both at various points in the text. The mathematical statement of an optimal control problem is given in Problem 1, and is understood to carry with it all assumptions on the constituent functions/sets that have been made up to this point.

Problem 2.1. Find the piecewise continuous control signal $\boldsymbol{u}(\cdot)$, initial state $\boldsymbol{x}(t_0)$ and parameter vector $\boldsymbol{p}$ that solve the following problem:

$$\begin{aligned} \min_{\boldsymbol{x}(t_0),\boldsymbol{u}(\cdot),\boldsymbol{p}} \quad & J(\boldsymbol{x},\boldsymbol{u},\boldsymbol{p}) \\ \text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{p}) \\ & S_0\left(\boldsymbol{x}\left(t_0\right),\boldsymbol{p}\right) = 0, \quad S_f\left(\boldsymbol{x}\left(t_f\right),\boldsymbol{p}\right) = 0, \\ & \boldsymbol{x}(t) \in \mathcal{X}, \quad \boldsymbol{u}(t) \in \mathcal{U}, \quad \boldsymbol{p} \in \mathcal{P}. \end{aligned}$$

where $t \in [t_0, t_f]$ for each time-dependent constraint. The initial condition $\boldsymbol{x}(t_0)$ is assumed to satisfy

$$S_0\left(\boldsymbol{x}\left(t_0\right),\boldsymbol{p}\right) = 0 \tag{2.3}$$

for some function $S_0 : \mathcal{X} \times \mathcal{P} \to \mathbb{R}^{n_0}$. We make the standard assumption that $n_0 \leq (n_x + n_p)$ and that the Jacobian matrix $\nabla S_0\left(\boldsymbol{x}\left(t_0\right),\boldsymbol{p}\right) \in \mathbb{R}^{n_0 \times (n_x+n_p)}$ exists and has full row rank. The initial boundary condition (2.3) effectively restricts the set of initial conditions to lie in some subset $\mathcal{X}_0 \subset \mathcal{X} \times \mathcal{P}$. The target set is described in a similar way to $\mathcal{X}_0$ by using

$$S_f\left(\boldsymbol{x}\left(t_f\right),\boldsymbol{p}\right) = 0 \tag{2.4}$$

for some function $S_f : \mathcal{X} \times \mathcal{P} \to \mathbb{R}^{n_f}$. Again, the standard assumption that $n_f \leq (n_x + n_p)$ and the Jacobian $\nabla S_f\left(\boldsymbol{x}\left(t_f\right),\boldsymbol{p}\right) \in \mathbb{R}^{n_f \times (n_x+n_p)}$ exists and has full row rank is made. The

terminal boundary condition (2.4) effectively restricts the set of target (or final) conditions to lie in some subset $\mathcal{X}_f \subset \mathcal{X} \times \mathcal{P}$.

2.3 Linear Quadratic Regulator

Linear Quadratic Regulator (LQR) is a classical method in optimal control theory where the states evolve based on a linear equation that correlates the states and control, and the evaluation is based on a quadratic index measure. Assume the dynamics follows,

$$\boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + B\boldsymbol{u}(k), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{2.5}$$

in a discrete-time setup. The dynamics can also be deemed as continuous time in the form of,

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}, \quad \boldsymbol{x}(0) = \boldsymbol{x}_0. \tag{2.6}$$

Here, $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector of the system with the given initial state $\boldsymbol{x}_0$, and $\boldsymbol{u} \in \mathbb{R}^m$ is the control signal used to regulate the system. The pair (A, B) are the system parameters that dictate the dynamics. We call the pair (A, B) controllable, if and only if the controllability matrix $\mathcal{C} = \begin{bmatrix} B & AB & \dots A^{n-1}B \end{bmatrix}$ has full-rank, where n is the size of the system. We assume that this pair is fixed for every analysis presented in this thesis. The objective is to find $\boldsymbol{u}$ that drives the state from $\boldsymbol{x}_0$ towards zero in an optimal way. In order to quantify optimality, a quadratic cost is specified as,

$$J(\boldsymbol{x}, \boldsymbol{u}) = \sum_{k=0}^{\infty} \boldsymbol{x}(k)^\intercal Q \boldsymbol{x}(k) + \boldsymbol{u}(k)^\intercal R \boldsymbol{u}(k), \tag{2.7}$$

where Q and R are the penalty parameters on the states and control signals, respectively. Within the framework of LQR, this cost is minimized subject to the linear dynamics introduced above. The solution to the discrete-time LQR is shown to rely on the solution to the discrete Algebraic Riccati Equation (ARE),

$$P = A^\intercal P A + Q - A^\intercal P B \left(R + B^\intercal P B\right)^{-1} B^\intercal P A. \tag{2.8}$$

Then the optimal control law in the discrete-time case is linear and defined by,

$$\boldsymbol{u}^*(k) = -\left(R + B^\intercal P B\right)^{-1} B^\intercal P A \boldsymbol{x}(k) = -K\boldsymbol{x}(k). \tag{2.9}$$

The policy derived above is also referred to as linear feedback control. For any stabilizing linear control law $\boldsymbol{u}(k) = -K\boldsymbol{x}(k)$, the value of the cost can be computed by finding the solution to the discrete-time Lyapunov equation,

$$P = (A - BK)^\intercal P (A - BK) + Q + K^\intercal R K, \tag{2.10}$$

whose solution can be expressed as an infinite sum,

$$P = \sum_{k=0}^{\infty} \left((A - BK)^\intercal\right)^k Q (A - BK)^k \tag{2.11}$$

where $A_{\text{cl}} = A - BK$ is referred to as the closed-loop system from $\boldsymbol{x}(k+1) = (A - BK)\boldsymbol{x}(k)$. Note that depending on how the policy is updated ($\boldsymbol{u} = -K\boldsymbol{x}$ or $\boldsymbol{u} = K\boldsymbol{x}$) we switch the closed-loop system to $A_{\text{cl}} = A + BK$. Also, when the linear quadratic system as described above follows the law $\boldsymbol{u} = -K\boldsymbol{x}$, then the cost function can be determined as,

$$J(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{x}_0^\intercal P \boldsymbol{x}_0. \tag{2.12}$$

2.4 Nonlinear Dynamical Systems

Consider the following nonlinear dynamical system,

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad t \in [t_0, t_f], \tag{2.13}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ is the system states and $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$ is the control input. We assume the $f(\boldsymbol{x}(t), \boldsymbol{u}(t))$ is locally Lipschitz uniformly and at least once differentiable, with measurable state $\boldsymbol{x}$. Although the dynamics (2.13) is continuous, we can only measure system states and control inputs at discrete times. Hence, we consider the following assumption when measuring data:

Assumption 2.2. *The system states can be measured at constant (positive) intervals of length $\delta t \in \mathbb{R}$. Hence at time t, we have a streaming data set $\mathcal{D}_t = \{x_i, u_i\}_{i=0}^{n_t}$ of size $n_t = \lfloor t/\delta t \rfloor$.*

Assume that $\{\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t)\}_{t=t_0}^{t_f}$ is the nominal (reference) trajectory that satisfies the dynamics (2.13), for some initial condition $\boldsymbol{x}_0$, and define,

$$\eta(t) \coloneqq \boldsymbol{x}(t) - \bar{\boldsymbol{x}}(t), \tag{2.14a}$$

$$\xi(t) \coloneqq \boldsymbol{u}(t) - \bar{\boldsymbol{u}}(t), \tag{2.14b}$$

where $\eta(t)$ is the deviation from the nominal trajectory. We can rewrite (2.13) in terms of the variables in (2.14) by using a first-order Taylor series expansion around the nominal trajectory as

$$\dot{\eta}(t) = A(t)\eta(t) + B(t)\xi(t) + \delta g\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right), \tag{2.15}$$

where $A(t)$ and $B(t)$ are the partial derivatives of f evaluated along the nominal trajectory, and $\delta g\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right) = g(\boldsymbol{x}(t), \boldsymbol{u}(t)) - g(\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t))$ represents the higher order (nonlinear) terms.

Assumption 2.3. *The higher order (nonlinear) term, $g\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right)$, is locally Lipschitz bounded.*

The general practice is to design a robust controller (with state denoted by $\xi(t)$) offline by considering the uncertainty that is known a priori [3]. We precisely assume that the offline robust controller is designed using the Lyapunov theory. Suppose that $\mathcal{X}_{\mathcal{F}} \in \mathbb{R}^{n_x}$ and $\mathcal{U}_{\mathcal{F}} \in \mathbb{R}^{n_u}$ are the (possibly nonconvex) sets of feasible state and control vectors. A Lyapunov function can be used to seek out nearby feasible trajectories around the nominal $\{\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t)\}_{t=t_0}^{t_f}$. In other words, denote a funnel by

$$\mathcal{F}(t) \subseteq \mathcal{X}_{\mathcal{F}} \times \mathcal{U}_{\mathcal{F}}.$$

The set $\mathcal{F}(t)$ is comprised of time-varying state and control trajectories that are invariant and contained inside the respective feasible regions.

For the original system dynamic $\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t))$, one can use the Lyapunov theory to design a robust feedback controller $\xi(t)$, which guarantees the invariance of $\mathcal{F}(t)$.

2.4.1 Quadratic Funnel Synthesis

The funnel synthesis [7] approach is based on the notion of quadratic stability as defined in [8] and [9–11]. For a fixed reference trajectory $\{\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t)\}_{t=t_0}^{t_f}$, the error dynamic for $\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t))$ can be equivalently expressed using structured nonlinearities as follows

$$\dot{\eta}(t) = A(t)\eta(t) + B(t)\xi(t) + E\delta p(t) \tag{2.16}$$

where $\eta(t)$ is defined in (2.14a), $\xi(t) = K(t)\eta(t)$, and the pair $(\delta q(t), \delta p(t)) \in \mathbb{R}^{n_q} \times \mathbb{R}^{n_p}$ capture the higher order terms through the nonlinear functions $\phi_i : \mathbb{R}^{n_{q_i}} \to \mathbb{R}^{n_{p_i}}$.

$$\delta q(t) = q(t) - \bar{q}(t) = (H + GK(t))\,\eta(t)\,, \quad \text{where} \quad \bar{q}(t) = H\bar{x} + G\bar{u} \tag{2.17a}$$

$$\delta p(t) = \phi\,(t, q(t)) - \phi\,(t, \bar{q}(t))\,, \tag{2.17b}$$

We refer the reader to [12], where the details of such formulation for funnel synthesis are discussed.

The goal is to seek the largest possible funnel that satisfies input and state constraints. This allows us to implicitly define a large family of trajectories by using the functions that define the funnel, thereby providing the ability to guarantee the availability of a feasible trajectory over a larger region of parameter variations. Particularly, consider the scalar-valued function $V : \mathbb{R}^{n_x} \to \mathbb{R}$ defined by

$$V(\eta(t)) = \eta(t)^{\intercal} Q(t)^{-1} \eta(t) \tag{2.18}$$

where $Q(t) \in \mathbb{S}_{++}^{n_x}$ is a matrix-valued function of time whose range space lies in the set of positive definite matrices. As a result, we have $V(\eta(t)) > 0$ for all $t \in [t_0, t_f]$ whenever $\eta(t) \neq 0$.

Having introduced each of the time-varying terms, we henceforth omit the argument of time "t" whenever possible. The 1-level set of $V(\eta(t))$ is the set of states that satisfy the quadratic inequality $\eta^{\intercal} Q^{-1} \eta \leq 1$, which is also the equation of a non-degenerate n_x-dimensional ellipsoid. We denote the ellipsoid defined by the positive definite matrix Q and

centered at the origin as

$$\mathcal{E}_Q \coloneqq \left\{\eta \in \mathbb{R}^{n_x} \mid \eta^\intercal Q^{-1}\eta \leq 1\right\} \tag{2.19}$$

If $\eta \in \mathcal{E}_Q$, then $C\eta \in \mathcal{E}_{CQC^\intercal}$, a fact that can be proven easily via Schur complements when C is full row-rank[2].

We can now formally define a quadratic funnel as described in [7].

Definition 2.1. (Quadratic Funnel). A quadratic funnel, $\mathcal{F}$, is a set in state and control space that is parameterized by a time-varying positive definite matrix $Q(t) \in \mathbb{S}^{n_x}_{++}$ and a time-varying matrix $K(t) \in \mathbb{R}^{n_u \times n_x}$. Specifically, we have

$$\mathcal{F} = \mathcal{E}_Q \times \mathcal{E}_{KQK^\intercal}, \quad \mathcal{E}_Q \subseteq \mathcal{X}_\mathcal{F}, \quad \mathcal{E}_{KQK^\intercal} \subseteq \mathcal{U}_\mathcal{F} \tag{2.20}$$

K is called the correction law associated with the quadratic funnel.

The authors in [7] showed that the following optimization renders the maximum feasible quadratic funnel in the state and control space, where a time-varying feedback controller of the form $\xi(t) = K(t)\eta(t)$ guarantees convergence to the nominal trajectory, starting anywhere inside the funnel.

Problem 2.2. (Quadratic Funnel Synthesis [7]). Given a nominal trajectory $\{\bar{x}(t), \bar{u}(t)\}_{t=t_0}^{t_f}$ that satisfies the nonlinear dynamics $\dot{x}(t) = f(x(t), u(t))$, an appropriate definition of the nonlinearity channels $(\delta q(t), \delta p(t)) \in \mathbb{R}^{n_q} \times \mathbb{R}^{n_p}$ and $\alpha > 0$, find the matrix-valued functions of time $Q(t), Y(t)$ and $M_\gamma(t)$ and scalar $\lambda(t)$ that solve the following optimization problem $\forall t \in [t_0, t_f]$.

---

[2]When C is not full row-rank, the ellipsoid $\mathcal{E}_{CQC^\intercal}$ is a degenerate ellipsoid.

$$\max_{Q(\cdot),Y(\cdot),\lambda(\cdot),M_\gamma(\cdot)} \quad \log\det Q\left(t_0\right) \tag{2.21}$$

subject to

$$0 \preceq Q \preceq Q_{\max}, \;\; 0 \leq \lambda, \;\; M_\gamma \in \mathcal{M}_{\phi,\Omega}, \tag{2.21a}$$

$$\begin{pmatrix} \mathbf{Q}A^\intercal + A\mathbf{Q} + T_1 & E + \lambda \mathbf{Q} C_{cl}{}^\intercal M_{12} \\ E^\intercal + \lambda M_{12}^\intercal C_{cl}{}^\intercal \mathbf{Q} & \lambda M_{22} \end{pmatrix} \preceq 0, \tag{2.21b}$$

$$\begin{pmatrix} Q & Y^\intercal \\ Y & R_{\max} \end{pmatrix} \succeq 0, \tag{2.21c}$$

where $\Omega = \mathcal{E}_{C_{cl}QC_{cl}^\intercal}$, $M_\gamma = \operatorname{diag}\left(\begin{bmatrix} \gamma^2 I & -I \end{bmatrix}\right)$, $C_{cl} = H + G\mathbf{K}$, $\mathbf{Y} = K\mathbf{Q}$, and $T_1 = -\dot{\mathbf{Q}} + \alpha\mathbf{Q} + \mathbf{Y}^\intercal B^\intercal + B\mathbf{Y} + \lambda \mathbf{Q} C_{cl}^\intercal M_{11} C_{cl}\mathbf{Q}$.

Remark 2.1. The ellipsoids $\mathcal{E}_{Q_{\max}}$, and $\mathcal{E}_{R_{\max}}$ defined by the positive definite matrices $Q_{\max} \succ 0$ and $R_{\max} \succ 0$ are respectively the maximum volume ellipsoids in the feasible state and control space.

$$\mathcal{E}_{Q_{\max}} \subseteq \mathcal{X}_{\mathcal{F}}, \quad \mathcal{E}_{R_{\max}} \subseteq \mathcal{U}_{\mathcal{F}}$$

Furthermore, as defined in [7], $\mathcal{M}_{\phi,\Omega}$ is the set of local multiplier matrices for the nonlinear mapping $\phi : \mathbb{R}^{n_q} \to \mathbb{R}^{n_p}$ over the set Ω, and any symmetric matrix $M \in \mathbb{S}^{(n_q+n_p)}$ such that $M \in \mathcal{M}_{\phi,\Omega}$, satisfies the following inequality.

$$\begin{pmatrix} q \\ \phi(q) \end{pmatrix}^\intercal M \begin{pmatrix} q \\ \phi(q) \end{pmatrix} \geq 0, \quad \text{for all } q \in \Omega. \tag{2.22}$$

Chapter 3

Machine Learning in Feedback systems

In this chapter, we present an overview of theoretical advances in the application of learning methods to feedback control systems. Section 3.1 delineates the characteristics of model-based versus model-free control problems. Additionally, Section 3.2 delves into the use of high-dimensional data as a perception module. The chapter concludes with an introduction to the simulation environments utilized in this dissertation.

3.1 The Evolution of Learning in Feedback Systems

Learning in feedback systems has advanced significantly over the years, motivated by novel sensing technologies, robust machine learning, and efficient computational methods. This evolution has been characterized by continuous efforts to handle increasingly complex system dynamics while balancing computational cost, model precision, robustness, and control performance. This review aims to trace this evolution, highlighting key milestones, enduring challenges, and recent efforts to bridge the gap between model-based and model-free control strategies.

3.1.1 System Identification and Control

System identification is an essential method in control systems engineering that focuses on the development of mathematical models from observed data. In the context of LTI systems, many research papers have explored various methodologies. The most classical approach is the Least Squares (LS) method, which minimizes the sum of the squares of the differences between the observed and predicted system outputs [13]. Another approach, the instrumental variables method, reduces the bias that can occur in the least squares approach when there is

a presence of noise [14]. These methods were then expanded to include the development of subspace identification methods that consider more advanced system properties [15].

System identification techniques have expanded to accommodate time-varying and nonlinear system dynamics. The most common approach used in identifying linear time-varying (LTV) systems is the local approximation method [16]. Other studies have introduced the use of adaptive filtering algorithms, such as the recursive least squares (RLS) method and the Kalman filtering-based method [17,18]. Finally, in the realm of nonlinear systems, several models and techniques have been proposed. Among these are Dynamic Mode Decomposition [19], Principal Component Analysis [20], and the Hammerstein-Wiener model, a class of nonlinear models that consists of a linear dynamic system sandwiched between two static nonlinearities [21]. Neural networks have also been used in the identification of nonlinear systems due to their capabilities for nonlinear function approximation [22]. Moreover, the development of NARMAX (Nonlinear AutoRegressive Moving Average with eXogenous inputs) models has provided a systematic and structured way to build nonlinear dynamic models [23].

Despite the significant advances made in the system identification domain, there remain a few challenges. For LTI systems, there is still the need to manage problems that arise from over-parameterization, while for LTV systems, the primary challenge lies in dealing with rapidly time-varying systems. Nonlinear system identification, on the other hand, continues to face complexities such as the identification of optimal model structures and parameter estimation.

Classical and modern control theory established a comprehensive base for control of dynamical systems with well-known models, including handling unstable conditions [24–26]. The efforts by Sree and Chidambaram [27] and Skogestad et al. [28] demonstrated the utility of model-based control in managing unstable LTI systems. Their work was revolutionary but revealed the challenge of model accuracy - the effectiveness of the control strategy was dependent on the precision of the system model.

This research was extended to LTV systems, with Taylor et al. [29] applying model-based control to time-varying systems, thus amplifying the complexity of the field. The move

into nonlinear systems by researchers such as de Oliveira et al. [30, 31] introduced further complexities, including the difficulty in developing and maintaining accurate nonlinear models.

Inaccuracies in the system model or unknown environmental uncertainties can lead to a range of problems, such as sub-optimal performance [32], instability [33], and reduced robustness [26, 34]. These challenges have spurred research focusing on learning mechanisms for feedback systems.

3.1.2 Data-Driven Control and Reinforcement Learning

In order to enable active learning for real-world dynamical systems, we need a control algorithm that readily incorporates task information, learns dynamic model representation, and is capable of incorporating policies for solving additional tasks during the learning process.

Recent years have seen major advances in the control of uncertain dynamical systems using reinforcement learning and data-driven approaches; Examples range from allowing robots to perform more sophisticated control tasks such as robotic hand manipulation [35, 36] and aggressive driving [37], to sequential decision making in game domains, e.g. AlphaGo [38] and Atari game playing [39]. In such methods, the system's dynamic is excited in order to obtain the "best" set of measurements that resolve a parameter or the "best-case" mapping (either of the state-control map or of the dynamics). Solution concepts are typically divided into two categories: Model-Based and Model-Free learning for control.

Model-Based Learning

The objective of this approach is to first learn the unknown parameters of the system model from data, then design a controller based on the updated model. Hence, the focus is on analyzing data–accuracy trade-offs.

Addressing the need for online learning within the context of model-based control, Kim and Ng [40] presented an approach for stable adaptive control. Another significant breakthrough in the area of model-based control was introduced by De Persis and Tesi [41], providing formulas for data-driven control in terms of stabilization, optimality, and robustness. This

was augmented by the work of Coulson, Lygeros, and Dörfler [42], and Baros et al. [43], who delved into data-enabled predictive control, a concept that bridged the gap between traditional control systems and the emerging data-driven control systems.

On the data-driven front, Van Waarde et al. [44] introduced the concept of data informativity, providing a new perspective on data-driven analysis and control. This theme was continued by Yu et al. [45], who added to the discourse on the persistence of excitation and controllability in data-driven control, extending the understanding of Willems' Fundamental Lemma.

Berberich et al. [46] further explored data-driven model predictive control, introducing stability and robustness guarantees. This was supported by Lozano et al. [47], who showed the application of robust prediction-based control for unstable delay systems.

Model-Free Learning

In model-free approaches, the agent directly learns the control inputs to optimize a control objective and does not explicitly learn a model of the world. The focus here is on exploration/exploitation type formulations and regret analysis. Since the goal is to learn how to control the system to achieve a specific task, the system is not necessarily fully learned. These algorithms attempt to either 1) estimate value functions (or state-action values) through Monte Carlo simulation, which is then used in some approximate dynamic programming variant [48] or 2) directly optimize a (parameterized) policy. Examples include REINFORCE algorithm, an on-policy gradient-based method [49]. Furthermore, Natural Policy Gradient improves upon REINFORCE by computing an ascent direction that approximately ensures a small change in the policy distribution [50].

One significant contribution in this area came from Fazel et al. [51], who presented a global convergence of policy gradient methods for the linear quadratic regulator. This was further advanced by Alemzadeh and Mesbahi [52], who applied distributed Q-learning to dynamically decoupled systems.

Addressing the complexity of singular systems, Ozcaldiran and Lewis [53] presented

regularizability considerations. This was followed by Vrabie et al. [54], who combined adaptive control with policy iteration in the context of continuous-time linear systems.

Emerging research from Jagtap, Pappas, and Zamani [55] introduced the idea of control barrier functions for unknown nonlinear systems using Gaussian processes. Reinforcement learning was also employed for online control with adversarial disturbances by Agarwal et al. [56].

Lale et al. [57] integrated adaptive control with regret minimization in a linear quadratic Gaussian (LQG) setting, a concept further explored by Cohen, Koren, and Mansour [58] who showed that learning linear-quadratic regulators could be done efficiently with only $\sqrt{T}$ regret. Simchowitz and Foster [59] followed this with their assertion that naive exploration is optimal for online Linear Quadratic Regulator (LQR).

The above literature demonstrated the potential of model-free methods in learning directly from system interaction but also revealed the challenge of slow learning rates. This limitation was particularly evident when the underlying system is unstable or when extending model-free methods to LTV systems and nonlinear systems, as shown by Jagtap et al. [55]. Despite showing promise, model-free methods also highlighted the limitations of computational resource requirements for high-dimensional systems.

Most of the aforementioned model-based and model-free methods are applied in a simulation environment where safety is not an issue, and countless amounts of data are easily accessible by resetting the dynamic to an initial safe region. Real-world applications, however, require safety measures. Additionally, data collection can be expensive or impractical due to resource limitations and safety constraints. Therefore, the objective of Chapter 4 and Chapter 5 is online regulation of unstable systems from a single trajectory without access to reset, in transient time. We provide a data-driven method that guarantees the stability of the synthesized system.

3.2 Deep Neural Networks as Perception Module

Deep learning technology has witnessed rapid advancements, paving the way for its integration into traditional feedback systems. Particularly in automated control and robotics, the fusion of deep neural network (DNN)s with feedback control systems has stirred much interest and extensive research in recent years [60–62]. Policies learned using such methods often rely on a number of hand-engineered components for perception and control, so as to present the policy with a more manageable and low-dimensional representation of observations and actions [63, 64].

The initial stage of this integration saw the use of neural networks primarily for system identification in feedback control systems [60]. However, with the emergence of deep learning, DNNs have begun to play an integral role as a perception model in feedback systems. These models are particularly applicable in complex environments where traditional modeling of the environment is challenging or impractical. The application of DNNs as perception models in autonomous driving is a prime example of this phenomenon. Here, DNNs have been used to interpret raw sensor data, enabling vehicles to understand their environment and make driving decisions [61]. However, the vision system can be complex and prone to errors, and it is typically not improved during policy training nor adapted to the goal of the task.

Levine et al. [62] provided a pioneering effort by integrating deep learning with feedback control in the realm of robotics. The researchers utilized deep learning to map raw image pixels directly to robot actions, thereby enabling the robot to learn a variety of motor skills in real-world environments. This policy, defined as a deep visuomotor policy, ingests RGB images, joint angles, and their velocities and then produces motor torques. The policy was manifested as a deep convolutional neural network, trained with a variant of guided policy search.

Despite the promising advancements, integrating DNNs into feedback systems presents a set of unique challenges. Deep learning models, often perceived as 'black boxes', lack transparency in their inner workings, and are incapable of providing explanations for their

predictions or actions. This lack of interpretability is a significant challenge for control systems, which often necessitate both interpretability and robustness. Furthermore, DNNs are data-driven models that require substantial amounts of training data. Procuring sufficient training data for many feedback control applications can be difficult, expensive, or even impossible.

Nevertheless, the incorporation of DNNs as perception models in feedback systems persists as an active and promising area of research. Chapter 6 is dedicated to analyzing perception in feedback systems, where the underlying dynamics of a system are well understood, and it is instead the integration of perceptual sensor data into the control loop that must be learned.

3.3 Simulation Environments

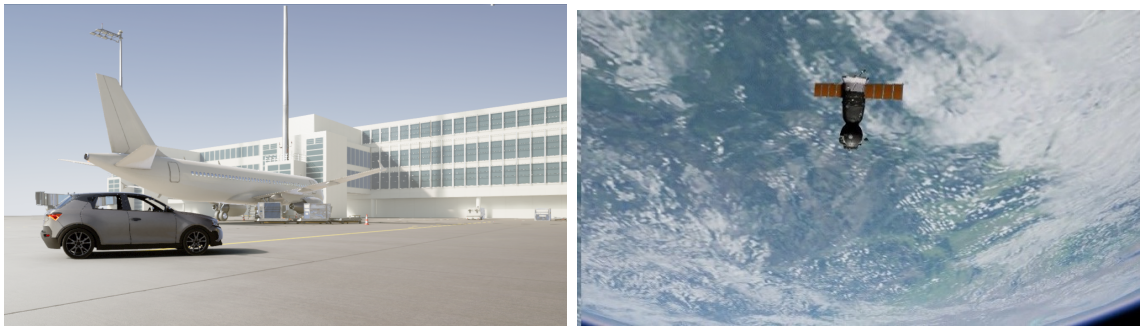
3.3.1 Unreal Engine

To support the development and testing of the CNN based pose estimation algorithms, we utilize Unreal Engine to render a three-dimensional (3D) visualization of the scene based on the involved dynamical system states. The software allows rendering high fidelity photo-realistic scenes by providing the ability to precisely place objects (vehicles, celestial bodies, etc.) in the environment, along with the lighting source. Additionally, we have fine control over the rendering settings and image filters, which allows adding noise and perturbing the observations to enrich the CNN training datasets.

To set up the scene in Unreal Engine, we create a "level", which is a collection of Actors with their associated properties and locations. Figure 3.1a showcases an airport environment with static (airplanes) and dynamic vehicles (cars), whereas Figure 3.1b demonstrates a spacecraft in the space environment with Earth in the background. There are two approaches for using Unreal Engine for simulation:

1. Programming custom game logic and functions to move the actors.

2. Utilizing the scripting interface to kick off a thread that listens to external updates.



(a) Airport environment (b) Space environment

Figure 3.1: Examples of the Unreal Engine environments.

Although the first approach is more user-friendly when used with cross-platform simulators such as Airsim [65], it limits the use of drones and ground vehicles. On the contrary, the second approach can be used together with simulators such as Basilisk [1] for multi-spacecraft simulations in space environments.

3.3.2 Basilisk

Basilisk is a software framework for spacecraft simulations developed by the University of Colorado [66]. The framework has comprehensive features such as faster-than-real-time simulation and hardware-in-the-loop support.

Basilisk was chosen for its ease of use and reconfigurability, as it has a fully-featured Python interface. An entire simulation is executed as a single script, which aids in rapid prototyping. New control laws are simply implemented as new Python modules that Basilisk can integrate internally while having fine-grained control of the framework in Python allows us to expose the state of the simulation in order to integrate external modules such as the CNN and have them run in lockstep with the rest of the simulation. Additionally, we retain the fidelity of dynamics and speed of execution as Basilisk is implemented in C/C++.

Part I

ITERATIVE DATA-DRIVEN CONTROL

Feedback control is ubiquitous in modern technology, including applications that provide stabilization in addition to performance. Stabilization with robustness guarantees is specifically important when dealing with open-loop unstable systems.

Autonomous mobility in safety-critical scenarios dictates the importance of learning from available data for quick adaptation to unexpected changes in the system model or new environmental uncertainties. From a system theoretic perspective, such systems feature an autonomy stack that handles parametric model uncertainties and disturbances with control-theoretic guarantees such as stability and tracking error convergence, common in adaptive control settings. This task is particularly challenging for closed-loop systems with unstable subsystems. Yet, most of the existing synthesis literature has focused on model-based control, where the designer has to discern fundamental limitations stemming from process instabilities. Furthermore, learning-based studies focus mostly on finite-horizon episodic settings, providing finite-time or dynamic regret bounds. In such scenarios, the controller does not necessarily guarantee the stability or robustness of the synthesized system; rather, the goal is to match the performance of the oracle for a well-behaved dynamical model.

The following two chapters are dedicated to constructing a theoretical framework aimed at quantifying the capacity of data-driven finite-time regulation for unstable linear and nonlinear systems. Uniquely, this is undertaken in the absence of an initial data set necessary for managing the associated (unknown) processes. Our objective is to formulate a distinct system-theoretic concept that diverges from traditional constructs that predominantly characterize the system's asymptotic behavior, such as stabilizability and controllability. This approach offers fresh perspectives on system behavior, transcending conventional methodologies.

Chapter 4

Online Regulation of Unstable Linear Systems.

This chapter focuses on online regulation of LTI system states, where the input matrix is known. This point of view has been adopted by the desire to ensure satisfactory performance for online data-guided control based on a single trajectory–even when the underlying system is unstable–from the onset. To realize the above program systematically, we introduce a class of linear systems exhibiting a property called "regularizability"[1]; this notion captures the input "effectiveness" as it relates to finite-time regulation. We then characterize regularizability using linear matrix inequalities (LMIs) and clarify how it relates to the spectral properties of the underlying LTI system. Next, by employing the notion of regularizability, we introduce the Data-Guided Regulation (DGR) algorithm. This online iterative synthesis procedure utilizes a single trajectory for an otherwise partially unknown (discrete time) LTI system. DGR does not use prior assumptions on the linear state dynamics nor access to an initial stabilizing controller or an input-output dataset; instead, the algorithm only relies on the knowledge of the input matrix. Knowledge of the input matrix is motivated by scenarios where it is known how the control input affects the state dynamics, yet the internal system states' interactions are uncertain (for example, consider the problem of controlling an unknown networked system from a given set of nodes). This assumption also proves beneficial for our setup to 1) ensure a satisfactory performance for the system trajectory from the onset of the regulation process, 2) avoid requiring an initial stabilizing controller, and 3) avoid requiring a PE input-output trajectory from an unstable system (that is often impractical and leads to ill-conditioned data matrices for post-processing). We postulate that when the input matrix is also unknown, deriving nontrivial guarantees for the closed-loop performance of unstable systems from the

[1]Not to be confused with the notion of "regularity" for singular systems [53].

onset might prove to be elusive. Finally, as pointed out above, having access to the input structure of a system is pertinent to several applications that involve learning [54, 67–69]; a similar assumption has been adopted for learning and control of nonlinear systems [55], where the system dynamics is affine in control with known input mapping and unknown state dynamics.

The contribution of the proposed work is as follows: (1) in addition to introducing the notion of regularizability for LTI systems, we show how it is distinct from related properties such as stabilizability. We believe that regularizability is of independent interest, particularly as it pertains to online regulation; (2) we derive conditions under which DGR can eliminate unstable modes of the (unknown) system and regulate its state trajectory.[2] DGR aims to simultaneously identify and regulate the hidden unstable modes in a feedback form from a single trajectory. As such, DGR can avoid some of the conditioning issues that arise in processing data generated by an unstable system. Using the notion of regularizability, we then proceed to derive upper bounds on the state trajectories based on a geometric quantity for LTI systems that we refer to as the "instability number;" (3) we show that while DGR performs well for a large class of unstable systems, special structures (e.g., symmetry) further facilitate deriving intuitive bounds on the system trajectory during the learning process; (4) finally, we show that the discrete nature of time-series data enables a recursive approach to DGR synthesis. In this direction, a recursive DGR is proposed that circumvents storing the entire data history and avoids demanding operations such as pseudoinverse computation or multiplying large matrices.

4.1 Review of Past Work

Control of open-loop unstable plants arising, for instance, in industrial and flight control applications, underscores the importance of stabilization with robustness guarantees. As such, controlling unstable systems is an ongoing research topic, particularly in safety-critical systems.

[2]Here, regulation is ensured by bounding the norm of the system states during the learning process; see Section 4.2 for more details.

It is well-known that unstable systems are fundamentally more difficult to control [25]; in fact, practical closed-loop systems with unstable subsystems are only locally stable [27]. Yet, most of the existing synthesis literature has focused on model-based control where the designer has to discern fundamental limitations stemming from process instabilities [28].

Recent interest in model-free stabilization, in the meantime, has been motivated by novel sensing technologies, robust machine learning, and efficient computational methods to reason about control and estimation of uncertain systems–all from measured (online) data [70, 71]. Safety-critical systems have necessitated non-asymptotic analysis on data-driven methods [72, 73]. In particular, there has been a growing interest in examining finite-time control of unknown linear dynamical systems from time-series or a single trajectory [67, 74–79]. Parallel to asymptotic analysis in traditional adaptive control and system identification (sysID) [34, 80, 81], model-based finite-time control has benefited from a least-squares approach to identification followed by robust synthesis–see for example [82]. In this direction, probabilistic bounds on the estimation error related to the required run-time have been examined. While it has been shown that model-based methods require fewer measurements for specific control problems in general [83],[3] data collection required for sysID can be expensive or impractical due to resource limitations and safety constraints. Furthermore, some of the studies mentioned above rely on *a priori* information about the system, such as estimates of system parameters [73], an initial stabilizing controller [40, 51, 52, 84], or assuming an open-loop stable system [75, 77].

It is known that an input-output trajectory of a controllable linear time-invariant (LTI) system can be parameterized by (offline) data trajectories generated from a persistently exciting (PE) input [85]. Building on this fact, there have been recent works on the stabilization of LTI systems directly from the available data (e.g., see [41–46]). However, ensuring a persistently exciting (PE) input-output data may not be practical for data-driven control or identification of unstable systems even in low dimensions without recourse to resets [41].

[3]That is, first finding a model estimate from data and then using that estimate for control design.

For instance, injecting white noise into an unstable system can result in ill-conditioned data matrices that, in turn, lead to numerical issues. Hence, existing data-guided methods might not be directly applicable for safety-critical control such as online flight control [47] or infrastructure recovery [86]. The work presented in this chapter is motivated by such applications, requiring no reliance on an initial stabilizing controller nor a PE input-output data trajectory for data-guided control.

4.2 Unstable Linear Systems

Consider a discrete-time LTI model of the form,

$$\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t, \qquad \boldsymbol{x}_0 \text{ given}, \tag{4.1}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the system parameters and $\boldsymbol{x}_t \in \mathbb{R}^n$ and $\boldsymbol{u}_t \in \mathbb{R}^m$ denote the state and control inputs at time index t, respectively. We assume that the system matrix A is unknown and (possibly) unstable, and that the input matrix B is known.

Considering regulation by having access to the input matrix is of interest in applications where it is known a priori how various control inputs affect the dynamic states, e.g., how the elevator deflection affects the aircraft pitch dynamics or how specific boundary nodes can influence a diffusive network. Intuitively, this assumption allows an online regulation mechanism to have a chance of stabilizing an unknown (and possibly) unstable system in real time from the onset of the learning process.

The following example motivates our setup and underscores why the data-guided perspective requires introducing new system theoretic notions.

Example 4.1. For any positive integer n, define the system matrix $A \in \mathbb{R}^{n\times n}$ and the input

matrix $B \in \mathbb{R}^n$ as,

$$A = \begin{pmatrix} \lambda_1 & 1 & 0 & \dots & 0 \\ 0 & \lambda_2 & 1 & & \vdots \\ 0 & 0 & \lambda_3 & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \dots & & 0 & \lambda_n \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

Note that for any choice of $\lambda_i \in \mathbb{R}$, the pair (A, B) is controllable (and therefore stabilizable). Furthermore, since the set $\{\lambda_i\}$ coincides with the spectrum of A, if any subset of $\{\lambda_i\}$ are equal, then A contains the corresponding Jordan block. Moreover, when $\lambda_i \neq \lambda_j$ $(i \neq j)$, then A is diagonalizable. Let $\boldsymbol{x}_0 = \boldsymbol{e}_1$ and observe that under (4.1), we have $\boldsymbol{e}_1^\intercal \boldsymbol{x}_t = \lambda_1^t$ for all $0 \leq t < n$ regardless of the input $\boldsymbol{u}_t$. This implies that, for "any" choice of input, for the first n iterations, the first state of the system grows exponentially fast with the rate λ_1 whenever $|\lambda_1| > 1$.

Remark 4.1. Example 4.1 constructs a family of controllable systems where no controller can regulate their respective first states–at least for the first n iterations. That is, a system state will grow exponentially fast regardless of the choice of $\boldsymbol{u}_t$, even when all eigenvalues of A except λ_1 are stable (e.g., $|\lambda_i| < 1$ for $i = 2, \cdots, n$). Note that in this example, the (right) eigenvector associated with the unstable mode of A (i.e., the eigen-pair $(\lambda_1, \boldsymbol{e}_1)$) is orthogonal to $\mathcal{R}(B) = \mathcal{R}(\boldsymbol{e}_n)$. This is despite the fact that the Popov-Belevitch-Hautus (PBH) controllability test holds (i.e., for any left eigenvector $\boldsymbol{v}$ of A, we have $\boldsymbol{v}^\intercal B \neq 0$). This example highlights that the controllability of a pair (A, B) does not capture the "regularizability" of an unstable linear system, especially when closed-loop regulation has to be achieved in a data-guided manner from the onset of the learning process. Finally, we point out that in the particular case when $\lambda_i = 0$ for $i = 2, \dots, n$, the controllability matrix corresponding to (A, B) is anti-diagonal with all anti-diagonal elements equal to identity. Therefore, it has singular values/eigenvalues equal to ± 1. This implies that the controllability matrix has a condition number equal to identity, as the controllability matrix does not distinguish

difficult-to-regularize modes.

To formalize the behavior of the class of systems mentioned above, we introduce a system theoretic notion that captures the effectiveness of the input as pertinent to online regulation. To motivate this notion, note that the dynamics in (4.1) can be represented as,

$$\begin{aligned} \boldsymbol{x}_{t+1} = &\ \Pi_{\mathcal{R}(B)^\perp} A\boldsymbol{x}_t + \ \Pi_{\mathcal{R}(B)} A\boldsymbol{x}_t + B\boldsymbol{u}_t \\ = &\ \Pi_{\mathcal{R}(B)^\perp} A\boldsymbol{x}_t + B(B^\dagger A\boldsymbol{x}_t + \boldsymbol{u}_t). \end{aligned}$$

Setting $\boldsymbol{u}_t = -B^\dagger A\boldsymbol{x}_t + \bar{\boldsymbol{u}}_t$, (4.1) can be rewritten as $\boldsymbol{x}_{t+1} = \widetilde{A}\boldsymbol{x}_t + B\bar{\boldsymbol{u}}_t$ where,

$$\widetilde{A} := \ \Pi_{\mathcal{R}(B)^\perp} A, \tag{4.2}$$

and $\bar{\boldsymbol{u}}_t$ is yet to be designed. Note that the signals $\widetilde{A}\boldsymbol{x}_t$ and $B\bar{\boldsymbol{u}}_t$ are now orthogonal. This implies that the control signal would not directly affect the part of dynamics that is generated by $\Pi_{\mathcal{R}(B)^\perp} A$. As such, in order to have even the possibility of achieving "some" online performance for this system in finite time, we require that this part of the dynamics be stable. This observation thereby motivates the following definition.

Definition 4.1. The pair (A, B) is called *regularizable* if $\widetilde{A} := \ \Pi_{\mathcal{R}(B)^\perp} A$ is Schur stable.

As we will show subsequently, regularizability of a pair (A, B) is related to the stabilizability of (A, B) as well as detectability of $(A, B^\intercal)$; a combination that is not typically encountered in LTI analysis. This connection is intuitive, as regulation of a system in finite time requires the states to be accessible (for control and observation) through the input matrix B. Regularizability also facilitates a new perspective on LTI systems, providing a basis for the analysis of online algorithms such as the one proposed in Section 4.4.

4.3 Regularizable Systems

In order to get a better sense of the notion of regularizability, we study the spectral properties of $\widetilde{A}$ in (4.2) and its relation with system matrices A and B. First, the following example highlights why regularizability of a system is distinct from its controllability.

Example 4.2. Consider the linear system with A defined as in Example 4.1 such that $|\lambda_1| > 1$ and $|\lambda_i| < 1$ for $i = 2, \ldots, n$. Note that the pair $(A, \boldsymbol{e}_n)$ is controllable (and thus stabilizable); however, this pair is not regularizable. On the other hand, the pair $(A, \boldsymbol{e}_1)$ is regularizable but not controllable.

Recall that a pair (A, B) is stabilizable if and only if $(A^\intercal, B^\intercal)$ is detectable. The detectability of $(A, B^\intercal)$ is seldom of interest in linear system theory [24]; however, we show that it is, indeed, a necessary condition for (A, B) to be regularizable. To this end, we first connect regularizability to the spectral properties of the pair (A, B).

Lemma 4.1. *Let* $\widetilde{A} = \Pi_{\mathcal{R}(B)^\perp} A$. *Then for each right eigenpair* $(\lambda, \boldsymbol{v})$ *of* A *the following holds:*

- $(\lambda, \boldsymbol{v})$ *is a right eigenpair of* $\widetilde{A}$ *whenever* $\boldsymbol{v} \in \mathcal{R}(B)^\perp$ *or* $\lambda = 0$.
- $(0, \boldsymbol{v})$ *is a right eigenpair of* $\widetilde{A}$ *whenever* $\boldsymbol{v} \in \mathcal{R}(B)$.

The proof of Lemma 4.1 directly follows from the definitions and therefore is omitted. Note that the above lemma does not address the scenario where $(\lambda, \boldsymbol{v})$ is an eigenpair of A, with $\lambda \neq 0$, and $\boldsymbol{v} = \boldsymbol{v}_1 + \boldsymbol{v}_2$, with nontrivial $\boldsymbol{v}_1 \in \mathcal{R}(B)$ and $\boldsymbol{v}_2 \in \mathcal{R}(B)^\perp$.

The following example illustrates that $\widetilde{A}$, as a product of matrix A with an orthogonal projection operator, has a spectral radius distinct from A.

Example 4.3. Consider the system in Example 4.1, where the identity off-diagonal elements of A are replaced with 10, $\lambda_1 = 0.9$ and $\lambda_i = 0$ for all $i = 2, \ldots, n$, and $B = \mathbb{1}$. It is straightforward to show that for all $n \geq 2$, A is Schur stable with a spectral radius of 0.9 while $\widetilde{A}$ is not, i.e., (A, B) is not regularizable. In this case, although A is Schur stable, its operator norm is about 10. Furthermore, the spectral radius of $\widetilde{A}$ would be 4.55 for $n = 2$ and increases to about 10 as n increases. This results in pathological behavior despite the fact that the system is originally stable, e.g., any infinite horizon closed-loop LQR controller for this system would demonstrate undesirable behavior —similar to Example 4.1— when

initialized from $\boldsymbol{x}_0 = \mathbb{1}$.[4]. Finally, it is worth noting that the controllability matrix of this pair is ill-conditioned in contrast to Example 4.1.

The preceding discussion exemplifies that even for a stable system, it is nontrivial to assert that state trajectories over a finite time horizon are "well-regulated." It is no surprise then that, in spite of its severe limitations from a system theoretic perspective, most of the recent works on data-guided control focus on *contractible* systems as they streamline composition rules and analysis for consecutive iterations in a learning algorithm [56, 57]. However, the succeeding remark shows why regularizability, as introduced in this work, is less restrictive, and thus—by replacing contractility—can mitigate those system theoretic limitations.

Remark 4.2. A pair (A, B) is said to be contractible if there exists a controller K such that $\|A - BK\| < 1$. Noting that

$$A - BK = \widetilde{A} + \ \Pi_{\mathcal{R}(B)}(A - BK),$$

for any vector $\boldsymbol{x} \in \mathbb{R}^n$, (by orthogonality) it follows that,

$$\begin{aligned} \|\widetilde{A}\boldsymbol{x}\|^2 &= \|(A - BK)\boldsymbol{x}\|^2 - \| \ \Pi_{\mathcal{R}(B)}(A - BK)\boldsymbol{x}\|^2 \\ &\leq \|(A - BK)\|^2 \, \|\boldsymbol{x}\|^2. \end{aligned}$$

This, in turn, implies that a contractible system is regularizable (as in that case $\|\widetilde{A}\| < 1$). In particular, if the original system matrix A is non-expansive (at least on the subspace $A^{-1}\{\mathcal{R}(B)^{\perp}\}$), then (A, B) is regularizable.

The following results further clarify the relation between regularizable systems and their system theoretic twins.

Proposition 4.2. *If (A, B) is regularizable, then*

- *(A, B) is stabilizable, and*

[4]One practical remedy to this problem is to split the dynamics into multiple time-scales using, say, a sampling heuristics [87]. However, time-scale separation often requires physical insights and is non-trivial to identify for general systems [88], let alone for a system with unknown dynamics.

- $(A, B^\intercal)$ *is detectable.*

Proof. For the first claim, note that $\widetilde{A} = A - \ \Pi_{\mathcal{R}(B)}A = A + BK$, where $K := -B^\dagger A$. Thus if (A, B) is regularizable, then K is a stabilizing closed-loop controller. For the second claim, we establish a contrapositive. Suppose that $(A, B^\intercal)$ is not detectable. Hence there exists a right eigenpair $(\lambda, \boldsymbol{v})$ of A, where $|\lambda| \geq 1$ and $\boldsymbol{v} \in \mathcal{N}(B^\intercal) = \mathcal{R}(B)^\perp$. Then, Lemma 4.1 implies that $(\lambda, \boldsymbol{v})$ must be a right eigenpair of $\widetilde{A}$. Since $|\lambda| \geq 1$, $\widetilde{A}$ is not Schur stable and therefore (A, B) is not regularizable.

□

Note that the consequences of Proposition 4.2 are equivalent whenever A is symmetric, as detectability of $(A, B^\intercal)$ is equivalent to stabilizability of $(A^\intercal, B)$. Also, note that Proposition 4.2 provides a necessary condition for regularizability, whereas the following counter-example underscores why the stabilizability of (A, B), even when combined with detectability of $(A, B^\intercal)$, is not sufficient.

Example 4.4. Let the system matrices A, B be defined as in Example 4.1 and consider the pair $(A_1, B_1) := (A + A^\intercal, B)$. By the structure of A_1, note that (A_1, B_1) is controllable. Since A_1 is symmetric, $(A_1, B_1^\intercal)$ is also observable. By direct computation, we observe that,

$$\widetilde{A} = \ \Pi_{\mathcal{R}(B)^\perp}A = \begin{pmatrix} 2\lambda_1 & 1 & 0 & \dots & 0 \\ 1 & 2\lambda_2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & 2\lambda_{n-1} & 1 \\ 0 & \dots & 0 & 0 & 0 \end{pmatrix}.$$

Now if any of λ_i's, for $i = 1, \dots, n-1$, is say, larger than $1/2$, then $\widetilde{A}$ would be unstable, implying that (A_1, B_1) is not regularizable.

In order to complete our understanding of regularizability, we provide several characterizations using linear matrix inequalities (LMI).

Proposition 4.3. *Consider a pair (A, B), and denote $\Pi_\perp := \Pi_{\mathcal{R}(B)^\perp}$. Then the following are equivalent:*

(i) The pair (A, B) is regularizable.

(ii) $\exists P \succ 0$ such that $\rho(A^\intercal\ \Pi_\perp P\ \Pi_\perp A P^{-1}) < 1$.

(iii) $\exists P \succ 0$ such that $\|P^{1/2}\ \Pi_\perp A P^{-1/2}\| < 1$.

(iv) $\exists P \succ 0$ such that $A^\intercal\ \Pi_\perp P\ \Pi_\perp A - P \prec 0$.

(v) $\exists W \succ 0$ such that $\begin{pmatrix} W & \Pi_\perp A W \\ W A^\intercal\ \Pi_\perp & W \end{pmatrix} \succ 0$.

(vi) $\exists P \succ 0$ and $G \in \mathbb{R}^{n\times n}$ such that,

$$\begin{pmatrix} P & A^\intercal\ \Pi_\perp G^\intercal \\ G\ \Pi_\perp A & G + G^\intercal - P \end{pmatrix} \succ 0.$$

(vii) $\exists P \succ 0$, and $G, H \in \mathbb{R}^{n\times n}$ such that,

$$\begin{pmatrix} GA + A^\intercal G^\intercal - P & A^\intercal H^\intercal - G \\ HA - G^\intercal & \Pi_\perp P\ \Pi_\perp - H - H^\intercal \end{pmatrix} \prec 0.$$

Proof. Noting that regularizability of (A, B) is equivalent to Schur stability of $\Pi_\perp A$, the first four equivalences are direct consequences of Theorem 7.7.7 in [89]. By using Schur complements and constructing a congruence induced by $\mathrm{diag}(I, P^{-1})$, (iv) becomes equivalent to (v). The last two equivalences are due to Theorem 1 in [30] and Theorem 1 in [31], respectively. □

We conclude this section by providing a sufficient condition for guaranteeing when a polytopic uncertain LTI system is regularizable.

Proposition 4.4. *Consider $A_i \in \mathbb{R}^{n\times n}$ for $i = 1, \ldots, N$ and suppose there exist matrices $P_i \succ 0$ and $G, H \in \mathbb{R}^{n\times n}$ satisfying,*

$$\begin{pmatrix} GA_i + A_i^\intercal G^\intercal - P_i & A_i^\intercal H^\intercal - G \\ HA_i - G^\intercal & \Pi_S P_i\, \Pi_S - H - H^\intercal \end{pmatrix} \prec 0,$$

for some linear subspace $S \subseteq \mathbb{R}^n$. Then a pair (A, B) is regularizable whenever $A \in$ convhull$\{A_i\}_1^N$ and

$$\begin{pmatrix} P_i & P_i\, \Pi_{\mathcal{R}(B)^\perp} \\ \Pi_{\mathcal{R}(B)^\perp} P_i & \Pi_S P_i\, \Pi_S \end{pmatrix} \succeq 0, \quad \forall i = 1, \ldots, N.$$

Proof. Since $A \in$ convhull$\{A_i\}_1^N$, there exists scalars $\alpha_i \in [0, 1]$ with $\sum_1^N \alpha_i = 1$ such that $A = \sum_1^N \alpha_i A_i$. By defining $P = \sum_1^N \alpha_i P_i$ and taking the convex combinations of the negative definite matrices in the hypothesis with weights α_i we obtain,

$$\begin{pmatrix} GA + A^\intercal G^\intercal - P & A^\intercal H^\intercal - G \\ HA - G^\intercal & \Pi_S P\, \Pi_S - H - H^\intercal \end{pmatrix} \prec 0. \tag{4.3}$$

Now by taking the Schur complement of the LMI in the hypothesis involving the input matrix B it follows that,

$$\Pi_S P_i\, \Pi_S \succeq \Pi_{\mathcal{R}(B)^\perp} P_i\, \Pi_{\mathcal{R}(B)^\perp}, \quad \forall i = 1, \ldots, N.$$

Convex combinations of this LMIs with the same coefficients lead to,

$$\Pi_S P\, \Pi_S \succeq\ \Pi_{\mathcal{R}(B)^\perp} P\, \Pi_{\mathcal{R}(B)^\perp}.$$

This, together with the LMI in (4.3) imply the LMI in Proposition 4.3.*(vii)*. As $P \succ 0$, we conclude that the pair (A, B) is regularizable. □

Remark 4.3. Note that the proof above also shows that the last LMI in the statement of Proposition 4.4 is equivalent to

$$\Pi_S P_i\, \Pi_S \succeq\ \Pi_{\mathcal{R}(B)^\perp} P_i\, \Pi_{\mathcal{R}(B)^\perp}, \quad \forall i = 1, \ldots, N; \tag{4.4}$$

which is certainly satisfied when $S = \mathcal{R}(B)^{\perp}$. Thus, a direct consequence of Proposition 4.4–together with the characterization in Proposition 4.3. (vii)–is as follows: if there exists an input matrix B such that (A_i, B) is regularizable for each $i = 1, \ldots, N$, then we can conclude that (A, B) is regularizable for any (unknown) matrix $A \in \text{convhull}\{A_i\}_1^N$. This observation does not follow directly from the definition as the spectral radius is not sub-additive. Moreover, Proposition 4.4 provides the flexibility of working with the linear subspace S independently of $\mathcal{R}(B)$, which proves to be useful for design purposes, e.g., devising an input matrix in order to make a polytopic uncertain system regularizable.

Finally, Proposition 4.4–in view of (4.4)–implies that regularizability is a monotonic system theoretic property with respect to the input, in the sense that enlarging $\mathcal{R}(B)$ would not destroy its regularizability. In fact, "enlarging" $\mathcal{R}(B)$ for a system would make it "more" regularizable (as $\rho(\widetilde{A})$ will be smaller).

4.4 Data-Guided Regulation (DGR) Algorithm

The primary focus of this section is devising an online, data-driven feedback controller to regulate the system's state trajectories, quantified in terms of a signal norm. In this direction, we propose an iterative procedure for updating the feedback gain (policy); the form of the controller can be motivated by considering, at each iteration t, the following optimization problem with a "one-step quadratic cost",[5]

$$\begin{aligned} \min_{\boldsymbol{u}_t} \quad & \|\boldsymbol{x}_{t+1}\|^2 + \alpha \|\boldsymbol{u}_t\|^2 \\ \text{s.t.} \quad & \boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t, \end{aligned} \tag{4.5}$$

where $\boldsymbol{x}_t$ is measured over time but the system matrix A is unknown, and $\alpha \geq 0$ is a regularization factor for the controller design.[6] In the case of known A, it is straightforward to

[5]The setup resembles dead-beat control design, with the caveat that the synthesis is data-guided.

[6]We note that considering a more elaborate form of cost (e.g., finite/infinite horizon LQR cost) for this optimization problem is certainly relevant. However, in this specific problem setup, i.e., no prior knowledge of the matrix A and absence of any prior input-state data, we have observed no significant numerical advantage in considering a more elaborate cost–particularly for upper bounding the state trajectories from the onset of the learning process.

characterize the set of minimizers of the above optimization problem through the first-order optimality condition,

$$(\alpha I_m + B^\intercal B)\boldsymbol{u}_t + B^\intercal A\boldsymbol{x}_t = 0;$$

as such, the corresponding input belongs to a linear subspace in $\mathbb{R}^m$ parameterized by the system matrices and data. The following proposition illustrates why regularizability, as presented in Section 4.3, is pertinent to the online regulation of LTI systems.

Proposition 4.5. *For every $\alpha \in [0, \varepsilon)$, with some small enough $\varepsilon > 0$, the minimum norm solution of the iterative optimization (4.5) stabilizes the system (4.1) if and only if the pair (A, B) is regularizable.*

Proof. Given a fixed $\alpha \geq 0$, the minimum norm solution to (4.5) at iteration t is $\boldsymbol{u}_t^* = -G_\alpha A\boldsymbol{x}_t$, where $G_\alpha \coloneqq (\alpha I + B^\intercal B)^\dagger B^\intercal$. Therefore, this iterative solution stabilizes the system in (4.1) if and only if $A - BG_\alpha A$ is Schur stable. Using properties of the pseudoinverse, $A - BG_0A = (I - BB^\dagger(BB^\dagger)^\intercal)A = (I - BB^\dagger)A = \widetilde{A}$, where $\widetilde{A}$ is as defined in Definition 4.1. The proof now follows by continuity of the spectral radius with respect to α. □

Note that $G_\alpha \to 0$ as $\alpha \to \infty$, implying that $\boldsymbol{u}_t \to 0$ for all t. As such, in general, the solution to (4.5) is stabilizing when α is small enough. The formulation of the optimization problem (4.5) requires the knowledge of system parameters; nonetheless, it forms the basis for the proposed algorithm when A is unknown and potentially unstable. The corresponding synthesis procedure is detailed in Algorithm 1. Specifically, for any $\alpha \geq 0$, at iteration t, DGR sets

$$\boldsymbol{u}_t^* = -K_t^*\boldsymbol{x}_t, \qquad K_t^* \coloneqq G_\alpha \mathcal{Y}_t \mathcal{X}_{t-1}^\dagger, \tag{4.6}$$

where $G_\alpha \coloneqq (\alpha I + B^\intercal B)^\dagger B^\intercal$ and $\mathcal{X}_{t-1}, \mathcal{Y}_t \in \mathbb{R}^{n\times t}$ are the measured data matrices,

$$\mathcal{X}_{t-1} \coloneqq \begin{pmatrix} \boldsymbol{x}_0 & \dots & \boldsymbol{x}_{t-1} \end{pmatrix},$$
$$\mathcal{Y}_t \coloneqq \begin{pmatrix} \boldsymbol{x}_1 - B\boldsymbol{u}_0 & \dots & \boldsymbol{x}_t - B\boldsymbol{u}_{t-1} \end{pmatrix}.$$

Algorithm 1 Data-Guided Regulation (DGR)

1: **Initialization** (at $t = 0$)
2: Measure $\boldsymbol{x}_0$; set $K_0 = \mathbf{0}$, $G_\alpha = (\alpha I + B^\intercal B)^\dagger B^\intercal$
3: Set $\mathcal{X}_0 = \begin{pmatrix} \boldsymbol{x}_0 \end{pmatrix}$ and $\mathcal{Y}_0 = \begin{pmatrix} \end{pmatrix}$
4: **While stopping criterion not met**[7]
5: Compute $\boldsymbol{u}_t = -K_t \boldsymbol{x}_t$
6: Run system (4.1) and measure $\boldsymbol{x}_{t+1}$
7: Update $\mathcal{Y}_{t+1} = \begin{pmatrix} \mathcal{Y}_t & \boldsymbol{x}_{t+1} - B\boldsymbol{u}_t \end{pmatrix}$
8: $K_{t+1} = G_\alpha \mathcal{Y}_{t+1} \mathcal{X}_t^\dagger$
9: $\mathcal{X}_{t+1} = \begin{pmatrix} \mathcal{X}_t & \boldsymbol{x}_{t+1} \end{pmatrix}$
10: $t = t + 1$

Intuitively, collecting more data results in capturing the essential (e.g., unstable) modes in the dynamics. As such, it is important to note that DGR is particularly relevant for online regulation of unstable systems, when the controller does not have access to enough state data for the purpose of identification or stabilization.

The proposed technique is close in spirit to modal analysis, where regression-based methods are leveraged to extract and control the dominant modes of the system [90, 91]. The emphasis of DGR, however, is on the significance of each temporal action for safety-critical applications; in these scenarios, it might be rather unrealistic to generate sufficient data from the inherently unstable modes.

From an implementation perspective, the DGR algorithm can become computationally expensive for large-scale systems. This is primary due to steps 7-9 of Algorithm 1, where the entire temporal data is stored in $\mathcal{X}_{t+1}$ and $\mathcal{Y}_{t+1}$; the pseudoinverse operation, in the meantime, has complexity $\mathcal{O}(n^2 t)$ required at iteration t. While for the purpose of analysis, we present the basic form of DGR (as in Algorithm 1), in Section 4.5 we will propose Fast

[7]The stopping criterion can be application-specific. For instance, for sysID, generating n linearly independent data is sufficient, while mere stabilization may require less; see [44].

Data-Guided Regulation (F-DGR) to circumvent the complexity of storing and computing on large datasets using a rank-one update on the data matrices, resulting in a recursive evaluation of $\mathcal{Y}_{t+1}\mathcal{X}_t^\dagger$ (see Algorithm 2).

4.4.1 Analysis of DGR

In this subsection, we provide the performance analysis for DGR in the general setting; as pointed out previously, DGR is particularly relevant when $t \leq n$, where n denotes the dimension of the underlying system. We examine the effects of DGR on the system's state trajectory and deduce adequate guarantees in terms of norm upper-bound and informativity of generated data. In addition, we will see how a particular structure of the system matrix A, such as $\mathcal{R}(A) \subset \mathcal{R}(B)$ or its diagonalizability, facilitates further insights into the operation of DGR as presented in the next subsection.

First, we show why regularizability is essential for the analysis of the trajectory generated under Algorithm 1; in hindsight, justifying its introduction in the first place.

Lemma 4.6. *For all $t > 0$, the trajectory generated by Algorithm 1 satisfies,*

$$\boldsymbol{x}_{t+1} = \ \Pi_{\mathcal{R}(B)^\perp} A\boldsymbol{x}_t + \ \Pi_{\mathcal{R}(B)}\ A\boldsymbol{z}_t + \Delta_\alpha \boldsymbol{w}_t,$$

where $\Delta_\alpha \coloneqq B(B^\dagger - G_\alpha)A$, $\boldsymbol{z}_0 \coloneqq \boldsymbol{x}_0$, $\boldsymbol{w}_0 = 0$, and $\boldsymbol{z}_t \coloneqq \ \Pi_{\mathcal{R}(\mathcal{X}_{t-1})^\perp}\boldsymbol{x}_t$ and $\boldsymbol{w}_t \coloneqq \ \Pi_{\mathcal{R}(\mathcal{X}_{t-1})}\boldsymbol{x}_t$ for $t > 0$. Furthermore, $\{\boldsymbol{z}_0, \boldsymbol{z}_1, \cdots, \boldsymbol{z}_t\}$ is a set of "orthogonal" vectors (possibly including the zero vector), and $\Delta_0 = 0$.

Proof. Let $B = U_r\Sigma_r V_r^\intercal$ be the "thin" SVD of B, where $r = \text{rank}(B)$. Since $BB^\dagger = U_r U_r^\intercal = \Pi_{\mathcal{R}(U_r)}$,

$$\begin{aligned}
\boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B\boldsymbol{u}_t \\
&= \left[A - BG_\alpha \mathcal{Y}_t \mathcal{X}_{t-1}^\dagger\right]\boldsymbol{x}_t \\
&= \left[A - \ BG_\alpha\ A\ \ \Pi_{\mathcal{R}(\mathcal{X}_{t-1})}\right]\boldsymbol{x}_t \\
&= \ \Pi_{\mathcal{R}(U_r)^\perp} A\boldsymbol{x}_t + BB^\dagger A(\boldsymbol{z}_t + \boldsymbol{w}_t) - BG_\alpha\ A\ \boldsymbol{w}_t \\
&= \ \Pi_{\mathcal{R}(U_r)^\perp} A\boldsymbol{x}_t + \ \Pi_{\mathcal{R}(U_r)}\ A\boldsymbol{z}_t + B(B^\dagger - G_\alpha)A\boldsymbol{w}_t.
\end{aligned}$$

Thus, the first claim follows as $\mathcal{R}(U_r) = \mathcal{R}(B)$. For the second claim, note that the definition of $\boldsymbol{z}_t$ implies that $\boldsymbol{z}_t \perp \mathcal{R}(\mathcal{X}_{t-1})$ for all $t > 0$, and $\boldsymbol{z}_k \in \mathcal{R}(\mathcal{X}_{t-1})$ for all $k = 1, \ldots, t-1$ and all $t > 0$. Hence $\{\boldsymbol{z}_0, \boldsymbol{z}_1, \cdots, \boldsymbol{z}_t\}$ consists of orthogonal vectors. Finally, $\Delta_0 = 0$ follows by the definition of G_α and properties of pseudoinverse. □

The preceding lemma implies that in the case of $\alpha = 0$, the time series generated by Algorithm 1 can be considered as the trajectory of a linear system with parameters $(\widetilde{A}, \widetilde{B})$ and "input" $\boldsymbol{z}_t$ where,

$$\widetilde{A} \coloneqq \Pi_{\mathcal{R}(B)^\perp} A, \qquad \widetilde{B} \coloneqq \Pi_{\mathcal{R}(B)} A, \tag{4.7}$$

and $\boldsymbol{z}_t = \widetilde{K}_t \boldsymbol{x}_t$, with the time-varying, state-dependent feedback gain $\widetilde{K}_t = \Pi_{\mathcal{R}(\mathcal{X}_{t-1})^\perp}$. Note that, in this case, if $\Pi_{\mathcal{R}(B)}$ and A commute,[8] then $\widetilde{A}\widetilde{B} = 0$ and $\boldsymbol{x}_{t+1} = \widetilde{A}^{t+1}\boldsymbol{x}_0 + \widetilde{B}\boldsymbol{z}_t$. Moreover, $\widetilde{A} = 0$ whenever $\mathcal{R}(A) \subset \mathcal{R}(B)$, i.e., the system dynamics will only be driven by the feedback signal $\boldsymbol{z}_t$; these cases will be examined further subsequently. In case of general α, the system trajectories evolve as,

$$\boldsymbol{x}_{t+1} = \widetilde{A}^{t+1}\boldsymbol{x}_0 + \textstyle\sum_{r=0}^{t} \widetilde{A}^{t-r}\big[\widetilde{B}\boldsymbol{z}_r + \Delta_\alpha \boldsymbol{w}_r\big]. \tag{4.8}$$

Finally, an attractive feature of DGR hinges upon the orthogonality of the "hidden" states $\boldsymbol{z}_t$ generated during the process.

Bounding the State Trajectories Generated by DGR

In the open loop setting, the generated data from an unstable system can grow exponentially fast with a rate dictated by the largest unstable mode. We show that DGR can prevent this undesirable phenomenon for unstable systems when the system is regularizable. The key property for such an analysis involves the notion of instability number.

[8]This is the case if (and only if) both matrices are simultaneously diagonalizable (Theorem 1.3.21 in [89]). If A is symmetric, then these matrices commute if (and only if) they are congruent (Theorem 4.5.15 in [89]).

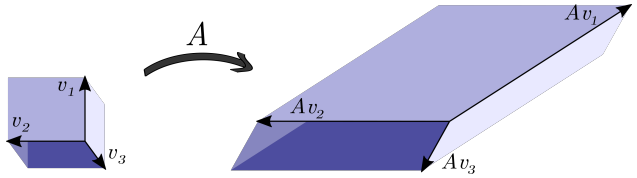


Figure 4.1: A unit cube in the domain of A that is mapped to a parallelepiped in its range space.

Definition 4.2. Given the matrix $A \in \mathbb{R}^{n\times n}$, for any positive integer $t \leq n$, its *instability number of order* t is defined as,

$$M_t(A) \coloneqq \sup_{\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_t\}\in\mathcal{O}_t^n} \|A\boldsymbol{v}_1\|\,\|A\boldsymbol{v}_2\| \cdots \|A\boldsymbol{v}_t\|,$$

where $\mathcal{O}_t^n$ is the collection of all sets of t "orthonormal" vectors in $\mathbb{R}^n$; for $t > n$ we define $M_t(A) = 0$.

Note that $M_t(A) \leq \|A\|^t$ for all t, where $\|.\|$ denotes the induced operator norm. However the behavior of $M_t(A)$ is fundamentally distinct from $\|A\|^t$. In fact, the instability number of a matrix is distinct from products of any subset of its eigenvalues. Consider, for example, a t-dimensional hypercube with its image under A as a parallelotope (see Figure 4.1 for a 3D schematic). The instability number is related to the multiplication of the lengths of edges radiating from one vertex of the parallelotope, while $\det(A^\intercal A)$ is related to its volume. The instability number of a matrix can in fact be difficult to compute. In what follows, we first provide upper and lower bounds on $M_t(A)$ characterizing its growth rate with respect to the largest singular value of A. Subsequently, these bounds will be used to provide a bound on the norm of the state trajectory generated by DGR.

Lemma 4.7. *Let $\sigma_1, \cdots, \sigma_n$ denote the singular values of $A \in \mathbb{R}^{n\times n}$ in a descending order. Then for $t \leq n$,*

$$\left[\frac{\sigma_1^2}{t}\right]^t \leq M_t^2(A) \leq \left[\frac{\sigma_1^2}{t}\right]^t + \sum_{j=1}^{t-1}\left[\frac{\sigma_1^2}{t-j}\right]^{t-j}\binom{t}{j}\delta^j + \delta^t,$$

where $\delta \coloneqq \sum_{i=2}^t \sigma_i^2$, with $M_t(A)$ as defined in Definition 4.2.

Proof. Let $A = W\Sigma U^\intercal$ be the Singular Value Decomposition (SVD) of A where Σ is diagonal containing the singular values in descending order and both $W, U \in \mathbb{R}^{n\times n}$ are unitary. This implies that,

$$\begin{aligned} M_t(A) &= \sup_{\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^n} \ \|\Sigma U^\intercal \boldsymbol{v}_1\| \, \|\Sigma U^\intercal \boldsymbol{v}_2\| \ \cdots \ \|\Sigma U^\intercal \boldsymbol{v}_t\| \\ &= \sup_{\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^n} \ \|\Sigma \boldsymbol{v}_1\| \, \|\Sigma \boldsymbol{v}_2\| \ \cdots \ \|\Sigma \boldsymbol{v}_t\|, \end{aligned}$$

where the last equality is due to the fact that $\{U^\intercal \boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^n$ only if $\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^n$, since U is unitary. For the lower-bound, if $t \leq n$, we can choose $\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^n$ such that $|\langle \boldsymbol{e}_1, \boldsymbol{v}_i\rangle| = 1/\sqrt{t}$ for all $i = 1, \cdots, t$. This choice is certainly possible as a result of applying Parseval's identity in a t-dimensional subspace with orthonormal basis $\{\boldsymbol{v}_i\}_1^t$ containing the unit vector $\boldsymbol{e}_1$, in which, $\boldsymbol{e}_1$ is represented with all coordinates equal to $1/\sqrt{t}$ with respect to this basis. We thus conclude that,

$$M_t(A) \geq |\sigma_1 \langle \boldsymbol{e}_1, \boldsymbol{v}_1\rangle| \cdots |\sigma_1 \langle \boldsymbol{e}_1, \boldsymbol{v}_t\rangle| \geq \left(\sigma_1/\sqrt{t}\right)^t,$$

where the left inequality follows from the fact that $\|\Sigma \boldsymbol{v}\| \ \geq \ |\sigma_1 \langle \boldsymbol{e}_1, \boldsymbol{v}\rangle|$ for any $\boldsymbol{v} \in \mathbb{R}^n$. For the upper-bound, define $\Sigma_t = \mathrm{diag}(\sigma_1, \ldots, \sigma_t)$ and since singular values are in descending order we have,

$$\begin{aligned} M_t(A) &\leq \sup_{\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^t} \prod_{i=1}^t \|\Sigma_t \boldsymbol{v}_i\| \\ &= \sup_{\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^t} \prod_{i=1}^t \left[\sigma_1^2 \left|\langle \boldsymbol{e}_1, \boldsymbol{v}_i\rangle\right|^2 + \sum_{j=2}^t \left|\sigma_j \langle \boldsymbol{e}_j, \boldsymbol{v}_i\rangle\right|^2\right]^{\frac{1}{2}} \\ &\leq \sup_{\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^t} \prod_{i=1}^t \left[\sigma_1^2 \left|\langle \boldsymbol{e}_1, \boldsymbol{v}_i\rangle\right|^2 + \delta\right]^{\frac{1}{2}}. \end{aligned}$$

Define $\gamma_i = \langle \boldsymbol{e}_1, \boldsymbol{v}_i\rangle$; then by Bessel's inequality $\sum_{i=1}^t \gamma_i^2 \leq 1$ whenever $\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^t$. Thereby, by denoting $\boldsymbol{\gamma} \coloneqq [\gamma_1 \ \ldots \ \gamma_t]^\intercal$, we can conclude that

$$\begin{aligned} M_t(A) &\leq \sup_{\boldsymbol{\gamma} \in \mathcal{B}_2^t} \prod_{i=1}^t \left[\sigma_1^2 \gamma_i^2 + \delta\right]^{\frac{1}{2}} \\ &= \sup_{\boldsymbol{\gamma} \in \mathcal{B}_2^t} \left[\sum_{i=1}^{t+1} \sigma_1^{2(t+1-i)} \, \delta^{i-1} \sum_{\substack{|\alpha| = t+1-i \\ \alpha \in \{0,1\}^t}} (\gamma_1^2)^{\alpha_1} \cdots (\gamma_t^2)^{\alpha_t}\right]^{\frac{1}{2}}, \end{aligned}$$

where α is a multi-index of dimension t, and the last equality follows by direct computation. Now it is easy to see that for a fixed multi-index α, if $|\alpha| = m > 0$ and $\alpha \in \{0,1\}^t$, then

$$\sup_{\boldsymbol{\gamma} \in \mathcal{B}_2^t} \ (\gamma_1^2)^{\alpha_1} \cdots (\gamma_t^2)^{\alpha_t} \leq \left(\tfrac{1}{m}\right)^m,$$

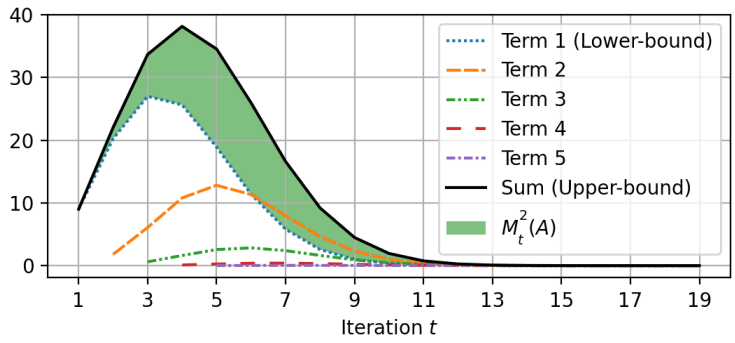


Figure 4.2: Illustration of the upper and lower bounds for instability number of a system with $\sigma_1 = 3$ and $\delta = 0.1$ as in Lemma 4.7.

which is followed by the symmetry in optimization variables. Therefore, we can conclude that

$$M_t(A) \leq \left[\delta^t + \sum_{i=1}^{t} \left[\frac{\sigma_1^2}{t+1-i}\right]^{(t+1-i)} \delta^{i-1} \binom{t}{t+1-i}\right]^{\frac{1}{2}},$$

implying the claimed upper bound. □

The lower and upper bounds in Lemma 4.7 show that, particularly when $\delta < 1$, $M_t(A)$ initially grows similar to $(\sigma_1/\sqrt{t})^t$ for $t \leq \sigma_1^2$, in contrast to the exponential growth of $\|A\|^t = \sigma_1^t$. This difference becomes more pronounced for $t > \sigma_1^2$ when $(\sigma_1/\sqrt{t})^t$ starts *decreasing*. This fact is illustrated via an example in Figure 4.2, where the first five dominant terms of the upper bound are plotted, and the green region shows where the actual value of $M_t^2(A)$ lies. The following result provides an upper bound on the state trajectories for the most general case through the lens of regularizability.

Theorem 4.8. *For any regularizable pair (A, B), the trajectory generated by Algorithm 1 satisfies the following bound for all $t > 0$,*

$$\|\boldsymbol{x}_{t+1}\| \leq L_{t+1}\|\boldsymbol{x}_0\|,$$

where L_t satisfies the recursion,

$$L_{t+1} = a_t + \sum_{r=1}^{t} b_{t,r} L_r, \qquad L_1 = \|A\overline{\boldsymbol{z}}_0\|,$$

with

$$b_{t,r} = \sqrt{\|\widetilde{A}^{t-r}\widetilde{B}\overline{\boldsymbol{z}}_r\|^2 + \|\widetilde{A}^{t-r}\Delta_\alpha\overline{\boldsymbol{w}}_r\|^2},$$

and $a_t = \|\widetilde{A}^t A\overline{\boldsymbol{z}}_0\|$*, where* $\overline{\boldsymbol{z}}_r = \boldsymbol{z}_r/\|\boldsymbol{z}_r\|$ *(if* $\boldsymbol{z}_r \neq 0$*, otherwise* $\overline{\boldsymbol{z}}_r = 0$*), and* $\overline{\boldsymbol{w}}_r$ *is similarly defined.*

Proof. Knowing that $\boldsymbol{x}_1 = A\boldsymbol{x}_0$, it follows that $\|\boldsymbol{x}_1\| \leq L_1\|\boldsymbol{x}_0\|$. Furthermore, for $t \geq 1$, (4.8) leads to,

$$\boldsymbol{x}_{t+1} = \widetilde{A}^t A\boldsymbol{x}_0 + \textstyle\sum_{r=1}^t \widetilde{A}^{t-r}\left[\widetilde{B}\boldsymbol{z}_r + \Delta_\alpha\boldsymbol{w}_r\right],$$

since $\widetilde{A} + \widetilde{B} = A$ and $\boldsymbol{w}_0 = 0$ by definition. This implies that,

$$\begin{aligned}\|\boldsymbol{x}_{t+1}\| &\leq \|\widetilde{A}^t A\boldsymbol{x}_0\| \\ &\quad + \textstyle\sum_{r=1}^t \|\widetilde{A}^{t-r}\widetilde{B}\overline{\boldsymbol{z}}_r\|\|\boldsymbol{z}_r\| + \|\widetilde{A}^{t-r}\Delta_\alpha\overline{\boldsymbol{w}}_r\|\|\boldsymbol{w}_r\|, \\ &\leq a_t\|\boldsymbol{x}_0\| + \textstyle\sum_{r=1}^t b_{t,r}\|\boldsymbol{x}_r\|,\end{aligned}$$

where we have used Cauchy–Schwarz inequality in conjunction with the equality $\|\boldsymbol{z}_r\|^2 + \|\boldsymbol{w}_r\|^2 = \|\boldsymbol{x}_r\|^2$. Using this recursive bound, the rest of the proof follows by induction. □

Remark 4.4. Note that in the analysis above, when the system is regularizable, a_t eventually decreases exponentially fast as t increases. Furthermore, the term $b_{t,r}$ in the sum increases as r approaches a fixed t. Finally, one can show that the obtained upper bound is tight by considering Example 4.1 with $\lambda_1 > 0$ and $\lambda_i = 0$ for $i > 1$.

Note that computing/estimating the upper bound in Theorem 4.8 requires knowledge of the matrix A, making these estimates more practical for structured systems (for example, see Corollary 4.11 and Remark 4.6).

In order to shed light on the intuition behind this upper bound, we next study simpler cases with $\alpha = 0$, where there exists small enough κ for which $b_{t,r} \leq \|\widetilde{A}^{t-r}\widetilde{B}\| \leq \kappa$ for all $r < t$. In particular, we can show that if the system is regularizable and $\widetilde{A}\widetilde{B} = 0$, then the trajectories of the closed-loop system will be bounded by a combination of instability number of different orders. This is stated in the following corollary of Theorem 4.8.

Corollary 4.9. *For any regularizable pair (A, B) with $\widetilde{A}\widetilde{B} = 0$, and $M_t(A)$ as in Definition 4.2, the system trajectory generated by Algorithm 1 with $\alpha = 0$ satisfies the following for all $t > 0$,*

$$\frac{\|\boldsymbol{x}_{t+1}\|}{\|\boldsymbol{x}_0\|} \leq a_t + \sum_{r=1}^{t-1} M_r(A) a_{t-r} + M_{t+1}(A)\,.$$

Proof. For brevity, let $b_t = b_{t,t}$, then as $\widetilde{A}\widetilde{B} = 0$, the recursion in Theorem 4.8 reduces to $L_{t+1} = a_t + b_t L_t$ with $L_1 = \|A\overline{\boldsymbol{z}}_0\|$; and its solution has the following form for all $t > 0$,

$$L_{t+1} = a_t + b_t \cdots b_2 b_1 L_1 + \textstyle\sum_{r=1}^{t-1} b_t \cdots b_{t+1-r}\, a_{t-r}. \tag{4.9}$$

As $\alpha = 0$ and orthogonal projection is non-expansive, we claim that $b_r = \|\widetilde{B}\overline{\boldsymbol{z}}_r\| \leq \|A\overline{\boldsymbol{z}}_r\|$ which vanishes whenever $\boldsymbol{z}_r = 0$. In the meantime, by Lemma 4.6, $\{\boldsymbol{z}_r\}_0^t$ must be a set of orthogonal vectors for any $t > 0$, and thus $\{\overline{\boldsymbol{z}}_r\}_0^t$ is a set of orthogonal vectors that are either normal or zero. Note that if $t \geq n$, then $\{\overline{\boldsymbol{z}}_r\}_0^t$ must contain at least one zero vector for dimensional reasons. Therefore, by Definition 4.2, we conclude that $b_t \cdots b_{t+1-r} \leq M_r(A)$, for each $r = 1, \ldots, t-1$. Similarly, as $L_1 = \|A\overline{\boldsymbol{z}}_0\|$, we have $b_t \cdots b_2 b_1 L_1 \leq M_{t+1}(A)$. By using these inequalities in (4.9), the claim follows by Theorem 4.8. □

The above observation further highlights the importance of the instability number in the context of DGR. Note that the terms in the upper bound involving $M_r(A)$ vanishes if $r > n$.

4.4.2 Informativity of the DGR Generated Data

In the sequel, we show that DGR generates linearly independent state-trajectory data; we refer to this as the informativity of data. We then proceed to make a connection between this independent structure and the number of excited modes in the system. Before we proceed, let us define $L_k^t(A)$, that is based on eigenvalues corresponding to k modes of a matrix A, as,

$$L_k^t(A) \coloneqq \begin{pmatrix} 1 & \lambda_1 & \cdots & (\lambda_1)^{t-1} \\ 1 & \lambda_2 & \cdots & (\lambda_2)^{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_k & \cdots & (\lambda_k)^{t-1} \end{pmatrix}, \quad 1 \leq t \leq n. \tag{4.10}$$

Remark 4.5. Note that $L_k^t(A)$ has a specific structure that hints at its invertibility. In fact, for $t = k$, $L_k^k(A)$ is the Vandermonde matrix formed by k eigenvalues of A, which would be invertible if and only if $\lambda_1, \cdots, \lambda_k$ are distinct. More generally, if $\{\lambda_1, \cdots, \lambda_k\}$ consists of r distinct eigenvalues (where $r \leq k$), then $L_k^r(A)$ has full column rank.

Intuitively, informative data–due to its linear independence structure–contain useful information for decision-making purposes. In particular, if the choice of $\boldsymbol{x}_0$ results in exciting all modes of A, one might expect that a useful online regulation algorithm should generate informative data at the same time that it is regulating the state trajectory. The next theorem formalizes how DGR realizes this expectation depending on what modes of the system are excited by the initial condition.

Theorem 4.10. *Let $\boldsymbol{x}_0$ excite $k_1 + k_2$ modes of A, such that k_1 modes are in $\mathcal{R}(B)$ and k_2 modes are in $\mathcal{R}(B)^\perp$. If the excited modes correspond to distinct eigenvalues, then $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{r-1}\}$, generated by Algorithm 1 with $\alpha = 0$, is a set of linearly independent vectors for any $r \leq \max\{k_1, k_2\}$.*

Proof. Without loss of generality, let $\lambda_1, \ldots, \lambda_{k_1}$ be the eigenvalues corresponding to the excited modes $u_1, \ldots, u_{k_1} \in \mathcal{R}(B)$, and similarly $\lambda_{k_1+1}, \ldots, \lambda_{k_1+k_2}$ be corresponding to $u_{k_1+1}, \ldots, u_{k_1+k_2} \in \mathcal{R}(B)^\perp$. Recall that $\mathcal{X}_{t-1} = [\boldsymbol{x}_0 \ \boldsymbol{x}_1 \ \ldots \ \boldsymbol{x}_{t-1}]$; then by definition of $\boldsymbol{z}_t$ in Lemma 4.6, for $t \geq 1$ there exists scalar coefficients $\zeta_0^t, \cdots, \zeta_{t-1}^t \in \mathbb{R}$ such that $\boldsymbol{z}_t = \boldsymbol{x}_t - \sum_{j=0}^{t-1} \zeta_j^t \boldsymbol{x}_j$. This together with the dynamics in Lemma 4.6 imply that $\boldsymbol{x}_1 = A\boldsymbol{x}_0$ and for $t \geq 2$,

$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} - \Pi_{\mathcal{R}(B)} \textstyle\sum_{j=0}^{t-2} \zeta_j^t A\boldsymbol{x}_j. \tag{4.11}$$

Since $\boldsymbol{x}_0$ excites k_1+k_2 modes of the system, we have $\boldsymbol{x}_0 = \sum_{\ell=1}^{k_1+k_2} \beta_\ell \boldsymbol{u}_\ell$, where β_ℓ are some nonzero real coefficients and $(\lambda_\ell, \boldsymbol{u}_\ell)$ are eigenpairs of A. Hence $\boldsymbol{x}_1 = A\boldsymbol{x}_0 = \sum_{\ell=1}^{k_1+k_2} \beta_\ell \lambda_\ell \boldsymbol{u}_\ell$, and we claim that for $t \geq 2$ there exist scalar coefficients $\xi_1^t, \cdots, \xi_{t-1}^t \in \mathbb{R}$ such that,

$$\boldsymbol{x}_t = \sum_{\ell=1}^{k_1} \beta_\ell \Big[(\lambda_\ell)^t - \sum_{i=1}^{t-1} \xi_i^t (\lambda_\ell)^i \Big] \boldsymbol{u}_\ell + \sum_{\ell=k_1+1}^{k_1+k_2} \beta_\ell (\lambda_\ell)^t \boldsymbol{u}_\ell. \tag{4.12}$$

The proof of the last claim is by induction. Note that $A^t\boldsymbol{x}_0 = \sum_{\ell=1}^{k_1+k_2} \beta_\ell(\lambda_\ell)^t\boldsymbol{u}_\ell$, and by substituting this into (4.11) for $t = 2$ we have that,

$$\begin{aligned}\boldsymbol{x}_2 &= A\boldsymbol{x}_1 - \zeta_0^2\Pi_{\mathcal{R}(B)}A\boldsymbol{x}_0 \\ &= \Pi_{\mathcal{R}(B)}\left[A^2\boldsymbol{x}_0 - \zeta_0^2 A\boldsymbol{x}_0\right] + \Pi_{\mathcal{R}(B)^\perp}A^2\boldsymbol{x}_0 \\ &= \textstyle\sum_{\ell=1}^{k_1}\beta_\ell\left[(\lambda_\ell)^2 - \zeta_0^2\lambda_\ell\right]\boldsymbol{u}_\ell + \sum_{\ell=k_1+1}^{k_1+k_2}\beta_\ell(\lambda_\ell)^2\boldsymbol{u}_\ell,\end{aligned}$$

where the last equality is due to the fact that $\boldsymbol{u}_\ell \in \mathcal{R}(B)$ for $\ell \leq k_1$ and $\boldsymbol{u}_\ell \in \mathcal{R}(B)^\perp$ for $\ell > k_1$. By choosing $\xi_1^2 = \zeta_0^2$, we have shown that (4.12) holds for $t = 2$. Now suppose that (4.12) holds for all $2, \ldots, t-1$; it now suffices to show that this relation also holds for t. By substituting the hypothesis for $2, \ldots, t-1$ into (4.11),

$$\begin{aligned}\boldsymbol{x}_t = &\textstyle\sum_{\ell=1}^{k_1}\beta_\ell\Big[(\lambda_\ell)^t - \sum_{i=1}^{t-2}\xi_i^{t-1}(\lambda_\ell)^{i+1}\Big]\boldsymbol{u}_\ell \\ &+ \textstyle\sum_{\ell=k_1+1}^{k_1+k_2}\beta_\ell(\lambda_\ell)^t\boldsymbol{u}_\ell - \sum_{\ell=1}^{k_1}\beta_\ell[\zeta_0^t\lambda_\ell + \zeta_1^t(\lambda_\ell)^2]\boldsymbol{u}_\ell \\ &- \textstyle\sum_{j=2}^{t-2}\zeta_j^t\sum_{\ell=1}^{k_1}\beta_\ell\Big[(\lambda_\ell)^{j+1} - \sum_{i=1}^{j-1}\xi_i^j(\lambda_\ell)^{i+1}\Big]\boldsymbol{u}_\ell.\end{aligned}$$

Therefore, $\boldsymbol{x}_t = \sum_{\ell=1}^{k_1}\beta_\ell\left[\star\right]\boldsymbol{u}_\ell + \sum_{\ell=k_1+1}^{k_1+k_2}\beta_\ell(\lambda_\ell)^t\boldsymbol{u}_\ell$,

where $\star$ replaces the expression,

$$(\lambda_\ell)^t - \sum_{i=1}^{t-2}\xi_i^{t-1}(\lambda_\ell)^{i+1} - \sum_{j=0}^{t-2}\zeta_j^t(\lambda_\ell)^{j+1} + \sum_{j=2}^{t-2}\sum_{i=1}^{j-1}\zeta_j^t\xi_i^j(\lambda_\ell)^{i+1}.$$

By appropriate choices of $\xi_1^t, \cdots, \xi_{t-1}^t \in \mathbb{R}$, we can rewrite $\star = (\lambda_\ell)^t - \sum_{i=1}^{t-1}\xi_i^t(\lambda_\ell)^i$. This completes the proof of the claim in (4.12) by induction. Now, let $\widehat{\boldsymbol{x}} = \sum_{j=0}^{r-1}\gamma_j\boldsymbol{x}_j$ for some $\gamma_j \in \mathbb{C}$ and some $r \leq \max\{k_1, k_2\}$. Then, by substituting $\boldsymbol{x}_j$ from (4.12) and exchanging the sums over j and ℓ we have,

$$\begin{aligned}\widehat{\boldsymbol{x}} = &\textstyle\sum_{\ell=1}^{k_1}\beta_\ell\Big[\gamma_0 + \gamma_1\lambda_\ell + \sum_{j=2}^{r-1}\gamma_j\big[(\lambda_\ell)^j - \sum_{i=1}^{j-1}\xi_i^j(\lambda_\ell)^i\big]\Big]\boldsymbol{u}_\ell \\ &+ \textstyle\sum_{\ell=k_1+1}^{k_1+k_2}\beta_\ell\sum_{j=0}^{r-1}\gamma_j(\lambda_\ell)^j\boldsymbol{u}_\ell.\end{aligned}$$

Now, by exchanging the sums over i and j, it follows that,

$$\widehat{\boldsymbol{x}} = \textstyle\sum_{\ell=1}^{k_1} \beta_\ell \Big[\gamma_0 + \sum_{i=1}^{r-2} \big[\gamma_i - \sum_{j=i+1}^{r-1} \gamma_j \xi_i^j\big](\lambda_\ell)^i + \gamma_{r-1}(\lambda_\ell)^{r-1}\Big]\boldsymbol{u}_\ell$$
$$+ \textstyle\sum_{\ell=k_1+1}^{k_1+k_2} \beta_\ell \big[\sum_{j=0}^{r-1} \gamma_j (\lambda_\ell)^j\big]\boldsymbol{u}_\ell.$$

Since $\{\boldsymbol{u}_\ell\}_1^{k_1+k_2}$ are eigenvectors associated with distinct eigenvalues, they are linearly independent. Thus, noting that $\beta_\ell \neq 0$ for all $\ell = 1, \cdots, k_1 + k_2$, then $\widehat{\boldsymbol{x}} = 0$ implies that,

$$\gamma_0 + \sum_{i=1}^{r-2} \Big[\gamma_i - \sum_{j=i+1}^{r-1} \gamma_j \xi_i^j\Big](\lambda_\ell)^i + \gamma_{r-1}(\lambda_\ell)^{r-1} = 0,$$

for all $\ell = 1, \ldots, k_1$; and $\sum_{j=0}^{r-1} \gamma_j (\lambda_\ell)^j = 0$,

for all $\ell = k_1 + 1, \ldots, k_1 + k_2$. By rewriting the last two sets of equations in matrix form, we get

$$\begin{pmatrix} L_{k_1}^r(A)(I - \Xi) \\ \widehat{L}_{k_2}^r(A) \end{pmatrix} \boldsymbol{\gamma} = 0, \tag{4.13}$$

where $\widehat{L}_{k_2}^r(A)$ is the last k_2 rows of $L_{k_1+k_2}^r(A)$ and

$$\Xi \coloneqq \begin{pmatrix} 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & \xi_1^2 & \xi_1^3 & \cdots & \xi_1^{r-1} \\ 0 & 0 & 0 & \xi_2^3 & \cdots & \xi_2^{r-1} \\ 0 & 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \xi_{r-1}^{r-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, \; \boldsymbol{\gamma} \coloneqq \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \\ \vdots \\ \\ \gamma_{r-1} \end{pmatrix}. \tag{4.14}$$

Note that $I - \Xi$ is invertible by construction. Since the excited modes correspond to distinct eigenvalues, if $r \leq \max\{k_1, k_2\}$, then either $L_{k_1}^r(A)$ or $\widehat{L}_{k_2}^r(A)$ has full column rank. Either way, (4.13) implies that $\boldsymbol{\gamma} = 0$ and thus $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{r-1}\}$ is a set of linearly independent vectors. This observation completes the proof as $r \leq \max\{k_1, k_2\}$ was chosen arbitrary. □

The preceding theorem guarantees the linear independence of the state-trajectory generated by DGR whenever the exited modes lie in $\mathcal{R}(B)$ or $\mathcal{R}(B)^\perp$, even though our observations

suggest that it must remain valid for arbitrary excitation of the modes. Nonetheless, DGR remains effective in terms of online regulation from an arbitrary choice of x_0 as guaranteed in Theorem 4.8, Corollary 4.9, and subsequently in Corollary 4.11.

4.4.3 Special Case of $\mathcal{R}(A) \subset \mathcal{R}(B)$ with $\alpha = 0$

In order to better understand the behavior of DGR, in this subsection, we study the more special case where $\mathcal{R}(A) \subset \mathcal{R}(B)$. This includes the case where $\text{rank}(B) = n$, i.e., one can directly control each state of the system (e.g. see [69, 92]). Note that $\mathcal{R}(A) \subset \mathcal{R}(B)$ implies that $\widetilde{A} = 0$ which, in turn, results in regularizability of (A, B). This, together with Corollary 4.9, results in the following corollary.

Corollary 4.11. *For any matrix $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$, where $\mathcal{R}(A) \subseteq \mathcal{R}(B)$, the trajectory generated by Algorithm 1 with $\alpha = 0$ satisfies the following for all $t > 0$,*

$$\|\boldsymbol{x}_{t+1}\| \leq M_{t+1}(A)\|\boldsymbol{x}_0\|\,,$$

with $M_t(A)$ as in Definition 4.2.

Proof. Note that $\widetilde{A} = \Pi_{\mathcal{R}(B)^\perp} A = 0$ whenever $\mathcal{R}(A) \subseteq \mathcal{R}(B)$. The claim now follows by Corollary 4.9 since $\widetilde{A}\widetilde{B} = 0$ and $a_k = 0$ for all $k = 1, \ldots, t$. □

Note that under the hypothesis of Corollary 4.11, in particular $\boldsymbol{x}_{t+1} = 0$ for all $t \geq n$ whenever DGR is in effect for the noiseless dynamics in (4.1) (see Figure 4.3); however, this might happen even before t reaches n as will be discussed in Proposition 4.13. Also, the latter bound becomes more structured for a symmetric A by combining the results from Corollary 4.11 and Lemma 4.7, which we skip for brevity.

Remark 4.6. In order to further illustrate the bound stated in Corollary 4.11, assume that $\delta e \leq 1$. Then, from Lemma 4.7,

$$\frac{\|\boldsymbol{x}_t\|^2}{\|\boldsymbol{x}_0\|^2} \leq \left[\frac{\sigma_1^2}{t}\right]^t + \sum_{j=1}^{\lfloor t/2\rfloor} \left[\frac{\sigma_1^2}{t-j}\right]^{t-j} \left[\frac{t}{j}\right]^j + \sum_{j=\lfloor t/2\rfloor+1}^{t-1} \left[\frac{t\,\sigma_1^2}{(t-j)^2}\right]^{t-j} + 1,$$

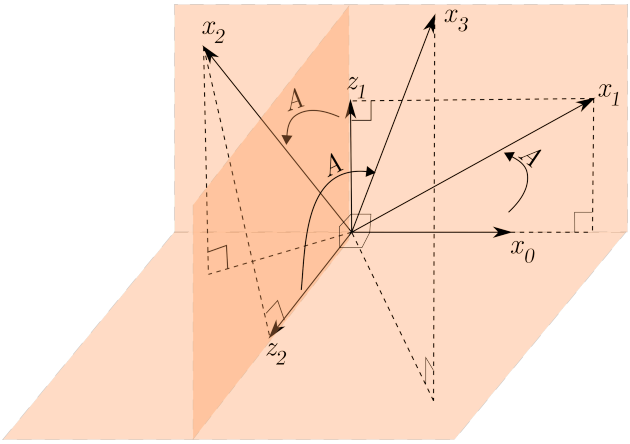


Figure 4.3: A geometric schematic of DGR when $\mathcal{R}(A) \subseteq \mathcal{R}(B)$. Since $\boldsymbol{z}_0 \coloneqq \boldsymbol{x}_0$, $\boldsymbol{z}_t \perp \mathcal{R}(\mathcal{X}_{t-1})$ and $\boldsymbol{z}_t \in \mathcal{R}(\mathcal{X}_t)$ for $t = 1, 2$, the set $\{\boldsymbol{z}_0, \boldsymbol{z}_1, \boldsymbol{z}_2\}$ consists of orthogonal vectors.

where we have also used $\binom{t}{j} \leq (e\,t/j)^j$. This implies that as t gets larger than σ_1^2, the terms with large powers admit smaller bases and those with large bases will gain smaller powers comparing to σ_1^{2t}. This is despite the fact that for small t, the relative norm of the state might grow.

In the sequel, as a result of linear independence established in Theorem 4.10 we show how the simplified bounds (derived in Section 4.4.1) clarify the elimination of the unstable modes in the system.

Corollary 4.12. *Suppose $\mathcal{R}(A) \subseteq \mathcal{R}(B)$ and let $\boldsymbol{x}_0$ excite k modes of A. If r eigenvalues corresponding to the k excited modes are distinct for some $r \leq k$, then $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{r-1}\}$, generated by Algorithm 1 with $\alpha = 0$, is a set of linearly independent vectors.*

Proof. Given that $\mathcal{R}(A) \subseteq \mathcal{R}(B)$, all the modes of A are contained in $\mathcal{R}(B)$, so without loss of generality, let $\lambda_1, \ldots, \lambda_k$ be the eigenvalues corresponding to the excited modes $u_1, \ldots, u_k \in \mathcal{R}(B)$. Then, following the proof of Theorem 4.10, (4.12) reduces to,

$$\boldsymbol{x}_t = \textstyle\sum_{\ell=1}^{k} \beta_\ell \left[(\lambda_\ell)^t - \sum_{j=1}^{t-1} \xi_j^t (\lambda_\ell)^j \right] \boldsymbol{u}_\ell.$$

Now, let $\widehat{\boldsymbol{x}} = \sum_{j=0}^{r-1} \gamma_j \boldsymbol{x}_j$ for some $\gamma_j \in \mathbb{C}$ and $r \leq k$. Then following the same argument as

in the proof of Theorem 4.10 about $\widehat{\boldsymbol{x}}$, (4.13) reduces to $L_k^r(A)(I-\Xi)\boldsymbol{\gamma}=0$, with similar definitions of Ξ and $\boldsymbol{\gamma}$ as in (4.10), and $L_k^r(A)$ as in (4.14). Since r eigenvalues corresponding to k excited modes are distinct, $L_k^r(A)$ has full column rank. As $I-\Xi$ is invertible, we conclude that $\boldsymbol{\gamma}=0$ meaning that $\{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{r-1}\}$ are linearly independent. □

An immediate consequence of the above corollary is that DGR generates data that is effective for the simultaneous identification of modes even with multiplicity greater than one.

Proposition 4.13. *Suppose that A is diagonalizable with $\mathcal{R}(A)\subseteq\mathcal{R}(B)$, and let $\boldsymbol{x}_0$ excite k modes of A corresponding to r distinct eigenvalues (where possibly, $r\leq k$). Then, in exactly r iterations of Algorithm 1 with $\alpha=0$, $\operatorname{span}\{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{r-1}\}$ coincides with the subspace containing these excited modes; furthermore, $\boldsymbol{x}_{r+1}=0$.*

Proof. Without loss of generality, let $\boldsymbol{x}_0$ excite the k modes of A corresponding to $\lambda_1,\cdots,\lambda_r$. Since A is diagonalizable, let $A=U\Lambda U^{-1}$ be its eigen-decomposition and so $\boldsymbol{x}_0$ excite $\{\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k\}$, i.e., $\boldsymbol{x}_0=\sum_{i=1}^k\beta_i\boldsymbol{u}_i$, with $\beta_i\neq 0$. Define

$$\mathcal{I}(\lambda_i)=\{j:\boldsymbol{u}_j \text{ is the eigenvector corresponding to } \lambda_i\},$$

for $i=1,\cdots,r$. Furthermore, define the r-dimensional subspace,

$$S\coloneqq\operatorname{span}\left\{\textstyle\sum_{j\in\mathcal{I}(\lambda_1)}\beta_j\boldsymbol{u}_j,\ \cdots,\sum_{j\in\mathcal{I}(\lambda_r)}\beta_j\boldsymbol{u}_j\right\},$$

where the span is taken over the complex field. We prove by induction that $\boldsymbol{x}_t\in S$ for all $t=1,\cdots,r$. Notice that $\boldsymbol{x}_0\in S$ and suppose that $\{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{t-1}\}\subset S$; recall from the proof of Corollary 4.12 that $\boldsymbol{x}_t=A\boldsymbol{z}_{t-1}$, where $\boldsymbol{z}_{t-1}=\Pi_{\mathcal{R}(\mathcal{X}_{t-2})^\perp}(\boldsymbol{x}_{t-1})$. Since $\boldsymbol{x}_{t-1}\in S$ and $\operatorname{span}\{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{t-2}\}\subset S$, one can conclude that $\boldsymbol{z}_{t-1}\in S$, and from the definition of S, $\boldsymbol{x}_t=A\boldsymbol{z}_{t-1}\in S$. On the other hand, since $\lambda_1,\lambda_2,\cdots,\lambda_r$ are distinct eigenvalues, by Corollary 4.12, $\mathbf{dim}\left(\operatorname{span}\{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{r-1}\}\right)=r$. By hypothesis of the induction $\operatorname{span}\{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{r-1}\}\subset S$, and since $\mathbf{dim}(S)=r$, we conclude that $\operatorname{span}\{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{r-1}\}$ must be the entire S, i.e. $\operatorname{span}\{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{r-1}\}=S$, proving the first claim. Lastly, $\boldsymbol{z}_r=\Pi_{\mathcal{R}(\mathcal{X}_{r-1})^\perp}(\boldsymbol{x}_r)=0$ since $\boldsymbol{x}_r\in S$, and thus $\boldsymbol{x}_{r+1}=A\boldsymbol{z}_r=0$, thereby completing the proof. □

Following Proposition 4.13, if $\boldsymbol{x}_0$ excites k modes of the system corresponding to distinct eigenvalues with trivial algebraic multiplicities, then Algorithm 1 identifies all the excited modes of the system in k iterations. Furthermore, this implies that $\boldsymbol{x}_{k+1} = 0$, i.e., DGR eliminates the unstable modes and regulates the unknown system in exactly $k+1$ iterations. This also implies that if $k < n$, then online regulation of the system is achieved, even before enough data is available for full identification of system parameters.

4.5 Boosting the Performance of DGR

DGR as introduced in Algorithm 1 can become computationally burdensome for large-scale systems. This is mainly due to storing the entire history of data in $\mathcal{X}_t$ and $\mathcal{Y}_t$ followed by the update of the controller that finds the pseudoinverse as well as multiplication of these data matrices (steps 7-9). Assuming the SVD-based computation of pseudoinverse, the complexity of the method is[9] $\mathcal{O}(n^2 t)$. In this section, we show that such computational burden can be circumvented using rank-1 modifications of data matrices as a result of the discrete nature of data collection in our setup. Note that for computing K_{t+1} from (4.6) we only need to access $\mathcal{Y}_{t+1}\mathcal{X}_t^\dagger$ (rather than $\mathcal{X}_t^\dagger$). To this end, we leverage the results of [93] in order to find $\mathcal{Y}_{t+1}\mathcal{X}_t^\dagger$ recursively as a function of $\mathcal{Y}_t\mathcal{X}_{t-1}^\dagger$, $\mathcal{X}_{t-1}\mathcal{X}_{t-1}^\dagger$, and $\boldsymbol{x}_t$.

Proposition 4.14. *Let $\mathcal{X}_{t-1}$ be as in Algorithm 1, $\boldsymbol{x}_t$ be the state measurement at iteration t and $\boldsymbol{z}_t = \Pi_{\mathcal{R}(\mathcal{X}_{t-1})^\perp}\boldsymbol{x}_t$. If $\boldsymbol{x}_t \notin \mathcal{R}(\mathcal{X}_{t-1})$ then*

$$\mathcal{X}_t^\dagger = \begin{pmatrix} \mathcal{X}_{t-1}^\dagger - \boldsymbol{\gamma}_t \boldsymbol{z}_t^\dagger \\ \boldsymbol{z}_t^\dagger \end{pmatrix}; \tag{4.15}$$

otherwise,

$$\mathcal{X}_t^\dagger = \begin{pmatrix} \mathcal{X}_{t-1}^\dagger - \epsilon_t \boldsymbol{\gamma}_t \boldsymbol{\zeta}_t^\intercal \\ \epsilon_t \boldsymbol{\zeta}_t^\intercal \end{pmatrix}, \tag{4.16}$$

where $\epsilon_t \in \mathbb{R}$, $\boldsymbol{\gamma}_t \in \mathbb{R}^t$, and $\boldsymbol{\zeta}_t \in \mathbb{R}^n$ are defined as,

$$\epsilon_t \coloneqq \frac{1}{\|\boldsymbol{\gamma}_t\|^2 + 1}, \;\; \boldsymbol{\gamma}_t \coloneqq \mathcal{X}_{t-1}^\dagger \boldsymbol{x}_t, \;\; \boldsymbol{\zeta}_t \coloneqq \left(\mathcal{X}_{t-1}^\dagger\right)^\intercal \boldsymbol{\gamma}_t. \tag{4.17}$$

[9]The multiplication $\mathcal{Y}_{t+1}\mathcal{X}_t^\dagger$ requires another $\mathcal{O}(n^2 t)$ operations that can be significant for large n.

Proof. Rearrange $\mathcal{X}_t$ into,

$$\mathcal{X}_t = \begin{pmatrix} \mathcal{X}_{t-1} & 0 \end{pmatrix} + \boldsymbol{x}_t \boldsymbol{e}_{t+1}^{\intercal}.$$

Then, it is implied from Theorem 1 in [93] that

$$\mathcal{X}_t^{\dagger} = \begin{pmatrix} \mathcal{X}_{t-1} & 0 \end{pmatrix}^{\dagger} + \left[\boldsymbol{e}_{t+1} - \begin{pmatrix} \mathcal{X}_{t-1} & 0 \end{pmatrix}^{\dagger} \boldsymbol{x}_t \right] \boldsymbol{z}_t^{\dagger},$$

whenever $\boldsymbol{x}_t \notin \mathcal{R}(\mathcal{X}_{t-1})$. Hence, by leveraging the SVD of $\mathcal{X}_{t-1}$ and definition of pseudoinverse we get

$$\mathcal{X}_t^{\dagger} = \begin{pmatrix} \mathcal{X}_{t-1}^{\dagger} \\ 0 \end{pmatrix} + \left[\boldsymbol{e}_{t+1} - \begin{pmatrix} \mathcal{X}_{t-1}^{\dagger} \\ 0 \end{pmatrix} \boldsymbol{x}_t \right] \boldsymbol{z}_t^{\dagger} = \begin{pmatrix} \mathcal{X}_{t-1}^{\dagger} - \boldsymbol{\gamma}_t \boldsymbol{z}_t^{\dagger} \\ \boldsymbol{z}_t^{\dagger} \end{pmatrix}.$$

For the case when $\boldsymbol{x}_t \in \mathcal{R}(\mathcal{X}_{t-1})$, Theorem 3 in [93] gives,

$$\mathcal{X}_t^{\dagger} = \begin{pmatrix} \mathcal{X}_{t-1}^{\dagger} \\ 0 \end{pmatrix} + \boldsymbol{e}_{t+1} \boldsymbol{\gamma}_t^{\intercal} \mathcal{X}_{t-1}^{\dagger} - \frac{1}{\sigma} \boldsymbol{p} \boldsymbol{q}^{\intercal},$$

where, $\sigma = \|\boldsymbol{\gamma}_t\|^2 + 1$, $\boldsymbol{p} = -\|\boldsymbol{\gamma}_t\|^2 \boldsymbol{e}_{t+1} - \begin{pmatrix} \boldsymbol{\gamma}_t \\ 0 \end{pmatrix}$, $\boldsymbol{q} = -\boldsymbol{\zeta}_t$. The rest of the proof follows from rearranging the terms and using the identities in (4.17). □

As mentioned earlier, the update of the controller requires $\mathcal{Y}_t \mathcal{X}_{t-1}^{\dagger}$ that could become prohibitive for large n. However, we can take advantage of Proposition 4.14 to find this term recursively in order to avoid memory usage as well as computational burden.

Theorem 4.15. *Let $\mathcal{X}_{t-1}$ be as in Algorithm 1 and $\boldsymbol{x}_t$ be the state measurement collected at t. For $t > 0$, define $\mathcal{P}_{t-1} \coloneqq \mathcal{X}_{t-1} \mathcal{X}_{t-1}^{\dagger}$, $\mathcal{Q}_{t-1} \coloneqq \mathcal{Y}_t \mathcal{X}_{t-1}^{\dagger}$ and $\boldsymbol{z}_t \coloneqq \Pi_{\mathcal{R}(\mathcal{X}_{t-1})^{\perp}} \boldsymbol{x}_t$. Then*

$$\begin{aligned} \boldsymbol{z}_t &= \left[\mathrm{I} - \mathcal{P}_{t-1} \right] \boldsymbol{x}_t, \\ \mathcal{Q}_t &= \mathcal{Q}_{t-1} + [\boldsymbol{x}_{t+1} - B\boldsymbol{u}_t - \mathcal{Q}_{t-1} \boldsymbol{x}_t] \boldsymbol{z}_t^{\dagger}, \\ \mathcal{P}_t &= \mathcal{P}_{t-1} + \boldsymbol{z}_t \boldsymbol{z}_t^{\dagger}. \end{aligned}$$

Algorithm 2 Fast Data-Guided Regulation (F-DGR)

Initialization

  Measure $\boldsymbol{x}_0$, set $K_0 = 0$ and $G_\alpha = [\alpha I + B^\intercal B]^\dagger B^\intercal$

  Set $\mathcal{P}_{-1} = \mathcal{Q}_{-1} = \mathbf{0}$ and $t = 0$

While stopping criterion not met

  Compute $\boldsymbol{u}_t = -K_t \boldsymbol{x}_t$

  Run system (4.1) and measure $\boldsymbol{x}_{t+1}$

  Set $\boldsymbol{z}_t = [\mathrm{I} - \mathcal{P}_{t-1}]\boldsymbol{x}_t$

    $\mathcal{Q}_t = \mathcal{Q}_{t-1} + [\boldsymbol{x}_{t+1} - B\boldsymbol{u}_t - \mathcal{Q}_{t-1}\boldsymbol{x}_t]\boldsymbol{z}_t^\dagger$

    $\mathcal{P}_t = \mathcal{P}_{t-1} + \boldsymbol{z}_t \boldsymbol{z}_t^\dagger$

    $K_{t+1} = G_\alpha \mathcal{Q}_t$

  $t = t + 1$

Proof. The expression for $\boldsymbol{z}_t$ follows directly by properties of pseudoinverse. Next, observing that $\boldsymbol{x}_{t+1} - B\boldsymbol{u}_t = A\boldsymbol{x}_t$ and so $\mathcal{Q}_t = A\mathcal{P}_t$, the recursive relation for $\mathcal{Q}_t$ can be derived from the one for $\mathcal{P}_t$. Finally, Proposition 4.14 implies that, if $\boldsymbol{x}_t \notin \mathcal{R}(\mathcal{X}_{t-1})$ then

$$\mathcal{P}_t = \mathcal{X}_t \mathcal{X}_t^\dagger = \begin{pmatrix} \mathcal{X}_{t-1} & \boldsymbol{x}_t \end{pmatrix} \begin{pmatrix} \mathcal{X}_{t-1}^\dagger - \boldsymbol{\gamma}_t \boldsymbol{z}_t^\dagger \\ \boldsymbol{z}_t^\dagger \end{pmatrix} = \mathcal{P}_{t-1} + \boldsymbol{z}_t \boldsymbol{z}_t^\dagger;$$

otherwise, $\mathcal{P}_t = \mathcal{P}_{t-1}$ because $\mathcal{X}_{t-1}\boldsymbol{\gamma}_t = \mathcal{X}_{t-1}\mathcal{X}_{t-1}^\dagger \boldsymbol{x}_t = \boldsymbol{x}_t$. But in this case, $\boldsymbol{z}_t = (\mathrm{I} - \mathcal{P}_{t-1})\boldsymbol{x}_t = 0$ and therefore, the same recursion holds. □

Given the recursions introduced in Theorem 4.15, the refined (fast) version of DGR is displayed in Algorithm 2. At each iteration t, we update $\mathcal{Q}_t$ based on the information from the new data and the projection $\mathcal{P}_{t-1}$ (hidden in $\boldsymbol{z}_t^\dagger$), which itself gets updated as a part of the recursion. The $n \times n$ matrix $\mathcal{Q}_t$ is then employed for the controller's update. Notice that $\boldsymbol{z}_t$ is the same as in Lemma 4.6; however, here we compute it using $\mathcal{P}_{t-1}$, which is obtained recursively.

Next, we discuss the convergence of DGR algorithm in the following. Note that, by Theorem 4.15, DGR and F-DGR are equivalent, and the following corollary provides a useful

necessary and sufficient condition for the convergence of $\mathcal{Q}_t$.

Corollary 4.16. *Let $\mathcal{Q}_t$ be as defined in Algorithm 2 and $\mathcal{X}_{t-1}$ as in Algorithm 1. For any $t > 0$, if $\boldsymbol{x}_{t+k} \in \mathcal{R}(\mathcal{X}_{t-1})$ for all $k \geq 0$, or $\mathcal{R}(A^\intercal) \subseteq \mathcal{R}(\mathcal{X}_{t-1})$ then Algorithm 2 has converged at t, i.e., $\mathcal{Q}_{t+k} = \mathcal{Q}_{t-1}$ for all $k \geq 0$. Conversely, if Algorithm 2 has converged at some $t > 0$, then for each $k \geq 0$, we must have $\boldsymbol{x}_{t+k} \in \mathcal{R}(\mathcal{X}_{t+k-1})$ unless $\mathcal{R}(A^\intercal) \subseteq \mathcal{R}(\mathcal{X}_{t+k-1})$.*

Proof. By Theorem 4.15, we know that

$$\mathcal{Q}_t - \mathcal{Q}_{t-1} = A(\mathcal{P}_t - \mathcal{P}_{t-1}) = A\boldsymbol{z}_t\boldsymbol{z}_t^\dagger.$$

Therefore, Algorithm 2 has converged at t if and only if $A\boldsymbol{z}_{t+k}\boldsymbol{z}_{t+k}^\dagger = 0$ for all $k \geq 0$. And the latter statement is valid if and only if, for each $k \geq 0$, either $\boldsymbol{z}_{t+k} = 0$ or $\boldsymbol{z}_{t+k} \in \mathcal{N}(A)$, which is equivalent to either $\boldsymbol{x}_{t+k} \in \mathcal{R}(\mathcal{X}_{t+k-1})$ or $\mathcal{R}(A^\intercal) \subseteq \mathcal{R}(\mathcal{X}_{t+k-1})$. □

A simple implication of the preceding corollary is that Algorithm 2 converges as soon as the collected data $\mathcal{X}_{t-1}$ is *rich* enough. For instance, in the worst case scenario–when $A \in \mathbb{R}^{n\times n}$ has no zero eigenvalues, and all of its modes are excited–the algorithm converges as soon as n linearly independent state measurements have been collected from the noise-less dynamics in (4.1). From this point on, $\boldsymbol{u}_t$'s as computed in DGR and F-DGR coincide with the solution of (4.5). However, the convergence of the controller in DGR and F-DGR may happen earlier in the process whenever the future state measurements lie in the range of previous ones. For example, under the hypothesis of Proposition 4.13, when $\boldsymbol{x}_0$ excites k modes of A corresponding to $r \leq k < n$ distinct eigenvalues, then both DGR and F-DGR converge in r iterations. In this case, the proposed controller may not coincide with the actual solution of the optimization in (4.5). Nonetheless, the online regulation of the system is guaranteed in general by Theorem 4.8. Note that, in the presence of process noise in the dynamics (4.1), Corollary 4.16 is not valid, and the convergence behavior of DGR (and F-DGR) will be dictated by the noise stochastics (see Figure 4.7 in Section 4.6).

Finally, $\boldsymbol{z}_t$ reflects the informativity of the newly generated data $\boldsymbol{x}_t$. In fact, based on its definition, $\mathcal{Q}_t$ provides an estimate of A up to iteration t. Hence, the update of $\mathcal{Q}_t$ as

in Algorithm 2 essentially adjusts the prior estimate of A based on the new information encoded in the term $A\boldsymbol{z}_t\boldsymbol{z}_t^\dagger$. All in all, the machinery provided in this section circumvents the computational load of finding pseudoinverses by leveraging the recursive nature of the solution methodology.

4.6 Numerical Example

In order to showcase the advantages of the proposed method in practical settings, we have implemented DGR on data collected from the X-29A aircraft. The Grumman X-29A is an experimental aircraft initially tested for its forward-swept wing; it was designed with a high degree of longitudinal static instability (due to the location of the aerodynamic center on the wings) for maneuverability, where linear models were leveraged to determine the closed-loop stability (Figure 4.4). The primary task of the control laws is to stabilize the longitudinal motion of the aircraft. To this end, the dynamic elements of the flight control system are designed for two general modes: 1) the Normal Digital Powered Approach (ND-PA) used in the takeoff and landing phase of the flight, and 2) the Normal Digital Up-and-Away (ND-UA) when otherwise.

For both flight modes, we study the case where the aircraft's dynamics have been perturbed and are unknown. This can be due to a misestimation of system parameters and/or any unpredicted flaw in the flight dynamics caused by malfunction/damage. In this setting, the control laws designed for the original system fail, and the system can become highly unstable. We then let DGR regulate the system; in this case, since the aircraft continues to operate safely, one can use any data-driven identification, stabilization, or robust control methods once enough data has been collected.

The longitudinal and lateral-directional dynamics each contain 4 states (see Figure 4.4). The nominal system parameters in each operating mode are obtained from Tables 9-10 and 13-14 in [94] (with fixed discretization step-size 0.05), whereas perturbation ΔA is assumed to shift the dynamics to,

$$\boldsymbol{x}_{t+1} = (A + \Delta A)\boldsymbol{x}_t + B\boldsymbol{u}_t + \boldsymbol{\omega}_t,$$

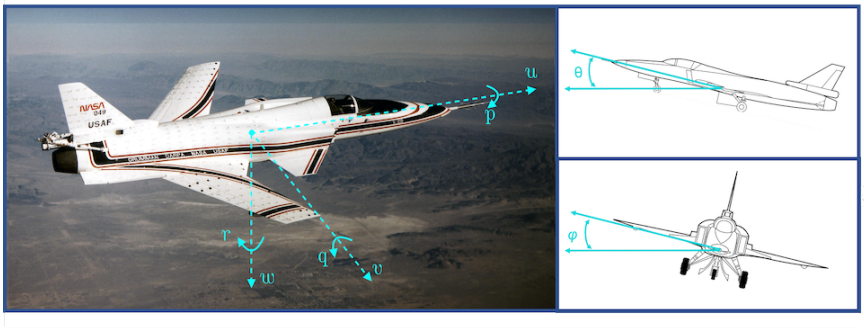


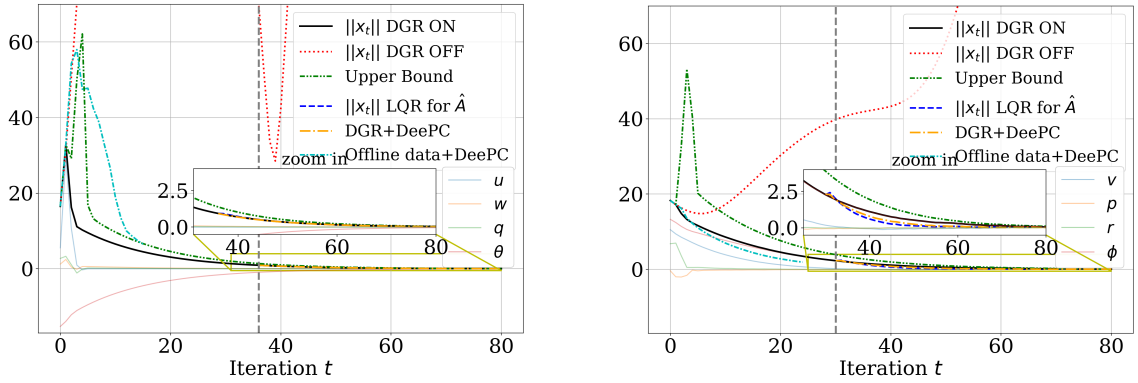
Figure 4.4: Grumman X-29A *(Credits: NASA Photo)*, mainly known for its extreme instability while providing high-quality maneuverability; the longitudinal and lateral-directional states are illustrated.

where the elements of ΔA are sampled from a normal distribution $\mathcal{N}(0, 0.05)$, and $\boldsymbol{\omega}_t \sim \mathcal{N}(0, 0.01)$ denotes the process noise. Note that even though the nominal dynamics is known in this example, the proposed machinery makes no such *a priori* estimate, and assumes a completely unknown dynamics $A_{\text{new}} := A + \Delta A$. The original controller for the unperturbed system in each mode is assumed to be a closed-loop infinite horizon LQR with state and input weights $Q = I$ and $R = 10^{-7}$.

We now aim to regulate the unstable system A_{new} from random initial states (where each state is sampled from $\mathcal{N}(0, 10.0)$). Note that both the original system and the perturbed system have effective input characteristics that make them regularizable (with $\rho(\widetilde{A}) = 0.998$ and $\rho(\widetilde{A}_{\text{new}}) = 0.927$ for ND-PA mode, and $\rho(\widetilde{A}) = 0.998$ and $\rho(\widetilde{A}_{\text{new}}) = 0.932$ for ND-UA mode). The resulting state trajectories for ND-PA and ND-UA modes are demonstrated in Figures 4.5 and 4.6, respectively. Without DGR, the norm of the state $\|\boldsymbol{x}_t\|$ would grow rapidly (red curve) as the unknown system is unstable and the original control laws fail.[10] As the plots suggest, with DGR in the feedback loop (with the choice of $\alpha = 5 \times 10^{-7}$),[11] The

[10]Since the LQR solution, in general, may have small stability margins for general parameter perturbations [95].

[11]The positive choice for α adjusts the compromise between state regulation and reducing the 2-norm of the input. This may lead to a larger upper bound on the state regulation, especially when the system is



(a) Longitudinal control

(b) Lateral-directional control

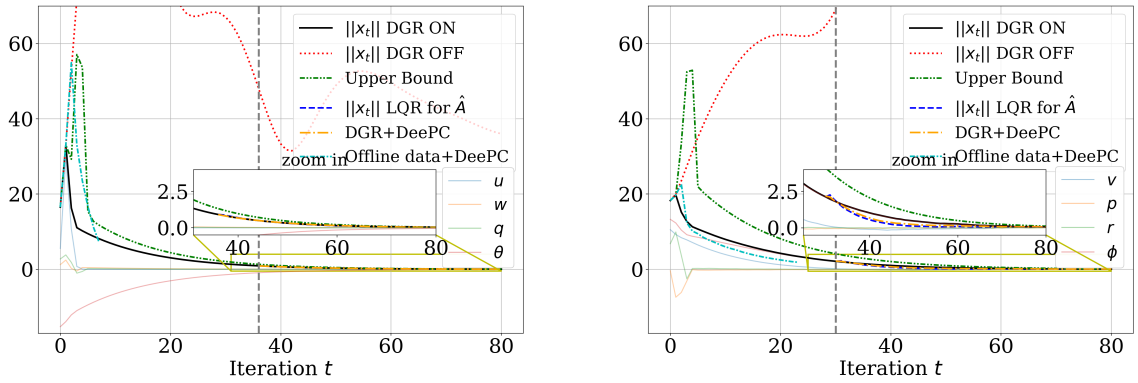
Figure 4.5: The state trajectory of X-29 in ND-PA mode with and without DGR.

unstable modes can be suppressed, resulting in the stabilization of the system (the norm of the states in this case is demonstrated in black, and each state is depicted in faded color).

Up to iteration $t = 36$ for longitudinal and $t = 30$ for lateral directional dynamics (shown with vertical dashed-line), enough data is generated in order to estimate the new system dynamics, or apply any other data-driven control using the data, (safely) generated by DGR up to this point. In what follows, we first showcase the complementary utility of DGR for identification and control; we then illustrate how it can also be incorporated for data-driven control.

In particular, the data is informative enough to identify system parameters through least squares denoted by $\hat{A}$. Therefore, one stopping criterion–which is also used here–is the point where the estimate of system parameters $\hat{A}$ has converged. Then, one can replace DGR with a closed-loop infinite horizon LQR controller with some cost-weights Q and R, which is obtained using the new estimate of the system dynamics. Here we set $Q = I$ and $R = 10^{-7}$ in order to make it comparable to the one-step quadratic cost used for DGR. In contrast to the original unstable LQR controller (red curve), it is shown that the new LQR controller for

unstable.



(a) Longitudinal control

(b) Lateral-directional control

Figure 4.6: The state trajectory of X-29 in ND-UA mode with and without DGR.

$\hat{A}$ (blue curve) is stabilizing since we now have a more accurate estimate of the (perturbed) system parameters using the data generated safely by DGR in the loop.

Next, while the DGR is still in effect, the generated data matrix is not ill-conditioned and thus can be utilized to implement a data-driven control algorithm from that point onward. Due to the presence of noise and uncertainty, we have implemented the regularized version of Data-driven Model Predictive Control (MPC) as in [42, 46] with parameters $T_{\text{ini}} = 1$, $N = 4$, $Q = 400I$, $R = 0.05I$, $\lambda_\sigma = 10^4$ and $\lambda_g = 1$ for both dynamics, where the input is persistently exciting. With DGR, after enough data has been generated for each dynamics, the data-driven MPC algorithm is initiated; the norm of the corresponding state vector is depicted in yellow dash-dotted line labeled as "DGR+DeePC."

On the other hand, one could consider implementing the data-driven MPC without DGR. However, this would require offline data which is not available a priori. Nonetheless, just for the purpose of comparison, this has been implemented based on *offline data* obtained from the original unstable plant. The resulting norm of the state has been depicted in cyan labeled as "Offline data+DeePC". Due to the ill-posed condition of the data matrix obtained from an unstable plant, it is observed that the practical tuning of the parameters could be

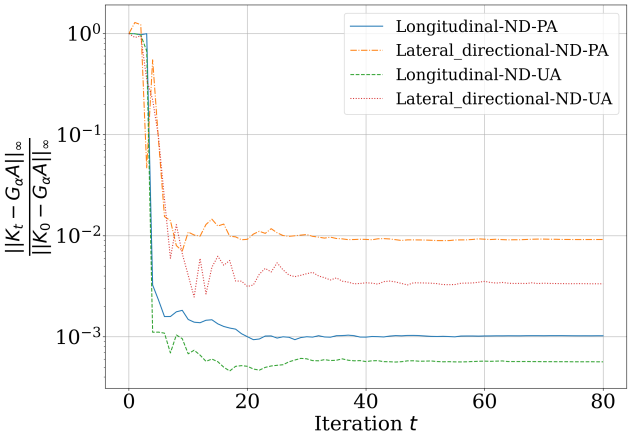


Figure 4.7: The convergence behavior of DGR algorithm for longitudinal and lateral-directional dynamics for both flight modes.

problematic. This is due to the fact that maintaining the stability and feasibility of the resulting convex optimization problems is challenging due to conditioning in the dataset (for similar observations, see, for example,v [41]). These trajectories are terminated whenever the corresponding optimization problem is not numerically solvable/feasible. We have used the CVXPY package for solving the convex programs derived in all these cases [96].

Finally, the convergence of DGR algorithm in terms of the designed controller K_t is illustrated in Figure 4.7, where the values are seen to be well behaved after 30 to 40 iterations from the noisy dynamics. Furthermore, we note that as the process noise $\boldsymbol{\omega}_t$ decreases, the error for K_t tends to zero; in the absence of noise, the limiting error is in fact negligible.

In these examples, the bound derived in Theorem 4.8–that requires the knowledge of A_{new}– is plotted in green for comparison. The behavior of the bound follows our observations in Remark 4.4; the bound increases as the algorithm initially tries to "detect" the unstable modes, followed by suppressing these modes for regulation. We finally note that for large enough iterations, the rate of change of the upper bound is dictated by $\rho(\widetilde{A}_{\text{new}})$, which in this case, is slightly less than one.[12]

[12]The code for these simulations can be found at `https://github.com/shahriarta/Data-Guided-Regulation`.

Chapter 5

Data-Guided Regulator for Adaptive Nonlinear Control

In this chapter, we extend the concept of "regularizability," as introduced in Section 4.3, to a class of partially unknown nonlinear systems, where time-varying uncertainty is modeled as an unknown component of the system dynamics. This investigation is motivated by the need for autonomous operations in safety-critical scenarios, where rapid adaptation to new environmental factors is crucial for maintaining safety. Notable examples include aircraft executing emergency landings under adverse weather conditions and agile quadrotors navigating through gates with low clearance amidst dynamic and strong winds [97].

In Section 5.3, we utilize linear matrix inequalities (LMIs) to characterize the notion of rapid-regularizability for a linear time-varying (LTV) representation of the nonlinear system with local Lipschitz continuous higher-order terms. We then elucidate the relationship between this notion and the spectral properties of the underlying LTV system.

Subsequently, in Section 5.4, the Data-Guided Regulator for Adaptive Nonlinear Control (DG-RAN) algorithm is introduced. This online iterative synthesis procedure employs discrete time-series data from a single trajectory to regulate system states while simultaneously generating informative data for identifying the dynamics of the disturbance.

Finally, an illustrative example is presented in Section 5.5 to demonstrate the practical application of the DG-RAN algorithm.

5.1 Review of Past Work

A rich body of literature has analyzed classical adaptive control algorithms' stability and convergence properties for continuous-time dynamical systems. Such studies include the use of PI (proportional-integral) controllers [98] for a class of linear time-varying systems to

guarantee (I) **infinite-time** convergence of the tracking error, i.e., the difference between actual and nominal states $e(t) = x(t) - \bar{x}(t)$, to the origin, for any constant exogenous disturbance (denoted by w), (II) **infinite-time** convergence of the tracking error $e(t)$ to a bound which is proportional to the bound on the magnitude of the rate of the exogenous signal $\frac{d}{dt}w(t)$. Furthermore, initial work in adaptive control of nonlinear systems used Lyapunov-like stability arguments and the certainty equivalence principle to construct stabilizing adaptive feedback policies for feedback linearized systems with matched uncertainties [29, 99–102], i.e., those that can be directly canceled through control. However, the certainty equivalence principle cannot be employed when the model uncertainties are outside the span of the control input, i.e., unmatched uncertainty. Departure from certainty equivalence significantly complicates the design process for a nonlinear system, as the controller must either anticipate or be robust to transients in the parameter estimates. Our contribution in Chapter 4 focused on studying the input "effectiveness" as it relates to **finite-time** regulation for a class of partially unknown linear systems, where the unknown part of the system can be modeled as unmatched uncertainty.

Many recent studies aim to integrate ideas from learning, optimization, and control theory to design and analyze adaptive controllers using learning-theoretic metrics. In contrast to the classical setting of adaptive nonlinear control, online Reinforcement Learning (RL) algorithms operate in discrete time and often come with *finite-time* [103–107] or dynamic regret bounds [108–110]. The most well-studied problem for continuous control in RL is the LQR problem with unknown dynamics. For LQR, both upper and lower bounds achieving $\sqrt{T}$ regret are available ([56, 58, 59, 111–113]). Results that extend beyond the classic LQR problem are not complete but rapidly growing. Recently, [107] showed $\sqrt{T}$ regret bounds in the finite horizon episodic setting for dynamics of the form $x_{t+1} = A\phi(x_t, u_t) + w_t$ where A is an unknown operator and ϕ is a known feature map. However, their algorithm is generally not tractable to implement. Such regret bounds provide a quantitative rate at which the control performance of the online algorithm approaches the performance of an oracle equipped with hindsight knowledge of the uncertainty. In such scenarios, the controller does not necessarily

guarantee the stability or robustness of the synthesized system; instead, the goal is to match the performance of the oracle for a well-behaved dynamical model.

Most learning-based studies focus on two restricting scenarios when dealing with uncertain dynamical systems. The first category of work focuses on a finite-horizon episodic setting [105–107], which might not be directly applicable to online safety-critical systems such as online flight control. The second category of work either assumes fully actuated systems [97] or examines the matched uncertainty setting [104]. To the best of our knowledge, the work presented in [114] is most relevant to this chapter. The primary contribution of [114] lies in introducing a novel direct adaptive control framework that effectively utilizes the certainty equivalence principle. This framework enables the design of stabilizing adaptive controllers for general nonlinear systems with unmatched uncertainties. Specifically, the work mentioned above proposes an adaptive controller by first defining the unmatched control Lyapunov function (clf), a family of clfs parameterized over all possible models. Subsequently, the framework dynamically adjusts the adaptation rate of the *online estimation* of the *constant* unknown parameter to mitigate the influence of estimation transients on stability. However, it is important to note several restrictive assumptions of this method. First, the proposed adaptive control framework strictly caters to nonlinear systems with *constant unknown parameters*. Second, it relies on the existence of a family of differentiable Lyapunov functions or a smooth manifold $\mathcal{M}$ that describes the contracting region for the system with any parameter estimates at all times t. These assumptions impose limitations as the existence of a Lyapunov function for a nonlinear system is generally non-trivial, and the requirement of differentiable Lyapunov functions for any variable estimate is highly restrictive. Additionally, recomputing the adaptive reference model for each new parameter estimate adds complexity and computational overhead. While the work in [114] presents an innovative adaptive control framework, the above assumptions restrict its feasibility for specific applications, particularly in dynamic environments where time-varying uncertainties are prevalent.

This chapter's contribution offers a remedy for the shortcomings mentioned above and is mainly adapted from [2].

5.2 Online Regulation of Unstable Nonlinear Systems

We examine the control of an uncertain dynamical system set to follow a predefined trajectory. In this direction, consider the uncertain nonlinear dynamics of the form,

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)) + F\dot{\theta}(t), \quad t \in [t_0, t_f], \tag{5.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ is the system state, $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$ is the control input, $F \in \mathbb{R}^{n_x \times n_\theta}$ is a known basis function for the environmental disturbance, and $\theta(t) \in \mathbb{R}^{n_\theta}$ is an unknown time-varying parameter. By employing handpicked or learned basis functions, systems characterized by non-parametric or nonlinearly parameterized uncertainties can be transformed into the form represented by (5.1). One example of such a system is an airplane ascending and landing in the presence of an unknown wind profile. We assume that $f(\boldsymbol{x}(t), \boldsymbol{u}(t))$ is locally Lipschitz uniformly and at least once differentiable, with measurable state $\boldsymbol{x}$. Although the dynamics (5.1) is continuous, we can only measure system states and control inputs at discrete times. Hence, we adopt the following assumption on the measured data.

Assumption 5.1. *The system states can be measured at constant (positive) length intervals $\delta t \in \mathbb{R}$. Hence at time t, we have a streaming data set $\mathcal{D}_t = \{x_i, u_i\}_{i=0}^{n_t}$ of size $n_t = \lfloor t/\delta t \rfloor$.*

Assume that $\{\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t)\}_{t=t_0}^{t_f}$ is the nominal (reference) trajectory that satisfies the dynamics in the absence of uncertainty, i.e., $\dot{\bar{\boldsymbol{x}}}(t) = f(\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t))$. For some initial condition $\boldsymbol{x}_0$, define,

$$\eta(t) \coloneqq \boldsymbol{x}(t) - \bar{\boldsymbol{x}}(t), \tag{5.2a}$$

$$\widetilde{\theta}(t) \coloneqq \theta(t) - \hat{\theta}(t), \tag{5.2b}$$

$$\xi(t) + \boldsymbol{z}(t, \mathcal{D}_t) \coloneqq \boldsymbol{u}(t) - \bar{\boldsymbol{u}}(t). \tag{5.2c}$$

$\eta(t)$ is the deviation from the nominal trajectory, $\hat{\theta}(t)$ is the estimate of the variable $\theta(t)$, and $\widetilde{\theta}(t)$ is the estimation error at time t. By adding and subtracting $f(\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t))$ to (5.1) and using a first-order Taylor series expansion around the reference trajectory, we can rewrite

this equation in terms of the variables in (5.2) as,

$$\dot{\eta}(t) = A(t)\eta(t) + B(t)\xi(t) + \delta g\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right) + B(t)\boldsymbol{z}(t, \mathcal{D}_t) + F\dot{\theta}(t), \tag{5.3}$$

where $A(t)$ and $B(t)$ are the gradients of f evaluated along the nominal state and control trajectories, and

$$\delta g\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right) = g(\boldsymbol{x}(t), \boldsymbol{u}(t)) - g(\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t))$$

represents the higher-order (nonlinear) terms.

Assumption 5.2. *The higher order (nonlinear) term, $g\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right)$, is locally Lipschitz bounded.*

A practical approach for developing such guidance-control design for nonlinear systems is via an offline robust control (with state denoted by $\xi(t)$) by considering the uncertainty bound known a priori [3]. However, in the online setting, the robust controller might fail when the nominal uncertainty does not capture the effects of environmental uncertainties. Therefore, one needs an online data-driven approach (with state denoted by $\boldsymbol{z}(t, \mathcal{D}_t)$) to update the control policy in order to satisfy the corresponding operational constraints.

In this direction, we assume the offline robust controller has been designed using the Lyapunov theory/linear matrix inequalities. Suppose that $\mathcal{X}_{\mathcal{F}} \in \mathbb{R}^{n_x}$ and $\mathcal{U}_{\mathcal{F}} \in \mathbb{R}^{n_u}$ are the (possibly nonconvex) sets of feasible state and control vectors. A Lyapunov function can be used, via its invariant set, to characterize nearby feasible trajectories around the nominal $\{\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t)\}_{t=t_0}^{t_f}$. Such sets, designated as funnels are sets of the form,

$$\mathcal{F}(t) \subseteq \mathcal{X}_{\mathcal{F}} \times \mathcal{U}_{\mathcal{F}};$$

$\mathcal{F}(t)$ consists of time-varying state and control trajectories that remain invariant and are confined within their respective feasible regions.

For the original dynamics $\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t))$, one can use Lyapunov theory, as described in Section 2.4.1, to design a robust feedback controller $\xi(t)$, to ensure invariance properties of $\mathcal{F}(t)$. However, as illustrated in Figure 5.1, the control policy may falter in the face of

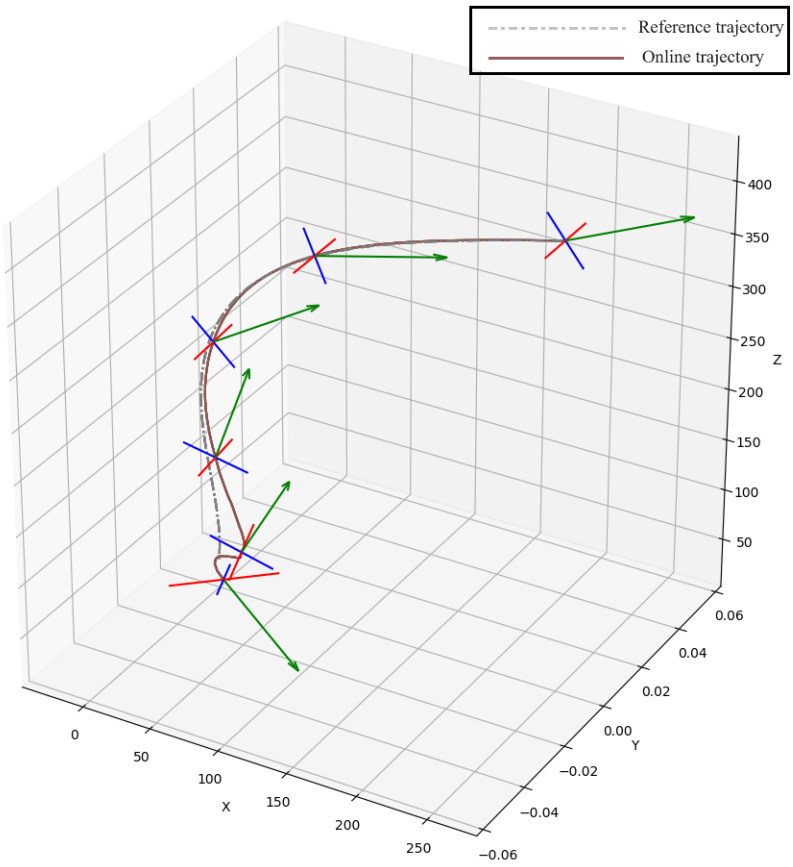


Figure 5.1: Landing sequence of a six-degree-of-freedom (6-DOF) vehicle under a power descent guidance algorithm, showcasing the performance amidst environmental disturbances (details in Section 5.5). Despite the robust controller's design to handle system nonlinearities and ensure convergence within a specific operational envelope, the unaccounted-for disturbances led to a failure at the touchdown phase, highlighting the challenges in real-world application scenarios.

environmental uncertainties that were not previously considered, highlighting the timeliness of our contributions in this work.

5.3 Rapidly-Regularizable Nonlinear Systems

We first introduce the notion of "rapid-regularizablity" for a class of nonlinear dynamical systems, then provide the necessary and sufficient conditions for the corresponding data-driven controller. This section concludes with the main result.

Definition 5.1. Denote $\bar{Q}(t)$ and $K(t)$ for $t \in [t_0, t_f]$ as the solution to Problem 2.2. The uncertain dynamical system of the form (5.3) is called "rapidly regularizable" if there exists a linear feedback controller $z(t, \theta) = \tilde{\mathbf{K}}_*(t)\theta(t)$ such that for all $t \in [t_0, t_f]$ the following holds.

$$\eta(t) \in \mathcal{E}_{\bar{Q}(t)}, \quad \xi(t) + z(t, \theta) \in \mathcal{U}_{\mathcal{F}}, \qquad \text{where } \xi(t) = K(t)\eta(t). \tag{5.4}$$

By definition, depending on the dynamics of the variable $\theta(t)$, there should exist a feedback policy $z(t, \theta)$ that further assists the existing robust controller $\xi(t)$ in keeping the system in the invariant state space $\mathcal{E}_{\bar{Q}} \subseteq \mathcal{X}_{\mathcal{F}}$. The following theorem provides a sufficient condition for the nonlinear system of the form (5.3) to be rapidly regularizable.

Theorem 5.3. *Denote $\bar{Q}(t)$, $\bar{Y}(t) = \mathbf{K}(t)\bar{Q}(t)$ and $M_\gamma(t)$ for $t \in [t_0, t_f]$ as the solution to Problem 2.2. Suppose the dynamics of $\theta(t)$ is known and $\dot{\theta}(t) = \tilde{S}\theta(t)$ where $\theta(t = 0) = \theta_0$. If there exists a feedback controller of the form $z(t, \theta) = \tilde{\mathbf{K}}_*(t)\theta(t)$, a matrix-valued function of time $\tilde{M}(t)$ and scalars $\lambda(t)$, $\nu(t), \beta(t)$ that for all time $t \in [t_0, t_f]$ solves the optimization problem laid out in equation (5.5), then the uncertain dynamical system of the form (5.3) is "rapidly regularizable".*

$$\min_{\tilde{\mathbf{K}}_*(\cdot),\tilde{M}(\cdot),\nu(\cdot),\beta(\cdot),\lambda(\cdot)} \quad \sigma_*(\cdot) \tag{5.5}$$

$$\textbf{\textit{subject to}} \quad 0<\beta,\ \ 0<\nu,\ \ 0<\lambda,\ \ \tilde{M}\in\mathcal{M}_{\phi,\Psi}, \tag{5.5a}$$

$$\begin{pmatrix} \bar{Q}A^{\intercal}+A\bar{Q}+T_2 & \gamma E & \nu\tilde{\gamma}E & T_3 \\ \gamma E^{\intercal} & -\beta I & 0 & 0\ \ 0 \\ \nu\tilde{\gamma}E^{\intercal} & 0 & -\nu I & 0\ \ 0 \\ * & * & * & T_4 \end{pmatrix} \preceq 0, \tag{5.5b}$$

$$z(t,\theta)+\xi(t)\in\mathcal{U}_{\mathcal{F}}, \tag{5.5c}$$

Note that $T_2 \coloneqq \mathbf{Y}^{\intercal}B^{\intercal}+B\mathbf{Y}-\dot{\bar{Q}}+\lambda\bar{Q}+\beta\bar{Q}C_{cl}^{\intercal}C_{cl}\bar{Q}$, $T_3 \coloneqq \begin{bmatrix} B\tilde{\mathbf{K}}_*\theta+F\tilde{S}\theta & 0 \end{bmatrix}$, *and* $T_4 \coloneqq \begin{pmatrix} -\lambda\sigma_* & \theta^{\intercal}\tilde{\mathbf{K}}_*^{\intercal}G \\ G\tilde{\mathbf{K}}_*\theta & -\nu I \end{pmatrix}$.

Proof. For the uncertain system (5.3), consider the Lyapunov function

$$V_\theta(t)=\eta(t,\theta)^{\intercal}\bar{Q}^{-1}(t)\eta(t,\theta),$$

defined by the matrix-valued function of time $\bar{Q}(t)$. By Lyapunov theory, if $z(t,\theta)=\tilde{\mathbf{K}}_*(t)\theta(t)$ satisfies conditions in (5.6), then the system states remain inside the funnel.

$$\dot{V}_\theta(t)<0, \quad \text{for all } \ 0\le\sigma_*(t)\le V_\theta(t), \quad \begin{pmatrix} \delta\tilde{p}(t) \\ \tilde{q}(t) \end{pmatrix}^{\intercal} \tilde{M}(t) \begin{pmatrix} \delta\tilde{p}(t) \\ \tilde{q}(t) \end{pmatrix} \ge 0 \tag{5.6}$$

We again use structured nonlinearities and define the set of nonlinear functions $\tilde{\phi}_i:\mathbb{R}^{n_{\tilde{q}_i}}\to\mathbb{R}^{n_{\tilde{p}_i}}$ such that,

$$\tilde{q}(t)\coloneqq G_i\tilde{K}_*(t)\theta(t) \tag{5.7a}$$

$$\delta\tilde{p}(t)\coloneqq\tilde{\phi}(t,\tilde{q}(t))=\phi(t,q(t)+\tilde{q}(t))-\phi(t,q(t)), \tag{5.7b}$$

where $q(t)$ and $\delta p(t)$ are defined in (2.17), and by the definition of $\delta\tilde{p}(t)$ we have

$$\delta g(\boldsymbol{x}(t),\boldsymbol{u}(t))=\phi(t,q(t)+\tilde{q}(t))-\phi(t,\bar{q}(t))=\delta\tilde{p}(t)+\delta p(t)\,.$$

Let $\chi = \begin{bmatrix} \eta & p & \tilde{p} & \mathbf{1} \end{bmatrix}^\intercal$. Expanding (5.6) in terms of the close loop system leads to,

$$\chi^\intercal \begin{pmatrix} A^\intercal \bar{Q}^{-1} + \bar{Q}^{-1} A + T_5 & \bar{Q}^{-1} E & \bar{Q}^{-1} E & \bar{Q}^{-1} (B\tilde{\mathbf{K}}_* \theta + F\tilde{S}\theta) \\ * & 0 & 0 & 0 \end{pmatrix} \chi \leq 0, \tag{5.8}$$

for all,

$$\begin{pmatrix} \eta \\ \mathbf{1} \end{pmatrix}^\intercal \begin{pmatrix} \bar{Q}^{-1} & 0 \\ 0 & -\sigma_* \end{pmatrix} \begin{pmatrix} \eta \\ \mathbf{1} \end{pmatrix} > 0 \,,$$

$$\begin{pmatrix} \delta\tilde{p} \\ \tilde{q} \end{pmatrix}^\intercal \tilde{M} \begin{pmatrix} \delta\tilde{p} \\ \tilde{q} \end{pmatrix} \geq 0 \,, \quad \begin{pmatrix} \delta q \\ \delta p \end{pmatrix}^\intercal M_\gamma \begin{pmatrix} \delta q \\ \delta p \end{pmatrix} \geq 0 \,,$$

where $T_5 \coloneqq \frac{d}{dt}\left(\bar{Q}^{-1}\right) + \mathbf{K}^\intercal B^\intercal \bar{Q}^{-1} + \bar{Q}^{-1} B\mathbf{K}$.

We further consider $\tilde{M} = \mathrm{diag}\left([-I,\ \tilde{\gamma}^2 I]\right)$ as a valid local multiplier matrix, where $\tilde{\gamma}$ is a local Lipschitz constant for the nonlinear function $\tilde{\phi}$ over the set Ψ . By using the S-procedure, the condition (5.8) holds if there exists scalars $\beta, \tilde{\beta}, \lambda \geq 0$ such that

$$\begin{pmatrix} A^\intercal \bar{Q}^{-1} + \bar{Q}^{-1} A + T_5 & \gamma \bar{Q}^{-1} E & \tilde{\gamma} \bar{Q}^{-1} E & \bar{Q}^{-1} \left(B\tilde{\mathbf{K}}_* \theta + F\theta\right) & 0 \\ \gamma E^\intercal \bar{Q}^{-1} & -\beta I & 0 & 0 & 0 \\ \tilde{\gamma} E^\intercal \bar{Q}^{-1} & 0 & -\tilde{\beta} I & 0 & 0 \\ * & * & * & T_6 & \end{pmatrix} \preceq 0, \tag{5.9}$$

where we overuse the notation to define T_5, and define T_6 as

$$T_5 \coloneqq \frac{d}{dt}\left(\bar{Q}^{-1}\right) + \mathbf{K}^\intercal B^\intercal \bar{Q}^{-1} + \bar{Q}^{-1} B\mathbf{K} + \lambda \bar{Q}^{-1} + \beta C_{cl}^\intercal C_{cl}, \qquad T_6 \coloneqq \begin{pmatrix} -\lambda\sigma_* & \theta^\intercal \tilde{\mathbf{K}}_*^\intercal G \\ G\tilde{\mathbf{K}}_* \theta & -\tilde{\beta}^{-1} I \end{pmatrix}.$$

Pre- and post-multiplying (5.9) by $\mathbf{diag}\{\bar{Q}, I, \tilde{\beta}^{-1}, I, I\}$ and a change of variable $\nu = \tilde{\beta}^{-1}$ now yields the desired result.

□

To address the presence of the unknown disturbance in the guidance-feedback paradigm, we first model its dynamics in terms of a linear system of the form $\dot{\theta}(t) = \tilde{S}\theta(t)$ with initial condition θ_0. Now we let $u(t) = \bar{u}(t)+\xi(t)+z(t,\theta)$ be the control input, where $\xi(t) = \mathbf{K}(t)\eta(t)$. Then, the system dynamics in (5.3) can be represented as,

$$\dot{\eta}(t) = [A(t) + B(t)\mathbf{K}(t)]\,\eta(t) + \delta g(\boldsymbol{x}(t), \boldsymbol{u}(t)) + B(t)\left[B(t)^{\dagger}F\tilde{S}\theta(t) + z(t,\theta)\right] + \Pi_{\mathcal{R}(B(t))^{\perp}}F\tilde{S}\theta(t). \tag{5.10}$$

Note the last two terms represent how the additional (data-driven) corrective action can be expressed in terms of its components in the range space of $B(t)$ and its orthogonal complement. Further, let $z(t,\theta) \coloneqq -B^{\dagger}F\tilde{S}\theta(t) + \hat{\mathbf{K}}(t)\theta(t)$, where $\hat{\mathbf{K}}(t)$ is yet to be designed. We can rewrite (5.10) as,

$$\dot{\eta}(t) = A_{cl}(t)\eta(t) + E\delta p(t) + E\delta\tilde{p}(t) + B(t)\hat{\mathbf{K}}(t)\theta(t) + \Pi_{\mathcal{R}(B(t))^{\perp}}F\tilde{S}\theta(t), \tag{5.11}$$

where $A_{cl}(t) = A(t)+B(t)\mathbf{K}(t)$, and $\delta p(t)$ and $\delta\tilde{p}(t)$ are defined in (2.17) and (5.7), respectively. Note that the signals $\Pi_{\mathcal{R}(B(t))^{\perp}}F\tilde{S}\theta(t)$ and $B(t)\hat{\mathbf{K}}(t)\theta(t)$ are now orthogonal. This implies that the control signal $z(t,\theta)$ would not directly affect the part of dynamics that is generated by $\Pi_{\mathcal{R}(B(t))^{\perp}}F\tilde{S}\theta(t)$. Moreover, $z(t,\theta)$ causes additional disturbance in terms of $\delta g(\boldsymbol{x}(t), \boldsymbol{u}(t))$ captured by $\delta\tilde{p}(t)$. The local Lipschitz bounds for $\delta g(\boldsymbol{x}(t), \boldsymbol{u}(t))$ drove from (2.17) and (5.7) are only accurate when $\eta(t) \in \mathcal{E}_{\bar{Q}(t)}$. As such, to have even the possibility of achieving online performance for keeping the system states in the feasible region, we require that the part of the system described by $E\delta\tilde{p}(t) + B(t)\hat{\mathbf{K}}(t)\theta(t) + \Pi_{\mathcal{R}(B(t))^{\perp}}F\tilde{S}\theta(t)$ be bounded for all $t \in [t_0, t_f]$. This observation thereby results in the following proposition.

Proposition 5.4. *The parametrized nonlinear system in (5.11) is rapidly regularizable if and only if the part of the system described by $\vartheta(t)$ is bounded such that $\|\vartheta(t)\|^2 < \sigma^*(t)$, where*

$$\vartheta(t) = E\delta\tilde{p}(t) + B(t)\hat{\mathbf{K}}(t)\theta(t) + \Pi_{\mathcal{R}(B(t))^{\perp}}F\tilde{S}\theta(t)\,, \quad \textit{and} \quad \hat{\mathbf{K}}(t) = \tilde{\mathbf{K}}_*(t) + B^{\dagger}F\tilde{S}$$

Furthermore, the ellipsoid

$$\mathcal{E} \coloneqq \left\{\eta : \eta^{\mathrm{T}}\bar{Q}^{-1}\eta \leq \|\vartheta(\cdot)\|^2_{\Delta t}\right\} \qquad \textit{with} \qquad \|\vartheta(\cdot)\|_{\Delta t} = \sup_{t_f \geq t \geq t_0} \|\vartheta(t)\|$$

is invariant and attractive.

5.4 Data-Guided Regulator for Adaptive Nonlinear Control

We now turn our attention to data-guided regulation, in the context of nonlinear systems. In this direction, suppose the dynamics of the disturbance $\theta(t)$ satisfies,

$$\dot{\theta}(t) = \tilde{S}\theta(t), \quad \theta_0 \in \mathbb{R}^{n_\theta}, \tag{5.12}$$

$$\text{where} \quad \theta_{n_t} = \theta(n_t\delta t) = S^{n_t}\theta_0\,, \quad \text{and} \quad S = \exp(\delta t\tilde{S})\,, \quad n_t \in \{0, 1, 2, \cdots\}. \tag{5.12a}$$

Note that in our setting, the parameters $\tilde{S}$ and θ_0, describing the dynamics of the disturbance, are unknown. For a "rapidly regularizable" system, the primary focus of this section is devising an online, data-driven feedback controller to further assist the offline-computed robust controller in regulating the system states. To this end, we propose an iterative procedure for updating the feedback gain $\tilde{\mathbf{K}}(t)$ at each time $t = n_t\delta t$. We provide an optimization problem for obtaining the corresponding sub-optimal data-driven controller and conclude the section with our main result.

Given a data set $\mathcal{D}_t = \{x_i, u_i\}_{i=0}^{n_t}$ available at time t, we are interested in characterizing the algorithmic map $\mathcal{D}_t \to \tilde{\mathbf{K}}(t)$, where $\tilde{\mathbf{K}}(t)$ is the data-driven regulating feedback gains for the uncertain dynamical system (5.3). We propose an iterative algorithm for updating the feedback gain (policy) based on the available (closed loop) data. The corresponding synthesis procedure is detailed in Algorithm 3.

Our goal is to use the measured data $\mathcal{D}_t = \{x_i, u_i\}_{i=0}^{n_t}$ to update estimates for θ_0 and $S = \exp(\tilde{S}\delta t)$. Specifically, at time "t_0", the algorithm measures x_0; analogously, ${n_t}^{th}$ measurement, x_{n_t}, is obtained at $t = n_t\delta t$, where $n_t \in \{1, 2, \cdots\}$. We define the observation vector as,

$$y_{n_t} \coloneqq F^\dagger\left[\Delta x_{n_t} - \Delta f_{n_t}\right], \tag{5.13}$$

---

[1]The stopping criterion can be application-specific. For instance, generating k_1 linearly independent data is sufficient to identify modes of the disturbance system affecting the dynamics.

Algorithm 3 Data-Guided Regulator for Adaptive Nonlinear Control (DG-RAN)

Initialization (at $t = 0$)

  Fix δt; Measure $\boldsymbol{x}_0$; set $\tilde{K} = 0$ and $n_t = 0$

  Set $\mathcal{Y}_{n_t} = \begin{pmatrix} \end{pmatrix}$, and $\mathcal{Z}_{n_t} = \begin{pmatrix} \end{pmatrix}$

while stopping criterion not met[1]

  If $n_t < \lfloor t/\delta t \rfloor$

    Set $n_t = \lfloor t/\delta t \rfloor$

    Measure $\boldsymbol{x}_{n_t}$

    Set $y_{n_t} = F^\dagger \left(\Delta x_{n_t} - \Delta f_{n_t} \right)$

    If $n_t == 1$

      Set $\mathcal{Z}_0 = \begin{pmatrix} y_1 \end{pmatrix}$

    else:

      Update $\mathcal{Y}_{n_t} = \begin{pmatrix} \mathcal{Y}_{n_t-1} & y_{n_t} - \mathcal{Y}_{n_t-1} \mathcal{Z}^\dagger_{n_t-3} y_{n_t-1} \end{pmatrix}$

$$\mathcal{Z}_{n_t-1} = \begin{pmatrix} \mathcal{Z}_{n_t-2} & \Pi_{\mathcal{R}\left(\mathcal{Z}_{n_t-1}\right)^\perp} y_{n_t} \end{pmatrix}$$

$$\hat{S}_{n_t} = \mathcal{Y}_{n_t} \mathcal{Z}^\dagger_{n_t-2}, \quad \hat{\theta}^\circ_{n_t} = \left(\hat{S}_{n_t} - I \right)^\dagger y_1$$

      Update $\tilde{\mathbf{K}}(t)$ by solving (5.19)

where $\Delta x_{n_t} \coloneqq x_{n_t} - x_{n_t-1}$ and $\Delta f_{n_t} \coloneqq \frac{\delta t}{2}\left(f(x_{n_t}, u_{n_t}) + f(x_{n_t-1}, u_{n_t-1})\right)$. When $n_t > 1$, the algorithm updates the estimates for θ_0 and $S = \exp(\tilde{S}\delta t)$ as follows:

$$\hat{S}_{n_t} = \mathcal{Y}_{n_t}\mathcal{Z}_{n_t-2}^{\dagger}, \qquad \hat{\theta}_{n_t}^{\circ} = \left(\hat{S}_{n_t} - I\right)^{\dagger} y_1, \tag{5.14}$$

where,

$$\mathcal{Y}_{n_t} = \begin{bmatrix} y_2 & y_3 - \mathcal{Y}_2\mathcal{Z}_0^{\dagger}y_2 & \cdots & y_{n_t} - \mathcal{Y}_{n_t-1}\mathcal{Z}_{n_t-3}^{\dagger}y_{n_t-1} \end{bmatrix}, \tag{5.14a}$$

$$\mathcal{Z}_{\ell} = \begin{bmatrix} \boldsymbol{z}_0 & \boldsymbol{z}_1 & \cdots & \boldsymbol{z}_{\ell} \end{bmatrix}, \quad \boldsymbol{z}_{\ell} \coloneqq \Pi_{\mathcal{R}(\mathcal{Z}_{\ell-1})^{\perp}} y_{\ell+1}, \quad \text{for } \ell \geq 1, \quad \boldsymbol{z}_0 = y_1, \tag{5.14b}$$

Note that for $l \geq 1$, we defined $\boldsymbol{z}_{\ell}$ as the orthogonal projection of the observation vector $y_{\ell+1}$ onto a linear subspace perpendicular to the range of $\mathcal{Z}_{\ell-1}$.

In what follows, we provide the performance analysis for DG-RAN in the general setting. DG-RAN is particularly relevant when $n_t \leq r$, where r denotes the dimension of the underlying disturbance system $\dot{\theta}(t)$ affecting system states $x(t)$. We examine the effects of DG-RAN in terms of the informativity of generated data in Section 5.4.1, and provide convergence analysis of the algorithm in Section 5.4.2.

5.4.1 Informativity of the DG-RAN Generated Data

In the sequel, we show that DG-RAN generates linearly independent data, describing the observable modes of the disturbance dynamics $\left(\Pi_{\mathcal{R}(F^{\intercal})}S\right)$ from the state-trajectory measurements $\mathcal{D}_t = \{x_i, u_i\}_{i=0}^{n_t}$. We refer to this as the informativity of data and then proceed to make a connection between this independence structure and the number of excited modes of the disturbance dynamics $\left(\Pi_{\mathcal{R}(F^{\intercal})}S\right)$ affecting the system states.

Lemma 5.5. *Let θ_0 excites $k_1 + k_2$ modes of S such that k_1 modes are in $\mathcal{R}(F^{\intercal})$ and k_2 modes are in $\mathcal{R}(F^{\intercal})^{\perp}$. For all $n_t > 2$, $\{\boldsymbol{z}_0, \boldsymbol{z}_1, \cdots, \boldsymbol{z}_{n_t-3}\}$ is a set of "orthogonal" vectors (possibly including the zero vector). Furthermore, If the excited modes correspond to distinct eigenvalues of S where $\lambda_i \neq 1$ for $\boldsymbol{v}_i \in \mathcal{R}(F^{\intercal})$, then $[y_1,\ \mathcal{Y}_{n_t}]$, generated by Algorithm 3, is a set of linearly independent vectors for any positive number $m \in \mathbb{N}$ such that $m \leq k_1 \leq r$.*

Proof. The definition of $\boldsymbol{z}_\ell$ in (5.14b) implies that $\boldsymbol{z}_\ell \perp \mathcal{R}(\mathcal{Z}_{\ell-1})$ for all $\ell \geq 1$, and $\boldsymbol{z}_i \in \mathcal{R}(\mathcal{Z}_{\ell-1})$ for all $i = 1, \ldots, \ell-1$ and all $\ell \geq 1$. Thus, the first claim follows since $\{\boldsymbol{z}_0, \boldsymbol{z}_1, \cdots, \boldsymbol{z}_\ell\}$ consists of orthogonal vectors. For the second claim, without loss of generality, let $\lambda_1, \ldots, \lambda_{k_1}$ be the eigenvalues of S corresponding to the excited modes $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{k_1} \in \mathcal{R}(F^\intercal)$. Similarly, $\lambda_{k_1+1}, \ldots, \lambda_{k_1+k_2}$ be corresponding to $\boldsymbol{v}_{k_1+1}, \ldots, \boldsymbol{v}_{k_1+k_2} \in \mathcal{R}(F^\intercal)^\perp$. For $n_t \geq 1$ the trajectory generated by Algorithm 3 satisfies

$$\Delta x_{n_t} = \textstyle\int_{(n_t-1)\delta t}^{n_t \delta t} f(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)) d\tau + F S^{n_t-1} (S-I)\, \theta_0. \tag{5.15}$$

By estimating the integral in (5.15), we have

$$\left(F^\dagger F\right)(S)^{n_t-1}(S-I)\,\theta_0 = F^\dagger\left(\Delta x_{n_t} - \Delta f_{n_t}\right) + \epsilon_{n_t} = y_{n_t} + \epsilon_{n_t},$$

where $\epsilon_{n_t} = F^\dagger\left(\int_{(k-1)\delta t}^{k\delta t} f(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)) d\tau - \Delta f_k\right)$ is assumed to be negligible. Note that $F^\dagger F = V_r V_r^\intercal = \Pi_{\mathcal{R}(V_r)}$, and $\theta_0 = \sum_{i=1}^{k_1+k_2} \rho_i \boldsymbol{v}_i$, where ρ_i are some nonzero real coefficients. Therefore, for the n_t^{th} observation, we have

$$y_{n_t} = \Pi_{\mathcal{R}(V_r)}(S)^{n_t-1}(S-I)\,\theta_0 = \Pi_{\mathcal{R}(V_r)}\left(\sum_{i=1}^{k_1} \rho_i(\lambda_i)^{n_t-1}(\lambda_i-1)\boldsymbol{v}_i + \sum_{i=k_1+1}^{k_2} \rho_i(\lambda_i)^{n_t-1}(\lambda_i-1)\boldsymbol{v}_i\right).$$

Since $\mathcal{R}(V_r) = \mathcal{R}(F^\intercal)$ and $\boldsymbol{v}_{k_1+1}, \ldots, \boldsymbol{v}_{k_1+k_2} \in \mathcal{R}(F^\intercal)^\perp$, we have

$$y_{n_t} = \Pi_{\mathcal{R}(V_r)}(S)^{n_t-1}(S-I)\,\theta_0 = \sum_{i=1}^{k_1} \rho_i(\lambda_i)^{n_t-1}(\lambda_i-1)\boldsymbol{v}_i. \tag{5.16}$$

Let $\boldsymbol{w}_1 \coloneqq y_1$ and $\boldsymbol{w}_2 \coloneqq y_2$, and $\boldsymbol{w}_i$ be the i^{th} column of $\mathcal{Y}_{n_t}$. We claim that for $n_t \geq 3$ there exist scalar coefficients $\xi_1^{n_t}, \cdots, \xi_{n_t-1}^{n_t} \in \mathbb{R}$ such that,

$$\boldsymbol{w}_{n_t} \coloneqq y_{n_t} - \mathcal{Y}_{n_t-1}\mathcal{Z}_{n_t-3}^\dagger y_{n_t-1} = S\boldsymbol{z}_{n_t-2} = \sum_{i=1}^{k_1} \rho_i \left[(\lambda_i)^{n_t-1} - \sum_{j=1}^{n_t-2} \xi_j^{n_t-1}(\lambda_i)^j\right](\lambda_i-1)\boldsymbol{v}_i. \tag{5.17}$$

The proof of the last claim is by induction. By the definition of $\boldsymbol{z}_\ell$ in (5.14b), for $\ell \geq 1$ there exist (scalar) coefficients $\zeta_1^\ell, \cdots, \zeta_\ell^\ell \in \mathbb{R}$ such that $\boldsymbol{z}_\ell = y_{\ell+1} - \sum_{j=1}^{\ell} \zeta_j^\ell z_{j-1}$. For $n_t = 3$, we have that,

$$\boldsymbol{w}_3 = Sy_2 - Sy_1 y_1^\dagger y_2 = S\left(y_2 - \Pi_{\mathcal{R}(\mathcal{Z}_0)} y_2\right) = S\boldsymbol{z}_1 = \sum_{i=1}^{k_1} \rho_i\left[(\lambda_i)^2 - \zeta_1^1(\lambda_i)\right](\lambda_i-1)\boldsymbol{v}_i,$$

where the last equality is due to (5.16). By choosing $\xi_1^2 = \zeta_1^1$, we have shown that (5.17) holds for $n_t = 3$. Suppose that (5.17) holds for all $3, \ldots, n_t - 1$; it now suffices to show that this relation also holds for n_t. By substituting the hypothesis for $3, \ldots, n_t - 1$ into (5.17),

$$\begin{aligned}\boldsymbol{w}_{n_t} =& S y_{n_t-1} - S\mathcal{Z}_{n_t-3}\mathcal{Z}_{n_t-3}^{\dagger} y_{n_t-1} = S\left(y_{n_t-1} - \Pi_{\mathcal{R}\left(\mathcal{Z}_{n_t-3}\right)} y_{n_t-1}\right) = S\boldsymbol{z}_{n_t-2} \\ &= \sum_{i=1}^{k_1} \rho_i \left(\lambda_i\right)^{n_t-1} (\lambda_i - 1)\boldsymbol{v}_i - \sum_{j=1}^{n_t-2} \zeta_j^{n_t-2} S\boldsymbol{z}_{j-1} \\ &= \sum_{i=1}^{k_1} \rho_i \left(\lambda_i\right)^{n_t-1} (\lambda_i - 1)\boldsymbol{v}_i - \sum_{j=1}^{n_t-2} \zeta_j^{n_t-2} \sum_{i=1}^{k_1} \rho_i \left[(\lambda_i)^j - \sum_{\ell=1}^{j-1} \xi_\ell^{j-1} \left(\lambda_i\right)^\ell\right] (\lambda_i - 1)\boldsymbol{v}_i.\end{aligned}$$

Therefore, $\boldsymbol{w}_{n_t} = \sum_{i=1}^{k_1} \rho_i \left[\star\right] (\lambda_i - 1)\boldsymbol{v}_i$ where $\star$ replaces the expression,

$$(\lambda_i)^{n_t-1} - \sum_{j=1}^{n_t-2} \zeta_j^{n_t-2} (\lambda_i)^j + \sum_{j=1}^{n_t-2} \zeta_j^{n_t-2} \sum_{\ell=1}^{j-1} \xi_\ell^{j-1} (\lambda_i)^\ell.$$

By appropriate choices of $\xi_1^{n_t-1}, \cdots, \xi_{n_t-2}^{n_t-1} \in \mathbb{R}$, we can rewrite $\star = (\lambda_i)^{n_t-1} - \sum_{j=1}^{n_t-2} \xi_j^{n_t-1} \left(\lambda_i\right)^j$. This completes the proof of the claim in (5.17) by induction. Now, let $\widehat{\boldsymbol{w}} = \sum_{\ell=1}^{m} \gamma_{\ell-1} \boldsymbol{w}_\ell$ for some $\gamma_{\ell-1} \in \mathbb{C}$ and some $m \leq k_1$. Then, by substituting w_ℓ from (5.17) and exchanging the sums over ℓ and i we have,

$$\widehat{\boldsymbol{w}} = \sum_{i=1}^{k_1} \rho_i \left[\gamma_0 + \gamma_1(\lambda_i) + \sum_{\ell=3}^{m} \gamma_{\ell-1} \left[\left(\lambda_i\right)^{\ell-1} - \sum_{j=1}^{\ell-2} \xi_j^{\ell-1} \left(\lambda_i\right)^j\right]\right] (\lambda_i - 1)\boldsymbol{v}_i.$$

Now, by exchanging the sums over ℓ and j, it follows that,

$$\widehat{\boldsymbol{w}} = \sum_{i=1}^{k_1} \rho_i \Big[\gamma_0 + \sum_{j=1}^{m-2} \big[\gamma_j - \sum_{\ell=j+1}^{m-1} \gamma_\ell \xi_j^\ell\big](\lambda_i)^j + \gamma_{m-1}(\lambda_i)^{m-1}\Big](\lambda_i - 1)\boldsymbol{v}_i.$$

Since $\{\boldsymbol{v}_i\}_1^{k_1}$ are eigenvectors associated with distinct eigenvalues ($\lambda_i \neq 1$), they are linearly independent. Thus, noting that $\rho_i \neq 0$ for all $i = 1, \cdots, k_1$, then $\widehat{w} = 0$ implies that,

$$\gamma_0 + \sum_{j=1}^{m-2} \big[\gamma_j - \sum_{\ell=j+1}^{m-1} \gamma_\ell \xi_j^\ell\big](\lambda_i)^j + \gamma_{m-1}(\lambda_i)^{m-1} = 0,$$

for all $i = 1, \ldots, k_1$. By rewriting the last two sets of equations in matrix form, we get

$$L_{k_1}^m(A)(I - \Xi)\boldsymbol{\gamma} = 0, \tag{5.18}$$

where

$$L_{k_1}^m(S) \coloneqq \begin{pmatrix} 1 & \lambda_1 & \cdots & (\lambda_1)^{m-1} \\ 1 & \lambda_2 & \cdots & (\lambda_2)^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_k & \cdots & (\lambda_k)^{m-1} \end{pmatrix}, \quad \Xi \coloneqq \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \xi_1^2 & \xi_1^3 & \cdots & \xi_1^{m-1} \\ 0 & 0 & 0 & \xi_2^3 & \cdots & \xi_2^{m-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \xi_{m-2}^{m-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, \quad \boldsymbol{\gamma} \coloneqq \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \\ \vdots \\ \\ \gamma_{m-1} \end{pmatrix}. \tag{5.18a}$$

Note that $I - \Xi$ is invertible by construction, and $L_{k_1}^m(S)$ has a specific structure that hints at its invertibility. In fact, $L_\ell^\ell(S)$ is the Vandermonde matrix formed by ℓ eigenvalues of S which would be invertible if and only if $\lambda_1, \cdots, \lambda_\ell$ are distinct. More generally, $L_\ell^r(S)$, where $r \leq \ell$, has full column rank if $\{\lambda_1, \cdots, \lambda_\ell\}$ consists of ℓ distinct eigenvalues. Hence for $m \leq k_1$, $L_{k_1}^m(S)$ has full column rank and (5.18) implies that $\boldsymbol{\gamma} = 0$ and thus $\{\boldsymbol{w}_1, \dots, \boldsymbol{w}_m\}$ is a set of linearly independent vectors. This observation completes the proof as $m \leq k_1$ was chosen arbitrarily. □

Remark 5.1. Note that if (F, S) is observable and θ_0 excites all modes of S, then it is trivial to show that all modes of S can be recovered from the informative data.

5.4.2 Convergence Properties of the DG-RAN Algorithm

For a rapidly regularizable system with (unknown) uncertainty dynamics of the form (5.12), Theorem 5.6 provides an optimization formulation for obtaining the sub-optimal data-driven feedback $z(t, \mathcal{D}_t)$ at each time $t = n_t \delta t$.

Theorem 5.6. *Denote $\bar{Q}(t)$ and $\bar{Y}(t) = \mathbf{K}(t)\bar{Q}(t)$ as the solution variables to Problem 2.2, for $t \in [t_0, t_f]$. Further, assume the uncertain dynamical system of the form (5.3) is "rapidly regularizable" as defined in Definition 5.1. Then at time $t = n_t \delta t$, for the variable estimate $\hat{\theta}(t) = (\hat{S}_{n_t})^{t/\delta t} \hat{\theta}^\circ_{n_t}$ derived from (5.14), there exists a data-driven feedback controller of the*

form $z(\tau, \mathcal{D}_t) = \tilde{\mathbf{K}}(\tau)\hat{\theta}(t)$*, a matrix-valued function of time* $\widehat{M}(\tau)$ *and scalars* $\lambda(\tau)$*,* $\nu(\tau), \beta(\tau)$ *that for all time* $\tau \in [t, t_f]$ *solves the following optimization problem:*

$$\min_{\tilde{\mathbf{K}}(\cdot), \tilde{M}(\cdot), \nu(\cdot), \beta(\cdot), \lambda(\cdot)} \quad \sigma(\cdot) \tag{5.19}$$

subject to

$$0 < \beta, \;\; 0 < \nu, \;\; 0 < \lambda, \;\; \tilde{M} \in \mathcal{M}_{\phi,\Psi}, \tag{5.19a}$$

$$\begin{pmatrix} \bar{Q}A^\intercal + A\bar{Q} + T_2 & \gamma E & \nu\tilde{\gamma}E & T_3 \\ \gamma E^\intercal & -\beta I & 0 & 0 \;\; 0 \\ \nu\tilde{\gamma}E^\intercal & 0 & -\nu I & 0 \;\; 0 \\ * & * & * & T_4 \end{pmatrix} \preceq 0, \tag{5.19b}$$

$$z(\tau, \mathcal{D}_t) + \xi(\tau) \in \mathcal{U}_{\mathcal{F}}, \tag{5.19c}$$

where

$$T_2 \coloneqq \mathbf{Y}^\intercal B^\intercal + B\mathbf{Y} - \dot{\bar{Q}} + \lambda\bar{Q} + \beta\bar{Q}C_{cl}^\intercal C_{cl}\bar{Q}, \;\; T_3 \coloneqq \begin{bmatrix} B\tilde{\mathbf{K}}\hat{\theta} + (\delta t)^{-1}F\ln(\hat{S}_{n_t})\hat{\theta} & 0 \end{bmatrix},$$

$$\text{and } T_4 \coloneqq \begin{pmatrix} -\lambda\sigma & \hat{\theta}^\intercal\tilde{\mathbf{K}}^\intercal G \\ G\tilde{\mathbf{K}}\hat{\theta} & -\nu I \end{pmatrix}.$$

Proof. Considering Lemma 5.5, at time $t = n_t\delta t$ we can rewrite the estimation for $\hat{S}_{n_t}$ as

$$\hat{S}n_t = S\mathcal{Z}_{n_t-2}\mathcal{Z}_{n_t-2}^\dagger = \sum_{i=1}^{k_1} \left[\Sigma(\Sigma - I)L_{k_1}^{n_t-1}(I - \Xi_{n_t})\right]_i \boldsymbol{v}_i,$$

where $L_{k_1}^{n_t-1}$ is defined in (5.18a) and we define Ξ_{n_t} by removing the first column and row of Ξ in (5.18a). Note that generally, if $\{\lambda_1, \cdots, \lambda_{k_1}\}$, the eigenvalues of S corresponding to the excited modes $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{k_1} \in \mathcal{R}(F^\intercal)$, consists of r_d distinct eigenvalues (where $r_d \leq k_1$), then $L_{k_1}^{r_d}$ has full column rank. Hence, we can recover all modes of $\Pi_{\mathcal{R}(F^\intercal)}S$ when $n_t = r_d + 1$, and for $n_t < r_d + 1$ we have

$$\hat{\theta}(t) = \Pi_{\mathcal{R}\left(L_{k_1}^{n_t-1}\right)}\theta(t) \tag{5.20}$$

Since the system is "rapidly regularizable" in terms of Theorem 5.3, we can conclude that for any estimate of $\hat{\theta}(t)$ (5.20), there exists a controller of the form $z(\tau, \mathcal{D}_t) = \tilde{\mathbf{K}}(\tau)\hat{\theta}(t)$ and a

Lyapunov function $V_{\hat{\theta}}(\tau) = \eta(\tau, \hat{\theta}(t))^\intercal \bar{Q}^{-1}(\tau)\eta(\tau, \hat{\theta}(t))$ that satisfies the following,

$$\dot{V}_{\hat{\theta}}(\tau) \leq 0, \quad \text{for all } \sigma(\tau) < V_{\hat{\theta}}(\tau), \quad \begin{pmatrix} \widehat{q}(\tau) \\ \widehat{p}(\tau) \end{pmatrix}^\intercal \widehat{M}(\tau) \begin{pmatrix} \widehat{q}(\tau) \\ \widehat{p}(\tau) \end{pmatrix} \geq 0, \tag{5.21}$$

where $\widehat{q}_i(t) \coloneqq G_i \tilde{K}(t)\hat{\theta}(t)$, and $\widehat{p}_i(t) \coloneqq \tilde{\phi}_i(\tilde{q}_i(t)) = \phi_i(\tilde{q}_i(t) + q_i(t)) - \phi_i(q_i(t))$. Expanding this condition and following the same procedure as in the proof of Theorem 5.3 yields the final result.[2] □

We conclude this section by providing the condition on δt such that the DG-RAN algorithm guarantees the invariance of $\mathcal{E}_{\bar{Q}}$.

Theorem 2. Consider a "rapidly regularizable" nonlinear dynamic system. If the length of the measurement intervals satisfies $\delta t \leq \ell_\theta$, where,

$$\ell_\theta = \frac{\Pi_{\mathcal{R}\left(L_{k_1}^{n_t-1}\right)}\theta(t) - \Pi_{\mathcal{R}\left(L_{k_1}^{n_t-2}\right)}\theta(t)}{\Pi_{\mathcal{R}\left(L_{k_1}^{n_t-1}\right)^\perp}\dot{\theta}(t)}, \tag{5.22}$$

then the DG-RAN algorithm guarantees the invariance of the set described by $\bar{Q}(t)$; in other words, for all $t \in [t_0, t_f]$, $\eta(t) \in \mathcal{E}_{\bar{Q}_t}$.

Proof. Denote $\bar{Q}(t)$ and $\mathbf{K}(t) = \bar{Q}^{-1}(t)\bar{Y}(t)$ as the solution variables to Problem 2.2, for $t \in [t_0, t_f]$. Consider the Lyapunov function of the form

$$V_c(t) = V(t) + \frac{1}{2}\widetilde{\theta}(t)^\intercal \Gamma^{-1}\widetilde{\theta}(t), \tag{5.23}$$

where $V(t) = \eta(t)^\intercal \bar{Q}(t)^{-1}\eta(t)$ is the Lyapunov function for system (5.3) with input $\boldsymbol{u}(t) - \bar{\boldsymbol{u}}(t) = \mathbf{K}(t)\eta(t) + \tilde{\mathbf{K}}(t)\hat{\theta}(t)$, and $\tilde{\theta}(t)$ is defined in (5.2b). We need to prove there exist $\epsilon(t) \in \mathbb{R}$ such that $\dot{V}_c(t) \leq 0$ for any $\eta \in \mathcal{E}_{\bar{Q}(t)}$, where $\epsilon(t) \leq V_c(t)$. We have

$$\begin{aligned} \dot{V}_c &= \frac{\partial V}{\partial t} + \frac{\partial V}{\partial \eta}\dot{\eta} + \frac{\partial V}{\partial \hat{\theta}}\frac{\partial \widetilde{\theta}}{\partial t} + \widetilde{\theta}^\intercal \Gamma^{-1}\frac{\partial \widetilde{\theta}}{\partial t} \\ &= \underbrace{\dot{\eta}^\intercal \bar{Q}^{-1}\eta + \eta^\intercal \bar{Q}^{-1}\dot{\eta} - \eta^\intercal\left(\bar{Q}^{-1}\dot{\bar{Q}}\bar{Q}^{-1}\right)\eta}_{\dot{V}_{\hat{\theta}}(t)} + \left(2\eta^\intercal \bar{Q}^{-1}F + \frac{\partial V}{\partial \hat{\theta}} + \widetilde{\theta}^\intercal \Gamma^{-1}\right)\frac{\partial \widetilde{\theta}}{\partial t}, \end{aligned} \tag{5.24}$$

[2] This optimization problem contains Differential Matrix Inequalities. Refer to [115] for details of a numeric solution for a similar problem.

where $\dot{\eta}(t) = (A(t) + B(t)\mathbf{K}(t))\,\eta(t) + g\,(\boldsymbol{x}(t), \boldsymbol{u}(t)) + B(t)\tilde{\mathbf{K}}(t)\dot{\hat{\theta}}(t) + F\dot{\theta}(t)$.

From Theorem 5.6, for all time $\tau \in [t, t_f]$, there exists a data-driven feedback controller of the form $z(\tau, \mathcal{D}_t) = \tilde{\mathbf{K}}(\tau)\hat{\theta}(\tau)$ such that $\dot{V}_{\hat{\theta}}(\tau) \leq 0$ for all $\sigma(\tau) < V_{\hat{\theta}}(\tau)$. Furthermore, for $t \in [n_t\delta t, (n_t + 1)\delta t)$, we have

$$\frac{\partial V(t)}{\partial \hat{\theta}} = \frac{\partial}{\partial \hat{\theta}} \left(\int_{n_t\delta t}^{t} V(\tau) d\tau \right) = \int_{n_t\delta t}^{t} \eta(\tau)^\top \bar{Q}(\tau)^{-1} \left[B(\tau)\tilde{\mathbf{K}}(\tau) + \left(\frac{\partial g}{\partial u} \right) \tilde{\mathbf{K}}(\tau) \right] d\tau, \quad (5.25a)$$

and

$$\frac{\partial \widetilde{\theta}(t)}{\partial t} = \dot{\theta}(t) - \dot{\hat{\theta}}(t) = \dot{\theta}(t) - \Pi_{\mathcal{R}\left(L_{k_1}^{n_t-1}\right)} \dot{\theta}(t) - \frac{1}{\delta t} \left(\Pi_{\mathcal{R}\left(L_{k_1}^{n_t-1}\right)} \theta(t) - \Pi_{\mathcal{R}\left(L_{k_1}^{n_t-2}\right)} \theta(t) \right), \quad (5.25b)$$

where $\dot{\theta}(t) - \Pi_{\mathcal{R}\left(L_{k_1}^{n_t-1}\right)} \dot{\theta}(t) = \Pi_{\mathcal{R}\left(L_{k_1}^{n_t-1}\right)^\perp} \dot{\theta}(t)$. For $t \in [0, t_f]$, let $\epsilon(t) = \sigma(\tau)$ where $\tau = t \in [t, t_f]$. By substituting (5.25a) and (5.25b) in (5.24), the proof is now complete. □

5.5 Numerical Example

This section provides a numerical example of the DG-RAN algorithm applied to a 6-DOF powered descent guidance in the presence of time-varying atmospheric effects with unknown dynamics, such as those encountered in planetary landing (see Figure 5.2). This problem has nonlinear and highly unstable dynamics, state and control constraints, and large state and control dimensions that collectively make the problem challenging. The authors in [115] investigate the performance of the funnel synthesis approach for the 6-DOF powered descent guidance in the absence of uncertainty. In the scenarios studied in [115] $\gamma-$iteration method was introduced to solve Problem 2.2.

The control input for the system under consideration involves a thrust vector generated by a single main engine and a reaction control system (RCS). The maneuver being performed is assumed to occur close to the landing site and over a relatively short duration, allowing for the approximation of gravity as a constant vector.

Additionally, planetary rotation is neglected, and atmospheric effects are modeled as an unknown LTI system described by (5.12). We assume the vehicle's inertia matrix, center of mass, and center of pressure remain constant throughout the maneuver.

We establish a surface-fixed landing frame denoted as $\mathcal{F}_{\mathcal{I}}$ to describe the vehicle's orientation. This frame has its origin at the intended landing site and is defined by the orthonormal vectors $\{x_{\mathcal{I}}, y_{\mathcal{I}}, z_{\mathcal{I}}\}$. The vectors $x_{\mathcal{I}}$, $y_{\mathcal{I}}$, and $z_{\mathcal{I}}$ represent the downrange, crossrange, and local up directions, respectively. Similarly, a body frame denoted as $\mathcal{F}_{\mathcal{B}}$ is defined with its origin at the vehicle's center of mass. The body frame is constructed using the orthonormal vectors $\{x_{\mathcal{B}}, y_{\mathcal{B}}, z_{\mathcal{B}}\}$, with $z_{\mathcal{B}}$ chosen to align with the vehicle's vertical axis.

Due to the requirement of additive difference variables in (5.2), to represent the difference between the nominal state trajectory $\bar{\boldsymbol{x}}$ and the actual state trajectory $\boldsymbol{x}$, a 3-2-1 Euler angle sequence is employed to parameterize the orientation of $\mathcal{F}_{\mathcal{B}}$ with respect to $\mathcal{F}_{\mathcal{I}}$.

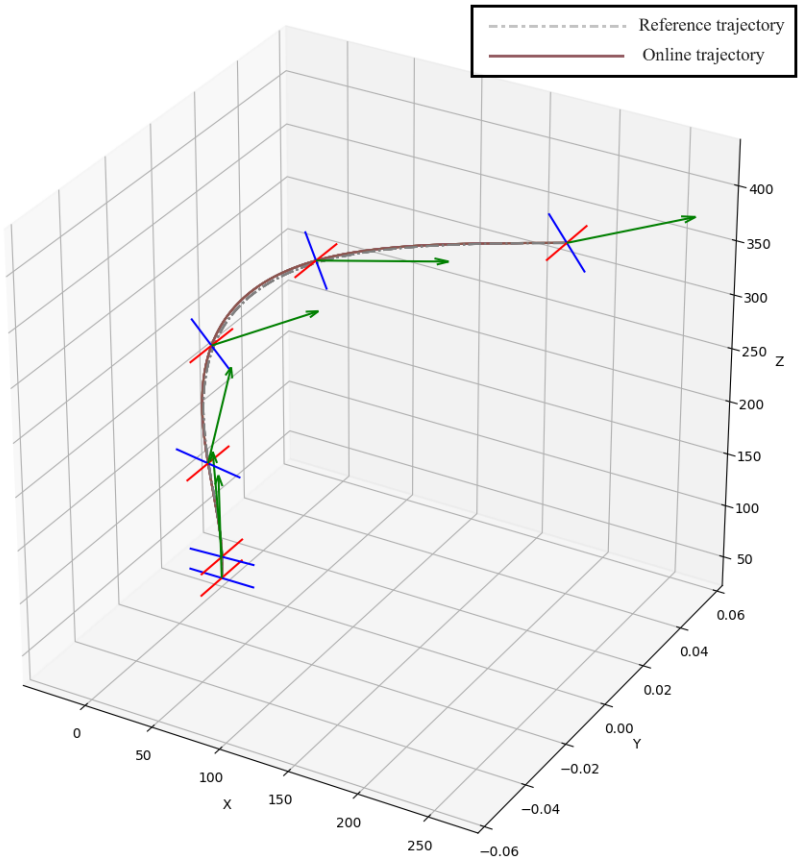


Figure 5.2: Successful landing of the six-degree-of-freedom (6-DOF) vehicle, as discussed in Figure 5.1, utilizing DG-RAN algorithm. This figure demonstrates the adaptive controller's enhanced capability to manage environmental disturbances effectively. DG-RAN ensures precise landing outcomes under unknown time-varying conditions by integrating real-time data into the control strategy.

The state, control, and nonlinear dynamics for this problem are

$$\boldsymbol{x} = \begin{bmatrix} m & r_{\mathcal{I}} & v_{\mathcal{I}} & \Theta & \omega_{\mathcal{B}} \end{bmatrix}^{\intercal}, \quad \boldsymbol{u} = \begin{bmatrix} F_{\mathcal{B}} & \tau_{\mathcal{B}} \end{bmatrix}^{\intercal}$$
$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} -\alpha_{\dot{m}} \left\| F_{\mathcal{B}} \right\|_2 & v_{\mathcal{I}} & \frac{1}{m} \left(C_{\mathcal{IB}}(\Theta) F_{\mathcal{B}} \right) + g_{\mathcal{I}} & T(\Theta)\omega_{\mathcal{B}} & J^{-1} \left(\tau_{\mathcal{B}} + r_{u,\mathcal{B}}^{\times} F_{\mathcal{B}} - \omega_{\mathcal{B}}^{\times} J \omega_{\mathcal{B}} \right) \end{bmatrix}^{\intercal}, \tag{5.26}$$

where $r_{\mathcal{I}} \in \mathbb{R}^3$ is the inertial position vector, $v_{\mathcal{I}} \in \mathbb{R}^3$ is the inertial velocity vector, $\Theta = (\varphi, \theta, \psi) \in \mathbb{R}^3$ represents the Euler angles, and $\omega_{\mathcal{B}} \in \mathbb{R}^3$ is the angular velocity in the body frame. The controlled inputs are the engine thrust vector $F_{\mathcal{B}}$ and an RCS torque vector $\tau_{\mathcal{B}}$. Note that $n_x = 13$ and $n_u = 6$ for this problem.

The true dynamics of the unknown disturbance is given by $\dot{\theta}(t) = \tilde{S}\theta(t)$ where,

$$\tilde{S} = \operatorname{diag}\{0.001,\ -0.12,\ 0.03,\ 0.05,\ 0.07,\ 0.2,\ 0.0005\}, \quad \theta_0 = 10^{-2} \begin{bmatrix} 0.2 & 0.1 & 0.1 & 3 & 1 & 2 & 3 \end{bmatrix}^{\intercal}$$
$$F = \begin{pmatrix} 0_{2\times2} & I_2 & 0_{1\times2} & 0_{2\times2} & 0_{4\times2} & 0_{3\times2} \\ 0_{2\times2} & 0_{2\times2} & 0_{1\times2} & I_2 & 0_{4\times2} & 0_{3\times2} \\ 0_{2\times3} & 0_{2\times3} & 0_{1\times3} & 0_{2\times3} & 0_{4\times3} & I_3 \end{pmatrix}^{\intercal}, \tag{5.27}$$

and F is the known basis function capturing the effects of the disturbance on the system dynamics. The disturbances in the system primarily affect the linear velocity and acceleration along the x-axis and y-axis, as well as the angular velocity in all three directions. These disturbances introduce variations and uncertainties in the system's motion, making the control problem more challenging. By addressing the impact of these disturbances, our proposed approach aims to regulate the system states and mitigate their effects, ultimately improving the overall performance and stability of the system.

The nominal trajectory was determined using the PTR algorithm, as described in [116], with specific boundary conditions and a final time of $t_f = 29.7$ s. The initial conditions at t_0 were set as: $\bar{m} = 3250$ kg, $\bar{r}_{\mathcal{I}} = (250, 0, 433)$ m, $\bar{v}_{\mathcal{I}} = (-35.7, 0, -11.8)$ m/s, $\Theta = (0, 59.8, 0)$ degrees, and $\omega_{\mathcal{B}} = 0_{3\times1}$. The final conditions at t_f were set as: $\bar{m} = 3130.3$ kg, $\bar{r}_{\mathcal{I}} = (0, 0, 30)$ m, $\bar{v}_{\mathcal{I}} = (0, 0, -1)$ m/s, $\Theta = 0_{3\times1}$, and $\omega_{\mathcal{B}} = 0_{3\times1}$. The constraint set for the state space $\mathcal{X}$ is

Table 5.1: The 6-DOF powered descent case study parameters.

Parameter	Value	Parameter	Value
J	$\mathbf{diag}\{1.36, 1.36, 1.9150\} \times 10^4\,\mathrm{kg\,m^2}$	$\alpha_{\dot{m}}$	4.5324×10^{-4} s/m
$r_{F,\mathcal{B}}$	$(0, 0, -0.25)$m	$g_{\mathcal{I}}$	$(0, 0, -1.62)\mathrm{m/s^2}$
$F_{\min}$	5400 N	$F_{\max}$	24750 N
κ_{tol}	0.5	α	0.1 s
$\delta_{\max}$	25 deg	$T_{\max}$	150Nm
N_s	100 n_M		

then taken to be,

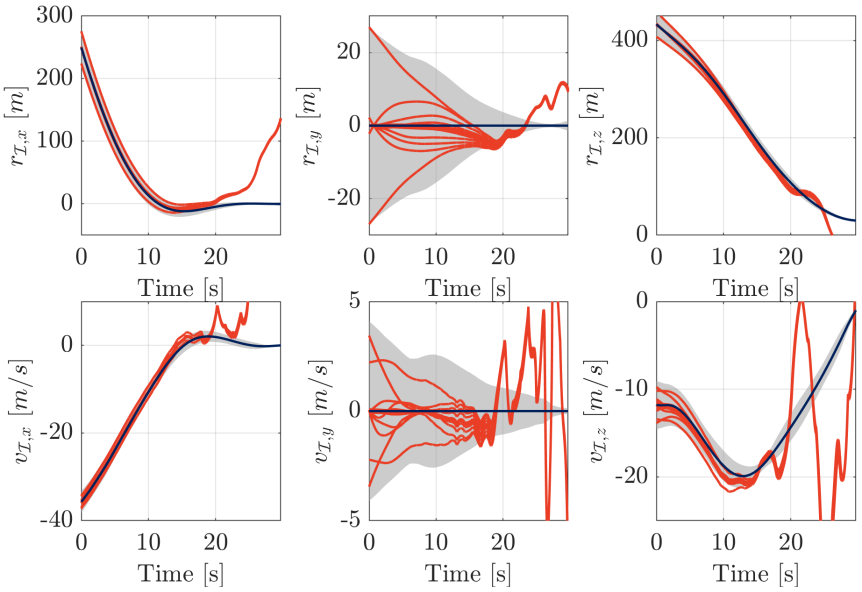
$$\mathcal{X} = \{x \mid x_{lb} \leq x \leq x_{ub}, \delta x_{lb} \leq x - \bar{x} \leq \delta x_{ub}, \|x\|_2 \leq \infty\}\ , \quad \mathcal{X}_f = \mathcal{E}_{Q_{\max,f}}\ ; \tag{5.28}$$

$\mathcal{X}$ enforces an absolute bound on the state vector and a bound on the deviation from the nominal trajectory, where,

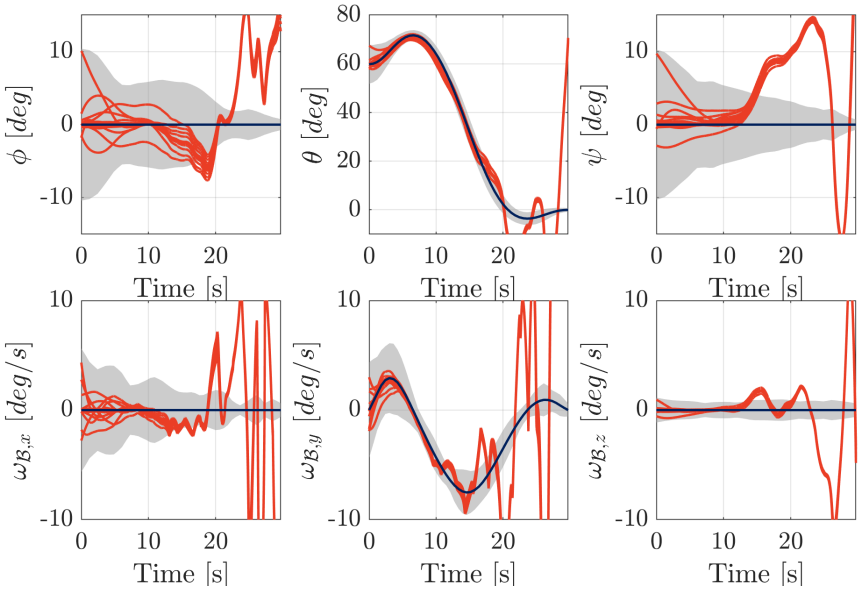
$$\begin{aligned}
x_{lb} &= -(-2100, 150, 150, 0, 40, 40, 30, \pi, \pi/2, \pi, 0.5, 0.5, 0.5)\\
x_{ub} &= +(3737.7, 350, 300, 500, 30, 30, 5, \pi, \pi/2, \pi, 0.5, 0.5, 0.5)\\
\delta x_{lb} &= -(\infty, 100, 100, 100, \infty, \infty, \infty, 2/9\pi, 2/9\pi, 2/9\pi, 2/9\pi, 2/9\pi, 2/9\pi),\\
\delta x_{ub} &= +(\infty, 100, 100, 100, \infty, \infty, \infty, 2/9\pi, 2/9\pi, 2/9\pi, 2/9\pi, 2/9\pi, 2/9\pi).
\end{aligned}$$

The terminal constraints specified by $\mathcal{X}_\mathcal{F}$ guarantee that every trajectory remains within a maximum deviation of 0.5 m in position, 0.25 m/s in velocity, and 3 degrees in attitude and 3 degrees/s in angular rate, from the nominal path at the final time. The additional information for this problem can be found in Table 5.1, which is loosely based on a lander similar to the Apollo class. The matrices A and B represent the partial derivatives of f along the nominal trajectory. The parameters H, G, and E are derived from a set of six ($n_p = 6$) nonlinear channels for δp and one ($n_{\tilde{p}} = 1$) nonlinear channel for $\delta\tilde{p}$[3].

[3]Details of the parameter matrices used in this section are adopted from [115]. Matrices H, G, and E are



(a) position and linear velocity in state space $\mathcal{E}_Q$



(b) rotation angles and angular velocity in state space $\mathcal{E}_Q$

Figure 5.3: The quadratic funnel computed by the γ-iteration for the 6-DOF power descent problem in the presence of unmodeled time-varying disturbances. The initial condition of each test case was randomly sampled from the funnel entry, and the system uses only the offline robust controller introduced in Problem 2.2.

Figure 5.3 shows the trajectories of the system in (5.3) generated using only the offline robust controller. The ellipsoid $\mathcal{E}_{\bar{Q}}$ is projected onto each state dimension (mass is omitted) and depicted as the shaded grey area. The red trajectories correspond to test cases for which an initial condition was randomly (uniformly) selected from the ellipsoid $\mathcal{E}_{\bar{Q}(t_0)}$, and the nominal control and correction law ($\xi(t) = \mathbf{K}(t)\eta(t)$) were used to integrate the equations of motion numerically.

$$u(t) = \bar{u}(t) + K(t)(x(t) - \bar{x}(t)) \tag{5.29a}$$

$$x(t) = x\left(t_0\right) + \int_{t_0}^{t} \Big(f(x(\tau), u(\tau)) + F\tilde{S}\theta(\tau) \Big) \, \mathrm{d}\tau \tag{5.29b}$$

In the presence of unmodeled disturbance, the offline controller fails to keep the states inside the quadratic state funnel, which results in undesirable system behavior.

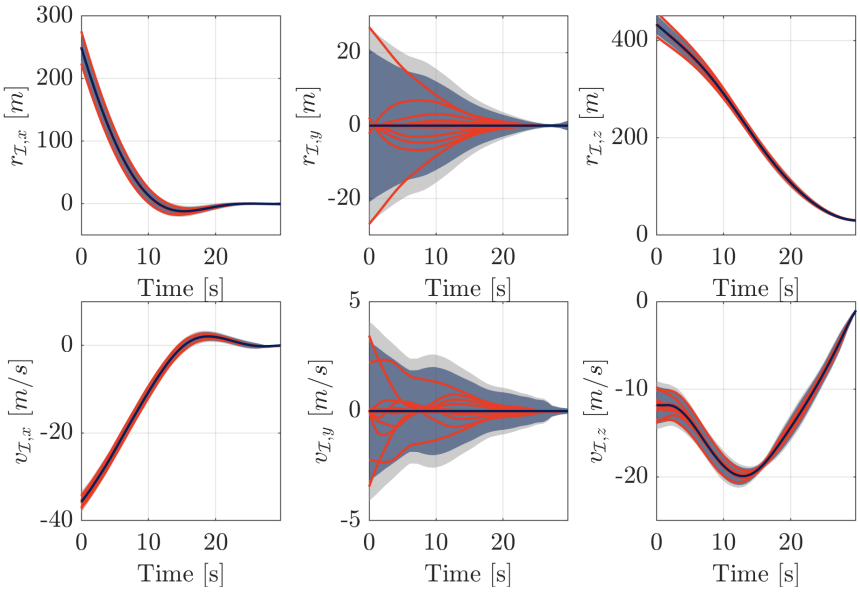
On the contrary, Figure 5.4 showcases the performance of the DG-RAN algorithm. The ellipsoid $\mathcal{E}_{\bar{Q}}$ is projected onto each state dimension and depicted as the shaded grey area, where the shaded blue area demonstrates the projection of ellipsoid $\mathcal{E}_{\bar{Q}}$ onto each state dimension, starting at time $t = n_t \delta t$ where $n_t = 10$ and $\delta t = 0.5$ s. Note that DG-RAN is an online algorithm where the proposed method was implemented uniquely for each trajectory to estimate the model of the disturbance dynamic and update $z(t, \mathcal{D}_t) = \tilde{\mathbf{K}}(t)\hat{\theta}(t)$ at each measurable time $t = n_t \delta t$.

$$u(t) = \bar{u}(t) + K(t)(x(t) - \bar{x}(t)) + \tilde{\mathbf{K}}(t)\hat{\theta}(t). \tag{5.30}$$

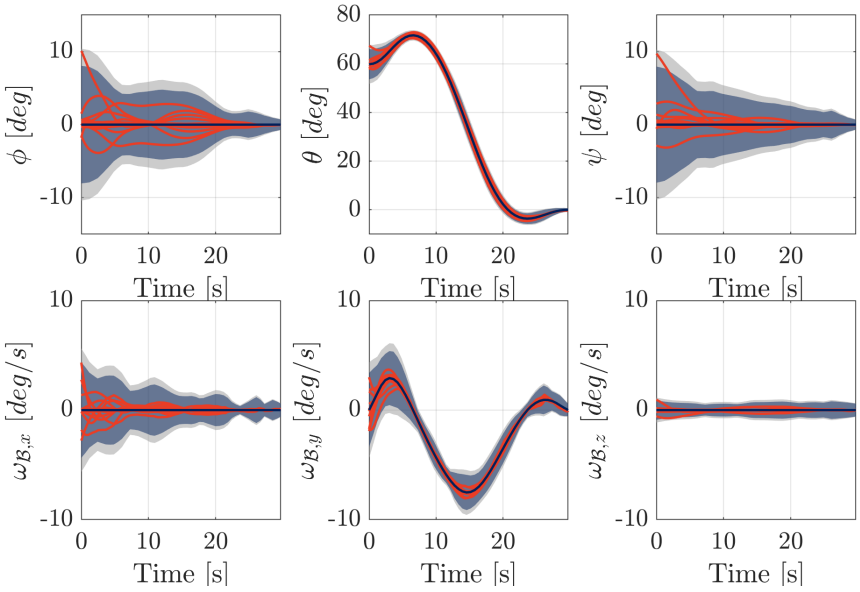
The proposed method can identify all modes of disturbance dynamic affecting the system at time $t = 5.5$ s with a data set $\mathcal{D}_t$ of size $n_t = 11$.

One important observation is how the DG-RAN algorithm identifies the disturbance modes in the order of their severity. Modes of the disturbance dynamic causing the trajectory to deviate from the nominal are quickly identified, and $z(t, \mathcal{D}_t)$ is updated to reflect such deviations. In particular, the data is informative enough to identify unknown parameters through a least squares solution denoted by $\hat{S}_{n_t}$. Therefore, one stopping criterion–which

equivalent to C, D, and E defined in section IV of [115].



(a) position and linear velocity in state space $\mathcal{E}_Q$



(b) rotation angles and angular velocity in state space $\mathcal{E}_Q$

Figure 5.4: State trajectories generated using the online DG-RAN algorithm for the 6-DOF power descent problem in the presence of unmodeled time-varying disturbances. The initial condition for this test case was randomly sampled from the funnel entry, and the proposed method was implemented online for each test case.

is also used here–is the point where the estimate of parameters $\hat{S}_{n_t}$ has converged. Then, $z(t, \mathcal{D}_t)$ is implemented without updates from $t =$ onward. In contrast to using only the offline controller (5.29a), Figure 5.4 showcases the performance of the updated controller (5.30) in the presence of environmental disturbances. We have a more accurate estimate of the disturbance dynamic at each iteration using the data generated safely by DG-RAN in the loop.

These results are quite promising–powered descent is a challenging problem with nonlinear dynamics, nonlinear and nonconvex constraints, and relatively large state and control dimensions. What these results demonstrate is that in the presence of unmodeled disturbances, we were able to generate feasible trajectories (both dynamically and with respect to the constraints considered) for any initial condition in a set that stretches more than 35 m in every position direction, 2 m/s in each velocity direction, 16deg in each Euler angle, and 2deg/s in each angular velocity direction, all by using online data-driven updates.

Part II

PERCEPTION-AWARE TRAJECTORY PLANNING AND CONTROL

Incorporating insight from rich, perceptual sensing modalities such as cameras have the potential to convey more information than simple, single-output sensor devices. These new methods utilize high-resolution images to train a neural network for object detection, state estimation, and localization. The use of such tools can be challenging in harsh lighting conditions and against a highly textured background [117, 118]. Given enough data, deep learning algorithms can learn efficient models that map high-dimensional data to an array of outputs. However, modeling the uncertainties in the output is environment-dependent and a challenging task. This becomes particularly crucial when the trained neural network is adopted as a perception module in the control loop for autonomous navigation—which requires specific safety guarantees.

In general, deep learning does not consider uncertainty representation in regression settings, and deep learning classification models often give normalized score vectors, which do not necessarily capture model uncertainty. To this end, in the following chapter, we seek to combine the best of learning with robust/optimal control for autonomous navigation in safety-critical scenarios.

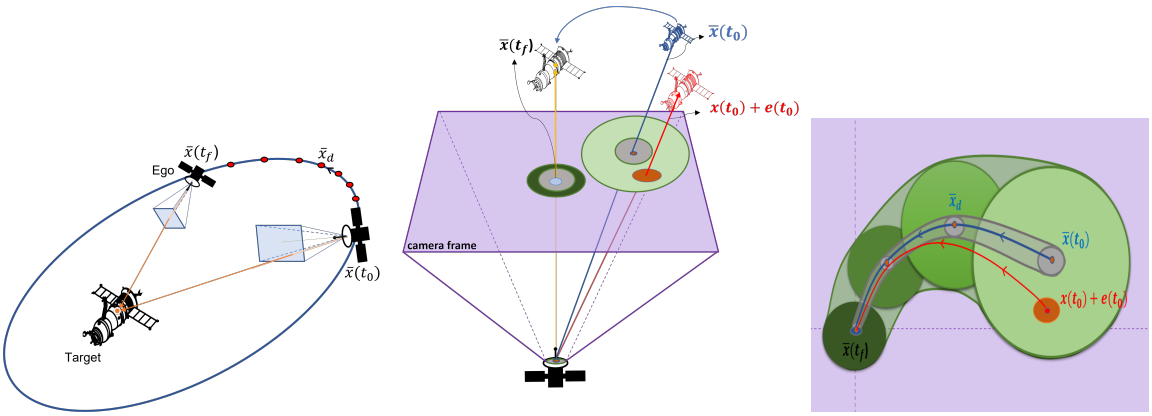
Chapter 6

Robust Controller Synthesis for Perception-Based Control

Consider the following scenario: A spacecraft (ego) has to navigate to a specific position and orientation with respect to another spacecraft (target) while keeping the target inside its camera field of view (see Figure 6.1a). This task requires an estimation of the relative state of the target spacecraft in the ego's body frame using a sequence of imagery observations. In this chapter, we lay the foundations to address this key Guidance Navigation and Control (GNC) challenge that must be solved for the aforementioned missions to succeed.

Our approach is to first, expand the analysis in [119] to design a virtual sensor by learning both a perception map (i.e., a map from observations to a linear function of the target state) and a "state-dependent" bound on its estimation errors. In contrast to [119], for space applications, the distance to the target object and the sun's position dramatically affect the information that can be extracted from an image. Hence, we model the error of the composed perception process as a function of the target and ego's relative states. Further, we utilize the implicit trajectory optimization methods to define a set of feasible trajectories in the state-and-control spaces. Then, based on a nominal trajectory and the UUB concept [120], we extend the invariant funnel synthesis technique [7] to a setup accounting for the state-dependent output measurement noise—where we can incorporate the error model and its upper bounds correspondingly. A schematic illustration of this idea is also depicted in Figure 6.1. Finally, we arrive at the composite robust control synthesis, which generates control signals directly from imagery data and can handle uncertainties in the (nonlinear) model and the (noisy) observations.

The result of this chapter has been mainly adapted from [3, 6]



(a) Orbital trajectory (b) Target in camera frame (c) Invariant Funnel

Figure 6.1: The orbital trajectory of the ego around the target in its Hill's frame. Figure 6.1b illustrates the nominal trajectory starting at $\bar{x}(t_0)$ (in blue), and terminal state $\bar{\boldsymbol{x}}(t_f)$ (in black). The actual initial state $x(t_0)$ (shown in red) is contained in the invariant funnel at $t = 0$ (the light green ellipse). Figure 6.1c shows the same nominal (blue) and actual (red) trajectories contained in the invariant funnel (green ellipses).

6.1 Review of Past Work

The traditional approach to autonomous navigation and control of a fleet of vehicles involves using a combination of sensors, such as differential GPS, laser/reflector meteorology, and inter-vehicle communication, for relative pose estimation and a control framework for precise formation keeping and reconfiguration. However, many of these active communications cannot be used when dealing with uncooperative agents.

In recent years, there has been a growing interest in using Machine Learning for object tracking and pose estimation from camera images. High-quality cameras have become readily available for a range of applications from autonomous driving to autonomous multi-spacecraft operations. For space applications, vision-based estimation can also augment differential GPS, which is usually not reliably available beyond Low Earth Orbit (LEO) and requires constant inter-spacecraft communication. Electro-Optical (EO) camera sensors are compact, space tested, and can provide a rich set of information, making them an ideal sensor for multi-spacecraft smallsat missions. A large body of research has demonstrated the superior performance of deep learning, specifically Convolutional and Recurrent Neural Networks (CNN and RNN). Drawing inspiration from robotics research and applications [121–124], the ESA Pose Estimation Challenge in 2019 helped in the advancement of a number of these vision-based pose estimation algorithms[1]. In fact, vision+CNN-based relative pose estimation is the only approach that works for proximity operations around passive/uncooperative targets in space applications.

Superior computational capabilities onboard the vehicle enable the extraction of relevant pose estimates from the images of other active agents in the environment. While older methods have relied on simple computer vision algorithms along with carefully placed fiducial markers, new powerful onboard computers allow the use of a new set of vision-based relative estimation tools based on artificial intelligence and Machine Learning.

While these methods are powerful, they rely on training a neural network with a database

[1]https://kelvins.esa.int/satellite-pose-estimation-challenge/

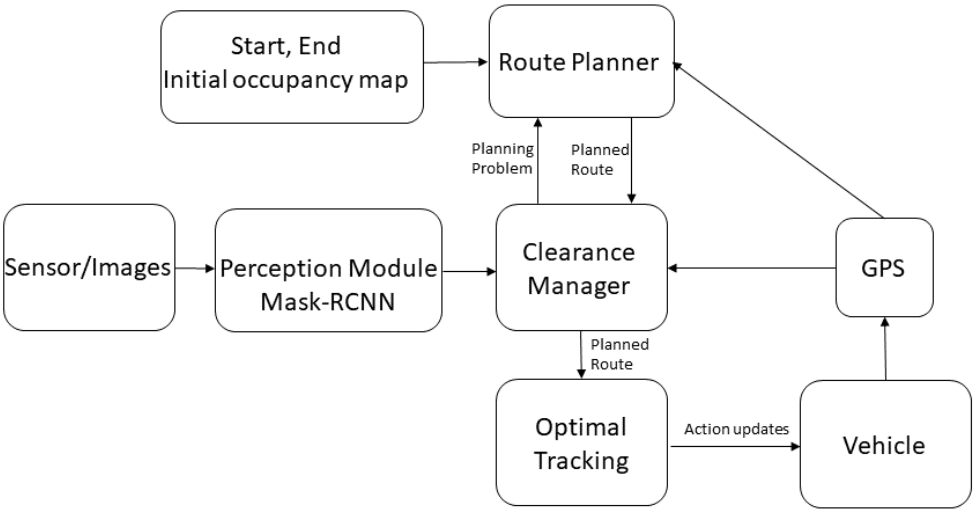


Figure 6.2: Autonomous Executive Functional Architecture

of synthetic images that cover all expected views from the onboard camera's perspective during the mission's lifetime. As such, these CNN/RNN act as black boxes that receive images and output an estimate of the pose. For instance, consider a scenario where a vehicle is to autonomously navigate in an airport environment. The autonomous executive functional architecture is depicted in Figure 6.2. The perception module consists of "Mask-RCNN" neural network architecture [125], along with a depth map. Mask-RCNN is a mask-based segmentation neural network with a Resnet backbone. Given an input image, it predicts a mask bounding box and a class for all predicted objects in the input image (see Figure 6.3).

Given the initial occupancy map, RRT* method [126] is used to generate a path from the start point to the goal location. When the vehicle reaches the vicinity of an unknown obstacle, it starts processing images taken from the scene. The perception module determines the position of the unknown obstacle. The map is then updated using this information, and a new path is generated. The vehicle then follows the new path to the goal location [2].

While these methods are powerful, they rely on training a neural network with a database

[2]The code for these simulations can be found at `https://github.com/NiyoushaRahimi/UW-MLP`

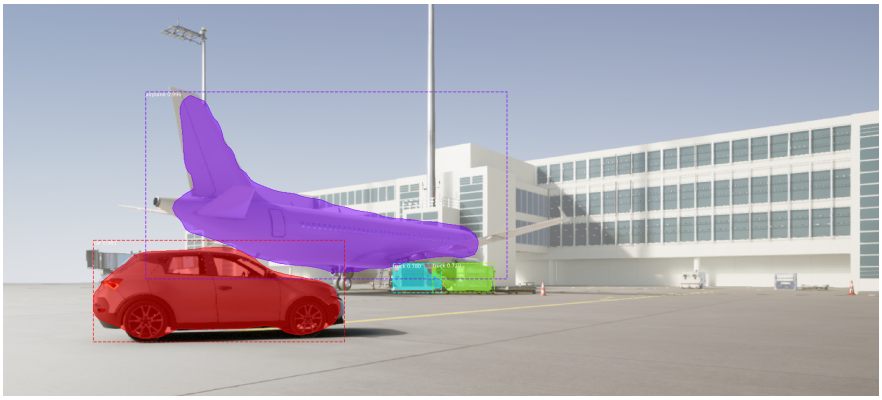


Figure 6.3: Semantic segmentation of an image using Mask-RCNN.

of synthetic images that cover all expected views from the perspective of the onboard camera during the lifetime of the mission. For space applications, this black box approach, along with reliance on synthetic training data that might be different from what might be encountered in orbit, can lead to unreliable estimates that are far off from reality and do not follow a usually assumed Gaussian error model. Most research on mitigating this issue has been focused on "explainable AI", and essentially trying to characterize this general estimation using neural networks and potential reasons for failure to produce a quality estimate. This focus on separating estimation (relative navigation) and control can lead to unforeseen stability issues that can arise from the failure of the ML (CNN/RNN) based relative pose estimation (both due to its black box nature as well as the insufficient ground-based training data); which can excite the otherwise stable formation flying controller such that the whole closed loop becomes unstable.

6.2 Nonlinear System Model with State-Dependent Perception Error

We consider a problem setting in which the main uncertainty comes from the image sensor, generating observations that capture information about the state of a target relative to an ego spacecraft. Then, a specific (trained) "perception map" is utilized to recover the actual relative state of the system from the imagery data on the fly. Particularly, we define the

(relative) state $x(t) \in \mathbb{R}^{n_x}$ (between ego and target spacecraft) as

$$x(t) := x_{\text{ego}}(t) - x_{\text{target}}(t); \tag{6.1}$$

which follows a nonlinear dynamics

$$\dot{x}(t) = f(x(t), u(t)), \quad t \in [t_0, t_f], \tag{6.2}$$

$$z(t) = \mathcal{Q}(x(t)), \tag{6.3}$$

where $u(t) \in \mathbb{R}^{n_u}$ denotes the control inputs, $z(t) \in \mathbb{R}^m$ represents sensory data as a function of the relative state, and $[t_0, t_f]$ is a prescribed time interval. The observed data $z(t) = \mathcal{Q}(x_{\text{ego}}(t) - x_{\text{target}}(t))$ is then determined by the unknown generative model $\mathcal{Q} : \mathbb{R}^{n_x} \to \mathbb{R}^m$, which is nonlinear and potentially quite high dimensional. The task of the observation module is to detect the target and determine its relative position with respect to the ego spacecraft. Here, the observations $z(t)$ are the captured images and the map $\mathcal{Q} : \mathbb{R}^{n_x} \to \mathbb{R}^m$ generates these images as a function of the relative states, as in (6.3). For now, suppose there exists a *perception map* $p : \mathbb{R}^m \to \mathbb{R}^l$ that imperfectly predicts the relative state $x(t)$, that is

$$p(z(t)) = Cx(t) + e_p(x(t)), \tag{6.4}$$

where $e_p(x(t))$ denotes the perception error which is *dependent* on the relative state. The details of this perception map and its error characterization is further discussed in the succeeding section.

The main control synthesis problem is composed of two main steps. First, a (nonlinear) controller, denoted by $\bar{u}(t)$, is designed which results in a nominal controlled state trajectory. Then, a correction input signal, denoted by $\xi(t)$, is synthesized that deals with both dynamics and measurement uncertainties rooted in the modeling and composed perception process, respectively. More precisely, we let $\{\bar{x}(t), \bar{u}(t)\}_{t=t_0}^{t_f}$ be the nominal trajectory that satisfies the dynamics (6.2) for some initial condition x_0, and define

$$\eta(t) := x(t) - \bar{x}(t) \quad \text{and} \quad \xi(t) := u(t) - \bar{u}(t). \tag{6.5}$$

By assuming that f is at least once differentiable, we can rewrite (6.2) in terms of the variables in (6.5) by using a first-order Taylor series expansion around the nominal trajectory as follows:

$$\dot{\eta}(t) = A(t)\eta(t) + B(t)\xi(t) + g(x(t), u(t)), \tag{6.6}$$

where $A(t)$ and $B(t)$ are the partial derivatives of f evaluated along the nominal trajectory, and the $g(x(t), u(t)$ represents the higher order (nonlinear) terms. The dynamical system in (6.6) can then be equivalently expressed using structured nonlinearities as follows

$$\begin{aligned}
\dot{\eta}(t) &= A(t)\eta(t) + B(t)\xi(t) + \sum_{i=1}^{N_w} E_i w_i(t) \\
h_i(t) &= F_i\eta(t) + G_i\xi(t), && i = 1, \dots, N_w \\
w_i(t) &= \phi_i\left(h_i(t)\right), && i = 1, \dots, N_w
\end{aligned}$$

where the pairs $(h_i, w_i) \in \mathbb{R}^{n_{h,i}} \times \mathbb{R}^{n_{w,i}}$ capture the higher order terms through the nonlinear functions ϕ_i. The details of this transformation are laid out in [12], and the constant matrices $F_i \in \mathbb{R}^{n_h \times n_x}$, $G_i \in \mathbb{R}^{n_h \times n_u}$ and $E_i \in \mathbb{R}^{n_x \times n_w}$ serve as nonlinear inputs/outputs channel selectors.

Finally, based on the nominal trajectory $\{\bar{x}(t), \bar{u}(t)\}_{t=t_0}^{t_f}$, we can define the output of the system using the composed perception process in (6.4) as

$$y(t) \coloneqq p(z(t)) - C\bar{x}(t) = C\eta(t) + e(\eta(t)), \tag{6.7}$$

where $e(\eta(t))$ denotes the state-dependent measurement noise—to be characterized later in Section 6.3. Hence, we arrive at the following (nonlinear) system dynamics and (state-dependent) noisy observations:

$$\begin{aligned}
\dot{\eta}(t) &= A(t)\eta(t) + B(t)\xi(t) + \sum_{i=1}^{N_w} E_i w_i(t) \\
h_i(t) &= F_i\eta(t) + G_i\xi(t), && i = 1, \dots, N_w \\
w_i(t) &= \phi_i\left(h_i(t)\right), && i = 1, \dots, N_w, \\
y(t) &= C\eta(t) + e(\eta(t)).
\end{aligned} \tag{6.8}$$

Depending on the the perception module, it is reasonable to assume that $C = I$ in most of applications which only simplifies the presentation of our proposed methodology, however, we provide clarifications whenever it is necessary.

6.3 Modelling Sensor Uncertainties

In order to obtain a perception map p that can recover the relative state "x" from the imagery data "z", we suppose access to initial training dataset $\mathcal{S}_0 = \{(x_d, z_d)\}_{d=1}^{N_0}$ which are collected *densely enough* from the nominal trajectory $\{\bar{x}(t), \bar{u}(t)\}_{t=t_0}^{t_f}$. This dataset is used to learn a perception map via any wide variety of traditional supervised methods [6]. To accurately model the uncertainties in the output measurements (6.7)—which will be subsequently used in the robust control synthesis—we estimate the safe regions around the nominal trajectory in terms of parameters learned from a second (testing) dataset. Next, we define the regions of the state space within which the sensing is reliable. To this end, we consider the following local assumptions:

Assumption 6.1. *The mapping $p \circ \mathcal{Q} - C$ is locally s-slope bounded with a radius of r around each training data points, i.e., for each $(x_d, z_d) \in \mathcal{S}_0$, there exists a bounded real number $s = s(x_d)$ such that*

$$\|p \circ \mathcal{Q}(x) - Cx - (p \circ \mathcal{Q}(x_d) - Cx_d)\| \leq s(x_d) \|(x - x_d)\|, \quad \forall x \in B_r(x_d). \tag{6.9}$$

Assumption 6.2. *The training error at each point of the training dataset is bounded, i.e., for each $(x_d, z_d) \in \mathcal{S}_0$, there exists a bounded real number $\gamma = \gamma(x_d)$ such that*

$$\|p(z_d) - Cx_d\| \leq \gamma(x_d).$$

Note that the modeling assumptions above are completely local which enables us to utilize the perception module to its best extent and avoid using the *uniform* worst-case-scenario

modeling. Next, the regions of the state space within which the sensing is reliable can then be described using a safe set which approximates sub-level sets of the norm of the state-dependent error function $e(x) = p(q(x)) - Cx$ around the nominal trajectory. More precisely, given $r > 0$, we define

$$\mathcal{X}_r(\bar{x}(t)) \coloneqq \Big\{ x \in B_r\left(\overline{x}(t)\right) \Big| \left\| p\left(\widetilde{z}_d\right) - C\widetilde{x}_d \right\| + s(t) \left\| \overline{x}(t) - \widetilde{x}_d(t) \right\| \leq l(t), \\ \widetilde{x}_d(t) \coloneqq \arg\min_{x_d \in \mathcal{S}_0} \|x - x_d\| \Big\}, \tag{6.10}$$

where $\widetilde{z}_d(t) \coloneqq \mathcal{Q}(\widetilde{x}_d(t))$, $\gamma(t) \coloneqq \gamma(\widetilde{x}_d(t))$, and

$$s(t) \coloneqq s\left(\widetilde{x}_d(t)\right), \quad l(t) \coloneqq rs(t) + \gamma(t). \tag{6.11}$$

Note that we overload the notation to define $s(t)$, which is the maximum slope bound of the mapping $p \circ \mathcal{Q} - C$ at all $x \in B_r(\widetilde{x}_d)$. Since the nominal trajectory $\{\bar{x}(t), \bar{u}(t)\}_{t=t_0}^{t_f}$ is fixed and the training data $\mathcal{S}_0$ is collected from this trajectory, we are able to define the safe set $\mathcal{X}_r(t) = \mathcal{X}_r(\bar{x}(t))$ as a function of time around the nominal trajectory. Further, note that parameters $s(x_d)$ can be learned using a second dataset (see [6] for more details). Finally, we can precisely define how densely the nominal trajectory should be sampled for the training dataset as follows:

Definition 6.1. We say that the training dataset $\mathcal{S}_0 = \{(x_d, z_d)\}_{d=1}^{N_0}$ is *sampled densely enough* from nominal trajectory $\{\bar{x}(t), \bar{u}(t)\}_{t=t_0}^{t_f}$ if there exists a positive radius $r > 0$ such that for any $x \in B_r\left(\overline{x}(t)\right)$ and any $t \in [t_0, t_f]$, we have $\|\bar{x}(t) - \widetilde{x}_d(t)\| \leq r$ where $\arg\min_{x_d \in \mathcal{S}_0} \|x - x_d\|$.

The error characteristics considered in our approach is different from that of [119] as each training data point x_d is associated with different parameters $\gamma(x_d)$ and $s(x_d)$, and we parametrized the safe set around the nominal trajectory as a function of time. The rationale behind this modeling is that here, the error of the composed perception process depends on the relative state between ego and target spacecrafts, certainly because, as we get farther away from the target, the amount of information that can be extracted from the images

decreases significantly. Moreover, environmental factors such as target reflection or the sun's position can also dramatically affect the perception error. Hence, the training error increases accordingly and in order to model this phenomenon, we assume that the training error is a function of the training points as described in Assumptions 6.1 and 6.2.

Now, we are well-equipped to characterize the perception error—altering the output of the system as modeled in (6.7)—in the following lemma.

Lemma 6.3. *Suppose the training dataset is sampled densely enough from the nominal trajectory (per Definition 6.1) with a radius r and Assumptions 6.1 and 6.2 hold with the same radius. Then, for any $x(t) \in \mathcal{X}_r(\bar{x}(t))$, the perception error is locally bounded with*

$$\|e(\eta(t))\| \leq s(t)\|\eta(t)\| + l(t).$$

Proof. For any $x(t) \in \mathcal{X}_r(\bar{x}(t))$ (corresponding to a sensory data $z(t) = \mathcal{Q}(x(t))$) such that $x(t) \in B_r(\overline{x}(t))$, consider $(\widetilde{x}_d(t), \widetilde{z}_d(t)) \in \mathcal{S}_0$ and observe that

$$\begin{aligned}\|p(z) - Cx\| &= \|p(z) - Cx - (p(\widetilde{z}_d) - C\widetilde{x}_d) + p(\widetilde{z}_d) - C\widetilde{x}_d\| \\ &\leq s(x_d)\,\|(x - \widetilde{x}_d)\| + \|p(\widetilde{z}_d) - C\widetilde{x}_d\|,\end{aligned}$$

where the last inequality follows from the local slope bound assumption. Recall that $\eta(t) := x(t) - \bar{x}(t)$ and the nominal trajectory $\{\bar{x}(t), \bar{u}(t)\}_{t=t_0}^{t_f}$ is known. Since the training dataset $\mathcal{S}_0 = \{(x_d, z_d)\}_{d=1}^{N_0}$ is sampled densely enough from the nominal trajectory, for any $(x(t), z(t))$ with $x(t) \in \mathcal{X}_r(\bar{x}(t))$ we can upper bound the perception error in (6.8) as follows

$$\begin{aligned}\|e(\eta(t))\| &\leq s(t)\|\,(x(t) - \bar{x}(t) + \bar{x}(t) - \widetilde{x}_d(t))\,\| + \|p(\mathcal{Q}(\widetilde{x}_d)) - C\widetilde{x}_d(t)\| \\ &\leq s(t)\|\eta(t)\| + s(t)\|\,(\bar{x}(t) - \widetilde{x}_d(t))\,\| + \|p(\mathcal{Q}(\widetilde{x}_d)) - C\widetilde{x}_d(t)\| \leq s(t)\|\eta(t)\| + l(t),\end{aligned}$$

where the last inequality follows by definition. □

Finally, note that as we have a more densely sampled nominal trajectory (i.e., as $r > 0$ is smaller,) the affine upper-model of the noise will have smaller shift $l(t)$, however, the slope-bound and training errors are inherent to the learning problem at hand and does not necessarily improve by just sampling more densely.

6.4 Robust Controller Synthesis

In this section we explicate how the problem reformulation and error modeling presented so far, can be utilized for synthesizing a robust controller that overcomes the uncertainties inherent to both the dynamics model and the perception process. First, as a result of Theorem 6.3, we can consider the uncertain nonlinear system in (6.8) with the following perception error

$$\|e(\eta(t))\| \leq s(t)\|\eta(t)\| + l(t). \tag{6.12}$$

We consider the class of controllers that, at each iteration t, can synthesis a correction signal $\xi(t) = \mathbf{K}(t)y(t)$ for some matrix-valued function of time $\mathbf{K}(t) \in \mathbb{R}^{n_u \times n_y}$. The closed-loop system representation obtains the following form:

$$\dot{\eta}(t) = \left(A(t) + B(t)\mathbf{K}(t)\right)\eta(t) + B(t)\mathbf{K}(t)e\left(\eta(t)\right) + \sum_{i=1}^{N_w} E_i w_i(t) \tag{6.13a}$$

$$h_i(t) = \left(F_i + G_i\mathbf{K}(t)\right)\eta(t) + G_i\mathbf{K}(t)e\left(\eta(t)\right), \qquad i = 1, \cdots, N_w \tag{6.13b}$$

$$w_i(t) = \phi_i\left(h_i(t)\right), \qquad i = 1, \cdots, N_w, \tag{6.13c}$$

$$\|e(\eta(t))\| \leq s(t)\|\eta(t)\| + l(t), \tag{6.13d}$$

As discussed in the previous section, the perception error is bounded as (6.12) in a vicinity of the nominal trajectory. We use this fact to model the term $\mathbf{K}(t)e(\eta(t))$ in the dynamics of $\eta(t)$ as a structured nonlinearity. In fact, we aim to design a feedback synthesis procedure that can handle any possible perception error $e(t)$ that satisfies the upper bound model $\|e(t)\| \leq s\|\eta(t)\| + l(t)$, including the realized error which inherently may depend on $\eta(t)$. The details are provided in the following subsection.

6.4.1 Modeling the Perception Error in Feedback

Herein, we propose a specific modeling of the the perception error that eventually enables us to derive at an LMI formulation for synthesizing the feedback gain K. It is worth noting that the direct approach for considering the error in measurements results in nonlinear inequalities

with no known LMI equivalent. In particular, we consider the error $e(t)$ as a nonlinear channel and model the effects of any such error (including the perception error $e(\eta(t))$)—that satisfies the upper-bound model $\|e(t)\| \leq s\|\eta(t)\| + l(t)$—as

$$w_e(t) \coloneqq \widetilde{\phi}\,(e(t)) = \mathbf{K}(t)\, e(\eta(t)), \quad \text{for some} \quad \widetilde{\phi} : \mathbb{R}^{n_y} \to \mathbb{R}^{n_u}. \tag{6.14}$$

By considering $e(t)$ as a nonlinear input channel, we then need to update the nonlinear input channel selectors in $h(t)$

$$h_i(t) = (F_i + G_i\mathbf{K}(t))\,\eta(t) + \widetilde{G}_i e(t), \qquad i = 1, \cdots, N_w \tag{6.15}$$

where the constant matrix $\widetilde{G}_i \in \mathbb{R}^{n_x \times n_y}$ serves as nonlinear input channel selector for $e(t)$. We then stack each of the ϕ_i to construct the function $\phi : \mathbb{R}^{n_h} \to \mathbb{R}^{n_w}$, and rewrite (6.13) together with (6.14) and (6.15) as

$$\dot{\eta}(t) = A_{cl}(t)\eta(t) + B(t)w_e(t) + Ew(t), \tag{6.16a}$$

$$h(t) = F_{cl}(t)\eta(t) + \widetilde{G}e(t), \tag{6.16b}$$

$$w(t) = \phi(h(t)), \tag{6.16c}$$

$$w_e(t) = \widetilde{\phi}\,(e(t)) \tag{6.16d}$$

where $E = \begin{bmatrix} E_1 & \cdots & E_{N_w} \end{bmatrix}$, and

$$w = \begin{pmatrix} w_1 \\ \vdots \\ w_{N_w} \end{pmatrix}, \quad h = \begin{pmatrix} h_1 \\ \vdots \\ h_{N_w} \end{pmatrix}, \quad F = \begin{pmatrix} F_1 \\ \vdots \\ F_{N_w} \end{pmatrix}, \quad G = \begin{pmatrix} G_1 \\ \vdots \\ G_{N_w} \end{pmatrix}, \quad \widetilde{G} = \begin{pmatrix} \widetilde{G}_1 \\ \vdots \\ \widetilde{G}_{N_w} \end{pmatrix},$$

such that $w \in \mathbb{R}^{n_w}$ and $h \in \mathbb{R}^{n_h}$, $F \in \mathbb{R}^{n_h \times n_x}$, $G \in \mathbb{R}^{n_h \times n_u}$, $\widetilde{G} \in \mathbb{R}^{n_h \times n_y}$, and $E \in \mathbb{R}^{n_x \times n_w}$ with $n_w = \sum_{i=1}^{N_w} n_{w,i}$ and $n_h = \sum_{i=1}^{N_w} n_{h,i}$, and $A_{\text{cl}}(t) \coloneqq A(t) + B(t)\mathbf{K}(t)$ and $F_{\text{cl}}(t) \coloneqq F + G\mathbf{K}(t)$.

The assumption that $\xi = Ky$ used to form the closed-loop system (6.16) thus results in the following implication:

$$\text{if} \quad y \in \mathcal{E}_{Q_y} \quad \Rightarrow \quad \xi \in \mathcal{E}_{KQ_yK^\intercal} \tag{6.17}$$

A fact that can be proven easily via Schur complements when K is full row-rank [12]. Before we proceed, we need to drive an upper bound for Q_y, given the information about the perception error.

Theorem 6.4. *If $\eta \in \mathcal{E}_Q$ and the perception error is bounded $\|e\| \leq s\|\eta\| + l$, then $y \in \mathcal{E}_{Q_y}$, where*

$$Q_y = \left(1 + s\left(1 + \rho\right)\right)^2 Q, \qquad Q_s := s^2 Q, \quad \rho := \sqrt{l\lambda_{\max}\left(Q_s^{-1}\right)} \tag{6.18}$$

and $\lambda_{\max}\left(Q_s^{-1}\right)$ is the largest eigenvalue of Q_s^{-1}.

Proof. Let $e = e_\eta + r$, where $\|e_\eta\| \leq s\|\eta\|$ and $\|r\| \leq l$. Define $Q_s := s^2 Q$, then by the Schur complement it follows that $e_\eta \in \mathcal{E}_{Q_s}$. We further claim that

$$e \in \mathcal{E}_{Q_e}, \quad \text{where} \quad Q_e := \left(1 + \rho\right)^2 Q_s. \tag{6.19}$$

The proof utilizes the fact that for any positive definite matrix P and any vectors x, y, we have $2x^\intercal P y \leq \frac{1}{a} x^\intercal P x + a y^\intercal P y$ for all $a > 0$. In particular, for any $\hat{\rho}$, we have

$$\begin{aligned} e^\intercal {Q_e}^{-1} e &= \left(e_\eta + r\right)^\intercal \left(\left(1 + \rho\right)^2 Q_s\right)^{-1} \left(e_\eta + r\right) \\ &\leq \left(1 + \rho\right)^{-2} \left(1 + \hat{\rho}\right) e_\eta^\intercal Q_s^{-1} e_\eta + \left(1 + \rho\right)^{-2} \left(1 + \frac{1}{\hat{\rho}}\right) r^\intercal Q_s^{-1} r \end{aligned}$$

Let $\hat{\rho} = \rho$ and note that since $\|r\| \leq l$, then $r^\intercal Q_s^{-1} r \leq l\lambda_{\max}\left(Q_s^{-1}\right)$. Hence we have $e^\intercal Q_e e \leq 1$ and the proof of the claim is complete. Similarly, for $y = \eta + e$ we have

$$\begin{aligned} y^\intercal {Q_y}^{-1} y &= \left(\eta + e\right)^\intercal Q_y \left(\eta + e\right) \\ &\leq \left(1 + s\left(1 + \rho\right)\right)^{-2} \left(1 + \hat{\rho}\right) \eta^\intercal Q^{-1} \eta + \left(1 + s\left(1 + \rho\right)\right)^{-2} \left(1 + \frac{1}{\hat{\rho}}\right) e^\intercal Q^{-1} e \end{aligned}$$

Let $\hat{\rho} = s\left(1 + \rho\right)$. The proof of the theorem follows from the over-approximation of Q_e, where $e^\intercal {Q_e}^{-1} e = e^\intercal \left(s(1+\rho)^2 Q\right)^{-1} e \leq 1$. □

6.4.2 Quadratic Funnel Synthesis in Presence of Measurement Noise

In this work, we have incorporated the funnel synthesis techniques introduced in Section 2.4.1 to provide one (or more) functions that implicitly define a set of trajectories in the state-and-control space. Suppose that $\mathcal{X}_{\mathcal{F}} \in \mathbb{R}^{n_x}$ and $\mathcal{U}_{\mathcal{F}} \in \mathbb{R}^{n_u}$ are the (possibly nonconvex) sets of feasible state and control vectors. Using these feasible sets, we formally define a quadratic invariant funnel for the output feedback problem.

Definition 6.2. (Quadratic Funnel). A quadratic funnel, $\mathcal{F}$, is a subset of the Cartesian product of the feasible state space $\mathcal{X}_{\mathcal{F}}$ and control space $\mathcal{U}_{\mathcal{F}}$ (so-called feasible state-and-control space) that is parameterized by time-varying positive definite matrices $Q \in \mathbb{S}^{n_x}_{++}$, $Q_y \in \mathbb{S}^{n_y}_{++}$ and a time-varying matrix $K \in \mathbb{R}^{n_u \times n_y}$. Specifically, we define

$$\mathcal{F} \coloneqq \mathcal{E}_Q \times \mathcal{E}_{KQ_yK^\intercal} \subseteq \mathcal{X}_{\mathcal{F}} \times \mathcal{U}_{\mathcal{F}}. \tag{6.20}$$

As the closed-loop system— described in (6.16)—is obtained from an output feedback controller with (state-dependent) measurement and process noise, one has to consider an appropriate stability criterion. Hence, we require that any proposed controller renders a UUB[3] closed-loop system. We utilized Theorem 4.18 in [120] to design a controller $\mathbf{K}$ such that the closed loop system (6.16) is UUB, in the presence of state-dependent noise and system nonlinearity, and starting anywhere in an invariant funnel. A sufficient condition to achieve this goal in view of the dynamics in (6.16) is detailed as follows:

$$\begin{aligned} &\alpha_1(\|\eta(t)\|) \leq V(\eta(t)) \leq \alpha_2(\|\eta(t)\|), \qquad \frac{\partial V}{\partial t} + \frac{\partial V}{\partial \eta}\dot{\eta}(t) \leq -V(\eta(t)), \qquad \forall\, t \in [t_0, t_f], \\ &\forall \eta \ni \|C_\mu \eta\| \geq \mu \geq 0, \quad \forall h \in \Omega, \quad \text{and} \quad \forall e \in \mathcal{E}_{Q_e} \ni \|e(t)\| \leq s(t)\|\eta(t)\| + l(t), \end{aligned} \tag{6.21}$$

where μ denotes the time-varying UUB parameter. Note that the quadratic funnel's invariance property means that $\eta(t) \in \mathcal{E}_Q$ for all times $t \in [t_0, t_f]$. This implies that the vector h, which

[3]See Definition 4.6 in [120] for more details on UUB concept.

is the input to the nonlinear terms in system (6.13), must satisfy the set inclusion

$$h \in \Omega, \quad \Omega = \mathcal{E}_{F_{cl} Q F_{cl}^{\intercal}} \oplus \mathcal{E}_{\widetilde{G} Q_e \widetilde{G}^{\intercal}} \tag{6.22}$$

where $\oplus$ is the Minkowski sum between two sets. In order to utilize the condition (6.21) for synthesizing a quadratic funnel, we need to express the nonlinear expressions "$w = \phi(h), \quad \forall h \in \Omega$" and "$w_e = \widetilde{\phi}(e), \quad \forall\, e \in \mathcal{E}_{Q_e}$"—appearing in (6.16)—in such a way that it is consistent with the quadratic form of the energy function $V(\eta(t))$. To this end, we use the idea of local multiplier matrices as discussed in Section 2.4.1.

Definition 6.3. Consider any nonlinear map $\phi : \mathbb{R}^{n_h} \to \mathbb{R}^{n_w}$ that sends $h \to \phi(h)$. We say a symmetric matrix $M \in \mathbb{S}^{(n_h+n_w)}$ is a *local multiplier matrix* for ϕ over the set Ω if

$$\mathcal{T}(h) := \begin{pmatrix} h \\ \phi(h) \end{pmatrix}^{\intercal} M \begin{pmatrix} h \\ \phi(h) \end{pmatrix} \geq 0, \quad \forall\, h \in \Omega. \tag{6.23}$$

Furthermore, we denote the set of local multiplier matrices for ϕ over the set Ω by

$$\mathcal{M}_{\phi,\Omega} \coloneqq \left\{ M \in \mathbb{S}^{(n_h+n_w)} \mid \mathcal{T}(h) \geq 0, \quad \forall\, h \in \Omega \right\}. \tag{6.24}$$

Similarly we define the set of local multiplier matrices for $\widetilde{\phi}(e)$, where M_e denotes a *local multiplier matrix* for $\widetilde{\phi}$ over the set Q_e:

$$\mathcal{T}_e(e) := \begin{pmatrix} e \\ \widetilde{\phi}(e) \end{pmatrix}^{\intercal} M_e \begin{pmatrix} e \\ \widetilde{\phi}(e) \end{pmatrix} \geq 0, \quad \forall\, e \in \mathcal{E}_{Q_e}. \tag{6.25}$$

Finally, the following theorem provides sufficient conditions that guarantee the invariance property of a quadratic funnel in the presence of both measurement and modeling uncertainties.

Theorem 6.5. *The output controller of the form $\xi(t) = \mathbf{K}(t)y(t)$ satisfies the feasibility conditions (6.21) if there exists $Q > 0$, $Y \in \mathbb{R}^{n_u \times n_x}$, and multiplier matrices $M \in \mathcal{M}_{\phi,\Omega}$, $M_e \in \mathcal{M}_{\widetilde{\phi},\mathcal{E}_{Q_e}}$ and scalars $\tau, \tau_e, \alpha, \lambda > 0$ such that for all $t \in [t_0, t_f]$,*

$$\begin{pmatrix} \mathbf{Q}A^{\intercal} + A\mathbf{Q} - \dot{\mathbf{Q}} + \mathbf{Q} + T_1 & E + \tau \mathbf{F}_{cl}^{\intercal} M_{12} & B & \tau \mathbf{F}_{cl}^{\intercal} M_{11} \widetilde{G} & 0 \\ E^{\intercal} + \tau M_{12}^{\intercal} \mathbf{F}_{cl} & \tau M_{22} & 0 & \tau M_{12}^{\intercal} \widetilde{G} & 0 \\ * & * & [& T_2 &] \end{pmatrix} \preceq 0 \tag{6.26}$$

where

$$
\begin{aligned}
T_1 &= \tau\ \mathbf{F}_{cl}^{\intercal} M_{11}\ \mathbf{F}_{cl} + \alpha \mathbf{Q} C_{\mu}^{\intercal} C_{\mu} \mathbf{Q} + \lambda s \mathbf{Q}\mathbf{Q}, \qquad \mathbf{F}_{cl} = F\mathbf{Q} + G\mathbf{Y} \\
T_2 &= \begin{pmatrix} \tau_e M_{e22} & \tau_e M_{e12}^{\intercal} & 0 \\ \tau_e M_{e12} & \tau \widetilde{G}^{\intercal} M_{11} \widetilde{G} + \tau_e M_{e11} - \lambda I & 0 \\ 0 & 0 & -\alpha\mu + \lambda R \end{pmatrix}
\end{aligned} \tag{6.27}
$$

Proof. Using local multiplier matrices and because $V(\eta(t))$ is not an explicit function of t, we can rewrite the condition in (6.21) as

$$
\begin{aligned}
&\dot{V}(\eta(t)) + V(\eta(t)) \leq 0, \quad \forall t \in [t_0, t_f], \quad \begin{pmatrix} h \\ w \end{pmatrix}^{\intercal} M \begin{pmatrix} h \\ w \end{pmatrix} \geq 0, \\
&\begin{pmatrix} e \\ w_e \end{pmatrix}^{\intercal} M_e \begin{pmatrix} e \\ w_e \end{pmatrix} \geq 0, \quad \|C_{\mu}\eta(t)\| \geq \mu \geq 0, \quad \|e(\eta(t))\| \leq s(t)\|\eta(t)\| + l(t),
\end{aligned} \tag{6.28}
$$

for some $M \in \mathcal{M}_{\phi,\Omega}$ and $M_e \in \mathcal{M}_{\widetilde{\phi},\mathcal{E}_{Q_e}}$. Expanding this condition in terms of the closed-loop system (6.16) leads to

$$
\begin{pmatrix} \eta \\ w \\ w_e \end{pmatrix}^{T} \begin{pmatrix} \mathcal{A}^T Q^{-1} + Q^{-1}\mathcal{A} - Q^{-1}\dot{Q}Q^{-1} + Q^{-1} & Q^{-1}E & Q^{-1}B \\ E^{\intercal}Q^{-1} & 0 & 0 \\ B^{\intercal}Q^{-1} & 0 & 0 \end{pmatrix} \begin{pmatrix} \eta \\ w \\ w_e \end{pmatrix} \preceq 0
$$

for all η, w and w_e that satisfy the following inequalities.

$$
\begin{pmatrix} \eta \\ e \\ w \end{pmatrix}^{\intercal} \begin{pmatrix} F_{cl}^{\intercal}\ M_{11}\ F_{cl} & F_{cl}^{\intercal}\ M_{11}\widetilde{G} & F_{cl}^{\intercal}\ M_{12} \\ \widetilde{G}^{\intercal}\ M_{11} F_{cl} & \widetilde{G}^{\intercal} M_{11} \widetilde{G} & \widetilde{G}^{\intercal} M_{12} \\ M_{12}^{\intercal}\ F_{cl} & M_{12}^{\intercal}\widetilde{G} & M_{22} \end{pmatrix} \begin{pmatrix} \eta \\ e \\ w \end{pmatrix} \succeq 0, \quad \begin{pmatrix} e \\ w_e \end{pmatrix}^{\intercal} M_e \begin{pmatrix} e \\ w_e \end{pmatrix} \succeq 0,
$$

$$
\begin{pmatrix} \eta \\ 1 \end{pmatrix}^{\intercal} \mathbf{diag}\Big(\begin{bmatrix} C_{\mu}^{\intercal} C_{\mu} & -\mu \end{bmatrix} \Big) \begin{pmatrix} \eta \\ 1 \end{pmatrix} \succeq 0, \quad \begin{pmatrix} \eta \\ e \\ 1 \end{pmatrix}^{\intercal} \mathbf{diag}\Big(\begin{bmatrix} sI & -I & R \end{bmatrix} \Big) \begin{pmatrix} \eta \\ e \\ 1 \end{pmatrix} \succeq 0
$$

Note that $\mathcal{A} = A + B\mathbf{K}$ and $F_{cl} = F + G\mathbf{K}$. By using the S-procedure, the last condition holds if and only if there exist scalars $\tau \geq 0, \alpha \geq 0$ and $\lambda \geq 0$ such that

$$\begin{pmatrix} \mathcal{A}^\intercal \mathbf{Q}^{-1} + \mathbf{Q}^{-1}\mathcal{A} - \mathbf{Q}^{-1}\dot{\mathbf{Q}}\mathbf{Q}^{-1} + \mathbf{Q}^{-1} + T_1 & \mathbf{Q}^{-1}E + \tau F_{cl}^\intercal M_{12} & \mathbf{Q}^{-1}B & \tau F_{cl}^\intercal M_{11}\widetilde{G} & 0 \\ E^\intercal Q^{-1} + \tau M_{12}^\intercal F_{cl} & \tau M_{22} & 0 & \tau M_{12}^\intercal \widetilde{G} & 0 \\ * & * & [& T_4 &] \end{pmatrix} \preceq 0 \tag{6.29}$$

where

$$T_1 = \tau\ F_{cl}^\intercal M_{11}\ F_{cl} + \alpha C_\mu^\intercal C_\mu + \lambda s I, \quad T_2 = \begin{pmatrix} \tau_e M_{e22} & \tau_e M_{e12}^\intercal & 0 \\ \tau_e M_{e12} & \tau \widetilde{G}^\intercal M_{11}\widetilde{G} + \tau_e M_{e11} - \lambda I & 0 \\ 0 & 0 & -\alpha\mu + \lambda R \end{pmatrix}$$

Pre- and post-multiplying (6.29) by $\mathbf{diag}\Big(\begin{bmatrix} Q & I & I & I \end{bmatrix} \Big)$, will complete the proof.

□

6.4.3 Feasibility of the Quadratic Funnel

We now discuss the conditions that are required for a quadratic funnel to be contained in a given feasible region. As it was discussed in [7], the goal is to find the "maximum quadratic funnel", denoted by $\mathcal{F}_{\max}$, that satisfies state and control constraints. Suppose that $\mathcal{X}_\mathcal{F} \subset \mathbb{R}^{n_x}$ and $\mathcal{U}_\mathcal{F} \subset \mathbb{R}^{n_u}$ are the (possibly nonconvex) feasible subsets of the state-and-control space. We assume that the nominal trajectory satisfies $\bar{x} \in \text{int}\ \mathcal{X}_\mathcal{F}$ and $\bar{u} \in \text{int}\ \mathcal{U}_\mathcal{F}$, or, in words, the nominal trajectory lies in the strict interior of the two corresponding feasible subsets. This assumption is necessary to avoid degenerate solutions during funnel synthesis procedure. Note that, this assumption does not preclude nominal trajectories that activate a constraint; it merely suggests that the constraint sets used for funnel synthesis are slightly more relaxed than those imposed during nominal trajectory synthesis, if necessary. This can be easily done by inflation/deflation of the feasibility subsets. Also, the set that determines the maximum size of the funnel at the final time $t = t_f$, is denoted by $\mathcal{X}_\mathcal{F}^f \subset \mathbb{R}^{n_x}$. Then, the maximum quadratic funnel satisfies

$$\mathcal{F}_{\max} = \mathcal{E}_{Q_{\max}} \times \mathcal{E}_{R_{\max}} \quad \text{and} \quad \mathcal{E}_{Q_{\max}} \subseteq \mathcal{X}_{\mathcal{F}}, \quad \mathcal{E}_{R_{\max}} \subseteq \mathcal{U}_{\mathcal{F}} \tag{6.30}$$

where $Q_{\max} \in \mathbb{S}^{n_x}_{++}$ and $R_{\max} \in \mathbb{S}^{n_u}_{++}$. The, inclusion in the terminal set $\mathcal{X}^f_{\mathcal{F}}$ can be guaranteed by ensuring that $\mathcal{E}_{Q_{\max}} \subseteq \mathcal{X}^f_{\mathcal{F}}$ when $t = t_f$.

To compute the matrix $Q_{\max}$ at any time $t < t_f$, we seek to find the best quasi-polytopic approximation of $\mathcal{X}_{\mathcal{F}}$ in the vicinity of the nominal trajectory. To this end, following the analysis presented in [7], we assume that the set $\mathcal{X}_{\mathcal{F}}$ can be expressed as

$$\mathcal{X}_{\mathcal{F}} = \{x \mid h_i(x) \leq 0, i = 1, \ldots, m_x, \|x\|_2 \leq x_{\max}\} \tag{6.31}$$

and the quasi-polytopic approximation of $\mathcal{X}_{\mathcal{F}}$ is

$$\mathcal{P}_{\bar{x}} = \{x \mid a_i^\intercal x \leq b_i,\ i = 1, \ldots, m_x\} \cap \{x \mid \|x\|_2 \leq x_{\max}\} \tag{6.32}$$

where the data $\{a_i, b_i\}_{i=1}^{m_x}$ are first-order approximations to the functions h_i computed using either direct linearization along the nominal trajectory, or by using a project-and-linearize technique if any h_i is a concave function [127]. Note that, if the state constraints h_i are all affine then $\mathcal{X}_{\mathcal{F}} = \mathcal{P}_{\bar{x}}$. The matrix $Q_{\max}$ is then computed as the maximum volume ellipsoid so that

$$\bar{x} \oplus \mathcal{E}_{Q_{\max}} \subset \mathcal{P}_{\bar{x}} \subseteq \mathcal{X}_{\mathcal{F}} \tag{6.33}$$

where $\oplus$ denotes the Minkowski sum of a vector and set, and so the term $\bar{x} \oplus \mathcal{E}_{Q_{\max}}$ is equivalent to the ellipsoid defined by $Q_{\max}$ centered at the vector $\bar{x}$. The assumption that $\bar{x}$ lies in the strict interior of $\mathcal{X}_{\mathcal{F}}$ ensures that $Q_{\max}$ describes a full n_x -dimensional ellipsoid.

Using techniques described in [8], we can write the condition (6.33) for any time $t \in [t_0, t_f]$ as the following optimization problem:

$$\begin{aligned} Q^{1/2}_{\max}(t) = \arg\max_{Z} \ & \log\det Z \\ \text{s.t.} \ & \|Z a_i(t)\|_2 + a_i^\intercal \bar{x}(t) \leq b_i(t), \quad i = 1, \ldots, m_x, \\ & 0 \prec Z \prec x_{\max} I_{n_x}\,. \end{aligned} \tag{6.34}$$

When $t = t_f$, a set of conditions can be added to constrain $\mathcal{E}_{Q_{max}} \subseteq \mathcal{X}_{\mathcal{F}}^f$. By assuming that the set $\mathcal{X}_{\mathcal{F}}^f$ can be described in the same way as that of $\mathcal{X}_{\mathcal{F}}$, this set of constraints is identical to those that are shown in (6.34), albeit calculated using the data corresponding to $\mathcal{X}_{\mathcal{F}}^f$. The matrices $R_{\max}$ can be computed analogously.[4]

Recall that, a quadratic funnel is defined by the (possibly time-varying) ellipsoids $\mathcal{E}_Q$ and $\mathcal{E}_{KQ_yk^\intercal}$. Further, note that we derived the convex feasible subsets (defined by $Q_{\max}$ and $R_{\max}$) for the state and control inputs. A sufficient condition to ensure that a quadratic funnel is fully contained within the original feasible domain defined by $\mathcal{X}_{\mathcal{F}}$ and $\mathcal{U}_{\mathcal{F}}$ is

$$\mathcal{E}_Q \subseteq \mathcal{E}_{Q_{\max}} \quad \text{and} \quad \mathcal{E}_{KQ_yK^\intercal} \subseteq \mathcal{E}_{R_{\max}}. \tag{6.35}$$

By using the definition of an ellipsoid (9), one can show that these conditions are equivalent to

$$Q \preceq Q_{\max} \quad \text{and} \quad KQ_yK^\intercal \preceq R_{\max} \tag{6.36}$$

The former is affine in the variable Q, whereas the latter is not jointly convex in the variables Q_y and K. By Theorem 6.4 and using a Schur complement, we can rewrite the second matrix inequality in (6.36) to be

$$(1 + s\,(1+\rho))^2 \; KQK^\intercal \preceq R_{\max} \quad \Leftrightarrow \quad \begin{pmatrix} (1 + s\,(1+\rho))^{-2}\, Q^{-1} & K^\intercal \\ K & R_{\max} \end{pmatrix} \succeq 0 \tag{6.37}$$

Pre- and post-multiplying by $\mathrm{diag}\{Q, I\}$ the feasibility condition for the control input becomes

$$\begin{pmatrix} (1 + s\,(1+\rho))^{-2}\, Q & Y^\intercal \\ Y & R_{\max} \end{pmatrix} \succeq 0 \tag{6.38}$$

Together, the constraints $Q \preceq Q_{\max}$ and (6.38) ensure that a quadratic funnel is contained in the feasible region defined by $\mathcal{X}_{\mathcal{F}}$ and $\mathcal{U}_{\mathcal{F}}$.

We are now in a position to lay out the performance guarantee of the vision-based controller synthesis.

---

[4]See [7] for more details.

Theorem 6.6. *Define the regions of the state space within which the sensing is reliable as described in (6.10). Then, if the feasible state space $\mathcal{X}_{\mathcal{F}}$ describing the quadratic funnel $\mathcal{F}$ lies within the safe set $\mathcal{X}_r(\bar{x}(t))$, then the perception error remains bounded and the closed loop trajectories defined by the maximum quadratic funnel $\mathcal{F}_{\max}$ lie within $\mathcal{X}_r(\bar{x}(t))$.*

Proof. If $\mathcal{X}_f \subseteq \mathcal{X}_r(\bar{x}(t))$, then by the invariance of the quadratic funnel $\mathcal{E}_Q \subseteq \mathcal{X}_r(\bar{x}(t))$. Moreover, by Theorem 6.3, for all $x(t) \in \mathcal{E}_Q$, the perception error remains bounded, i.e. $e(\eta(t)) \leq s(t)\|\eta(t)\| + l(t)$. □

6.5 The Quadratic Funnel Synthesis Problem

In this section, we combine the properties of the quadratic funnel and pose a single optimization problem. The quadratic funnel synthesis problem for the nonlinear system (6.8) with perception error is summarized in Problem 6.1.

Problem 6.1. (Quadratic Funnel Synthesis). Given a nominal trajectory $\{\bar{x}(t), \bar{u}(t)\}_{t=t_0}^{t_f}$ that satisfies the nonlinear dynamics (6.2), an appropriate definition of the system (6.8), find the matrix-valued functions of time $Q(t)$, $Y(t), K(t), M(t), M_e(t)$ and scalars $\tau(t), \tau_e(t), \alpha(t)$ and $\lambda(t)$ that solve the following optimization problem, for $t \in [t_0, t_f]$.

$$\max_{\substack{Q(\cdot),Y(\cdot),K(\cdot),M(\cdot),M_e(\cdot),\\ \tau(\cdot),\tau_e(\cdot),\alpha(\cdot),\lambda(\cdot)}} \log\det Q\left(t_0\right) \tag{6.39a}$$

$$\textbf{subject to} \quad 0 \preceq Q \preceq Q_{\max}, \quad M \in \mathcal{M}_{\phi,\Omega}, \quad M_e \in \mathcal{M}_{\widetilde{\phi},\mathcal{E}_{Q_e}} \tag{6.39b}$$

$$0 \leq \tau, \quad 0 \leq \tau_e, \quad 0 \leq \alpha, \quad 0 \leq \lambda, \tag{6.39c}$$

$$\begin{pmatrix} \mathbf{Q}A^\intercal + A\mathbf{Q} - \dot{\mathbf{Q}} + \mathbf{Q} + T_1 & E + \tau\mathbf{F}_{cl}^\intercal M_{12} & B & \tau\mathbf{F}_{cl}^\intercal M_{11}\widetilde{G} & 0 \\ E^\intercal + \tau M_{12}^\intercal \mathbf{F}_{cl} & \tau M_{22} & 0 & \tau M_{12}^\intercal \widetilde{G} & 0 \\ * & * & [& T_2 &] \end{pmatrix} \preceq 0, \tag{6.39d}$$

$$\begin{pmatrix} \left(1 + s\left(1+\rho\right)\right)^{-2} Q & Y^\intercal \\ Y & R_{\max} \end{pmatrix} \succeq 0 \tag{6.39e}$$

The matrices T_1 and T_2 are given by (6.27), and $\mathbf{F}_{cl} = F\mathbf{Q} + G\mathbf{Y}$.

A quadratic funnel obtained by solving Problem 6.1 can be used to generate a feasible trajectory in the following manner. Let $\bar{x}\,(t_0)$ be the initial condition of the reference state trajectory. For any initial condition $x\,(t_0)$ such that $x\,(t_0) - \bar{x}\,(t_0) \in \mathcal{E}_{Q(t_0)}$, the state and control vectors are defined as

$$u(t) = \bar{u}(t) + \mathbf{K}(t)\, y(t), \tag{6.40a}$$

$$x(t) = x\,(t_0) + \int_{t_0}^{t} f(x(\tau), u(\tau))\mathrm{d}\tau, \tag{6.40b}$$

and will satisfy $\eta = x - \bar{x} \in \mathcal{E}_Q$ and $\xi = u - \bar{u} = Ky \in \mathcal{E}_{KQ_yK^\intercal}$, for all $t \in [t_0, t_f]$. The feasibility with respect to the constraint sets $\mathcal{X}_\mathcal{F}$ and $\mathcal{U}_\mathcal{F}$ then follows. Notice that dynamic feasibility follows from (6.40) by construction, and the control u is guaranteed to be stabilizing for the original nonlinear dynamics.

6.5.1 The Solution Strategy

Problem 6.1 is a nonlinear, nonconvex optimization problem with matrix variables. The main difficulty in solving Problem 6.1 lies in the fact that it is not possible (to the best of our knowledge) to express (6.39d) as a linear DMI in the problem's variables. Moreover, the inclusion $M \in \mathcal{M}_{\phi,\Omega}$ and $M_e \in \mathcal{M}_{\widetilde{\phi},\mathcal{E}_{Q_e}}$ can at best be approximated. In this section we will provide a solution strategy to solve the optimization presented in Problem 6.1.

Since a direct solution of Problem 6.1 is not possible in general, we use the iterative method presented in [7]. This iterative method uses the following observation: if we fix the variables $\{Q(t), Y(t), K(t), \tau(t), \tau_e(t), \alpha(t), \lambda(t)\}$, then Problem 6.1 is convex in M and M_e provided we have an appropriate description of the convex cone $\mathcal{M}_{\phi,\Omega}$ and $\mathcal{M}_{\widetilde{\phi},\mathcal{E}_{Q_e}}$; conversely, for fixed M and M_e, Problem 6.1 is convex in the variables $\{Q(t), Y(t), K(t), \tau(t), \tau_e(t), \alpha(t), \lambda(t)\}$. A natural and common technique is to alternate between fixing one set of variables and solving for the other.

Finally, the nomenclature " γ -iteration" comes from the fact that the matrix

$$M_\gamma = \begin{pmatrix} \gamma^2 I & 0 \\ 0 & -I \end{pmatrix} \tag{6.41}$$

is a valid local multiplier matrix if γ and γ_e are local Lipschitz constants for the nonlinear functions ϕ over the set $\mathcal{M}_{\phi,\Omega}$ and $\widetilde{\phi}$ over the set $\mathcal{M}_{\phi,\mathcal{E}_{Q_e}}$, respectively. By solving for γ and γ_e, we may easily construct local multiplier matrices that can be used to obtain a new iterate for the variables $\{Q(t), Y(t), K(t), \tau(t), \tau_e(t), \alpha(t), \lambda(t)\}$. We now discuss a strategy to compute γ for an arbitrary nonlinear function and sets.

6.5.2 The M-Problem

The M-problem is the subproblem that is obtained by fixing the variables $\{Q(t), Y(t), K(t), \tau(t), \tau_e(t), \alpha(t), \lambda(t)\}$ in Problem 6.1. The quadratic inequality (6.23) under the assumption that $M = M_\gamma$ becomes

$$w^\intercal w \leq \gamma^2 h^\intercal h \quad \Longleftrightarrow \quad \|w\|_2 \leq \gamma \|h\|_2 \tag{6.42}$$

By introducing a matrix $\Delta \in \mathbb{R}^{n_w \times n_h}$, we can then write $w = \Delta h$ and $\|\Delta\|_2 \leq \gamma$ in place of the nonlinear function ϕ in (6.16). The same procedure is applied for (6.25). We introduce matrix $\Delta_e \in \mathbb{R}^{n_u \times n_y}$, and replace the nonlinear function $\widetilde{\phi}$ in (6.16) with $w_e = \Delta_e e$ and $\|\Delta_e\|_2 \leq \gamma_e$. We can rewrite the closed-loop formation (6.16) as

$$\dot{\eta} = (A + B\mathbf{K} + E\Delta(\, F + G\mathbf{K}\,))\,\eta + (E\Delta\widetilde{G} + B\Delta_e)e, \tag{6.43}$$

where

$$\|\Delta\|_2 \leq \gamma, \quad \|\Delta_e\|_2 \leq \gamma_e$$

To compute, or estimate, the value of γ and γ_e we follow the sampling procedure introduced in [7]. We would like to find the points $\eta \in \mathcal{E}_Q$ and $e \in \mathcal{E}_{Q_e}$ that corresponds to the largest value of $\|\Delta\|_2$ and $\|\Delta_e\|_2$, respectively. In our quest to find the most non-conservative local

Lipschitz constants, we attempt to use the dependency of the perception error $e(\eta)$ on η during sampling. To this end, *at each discretized temporal point* t, we spatially sample a set of N_s states from the state-funnel $\mathcal{E}_{Q(t)}$ denoted by $\{\eta_s\}_{s=1}^{N_s}$, with s denoting the spatial iterator. For each η_s, the perception error $e(\eta_s)$ is then sampled from $\mathcal{E}_{Q_e}$.

At time t, given each η_s, we define the inner optimization problem

$$\begin{aligned} \Gamma(\eta_s) = \min_{\Delta,\, \Delta_e} \quad & \|\Delta\|_2 + \|\Delta_e\|_2 \\ \text{s.t.} \quad & \dot{\eta}_s - (A(t) + B(t)K(t))\,\eta_s = E\Delta(F + GK(t))\eta_s + (E\Delta\widetilde{G} + B(t)\Delta_e)\,e(\eta_s) \end{aligned} \tag{6.44}$$

Using

$$x_s = \bar{x}(t) + \eta_s, \quad y_s = \eta_s + e(\eta_s), \quad \xi_s = K(t)y_s, \quad \text{and} \quad u_s = \bar{u}(t) + \xi_s,$$

we can compute the followings at each temporal point t:

$$(A + BK)\,, \quad (F + GK), \quad \dot{\eta}_s = f\,(x_s, u_s) - f(\bar{x}, \bar{u})$$

This inner problem is the key component that allows us to be as non-conservative as possible in our search for local Lipschitz constants. We denote its solution as

$$\Gamma(\eta_s) \coloneqq \|\Delta^*(\eta_s)\|_2 + \|\Delta_e^*(\eta_s)\|_2 \tag{6.45}$$

where $\Delta^*(\eta_s)$ and $\Delta_e^*(\eta_s)$ represent the minimizers of (6.44), i.e., the "smallest" matrices satisfying the nonlinear equations of the dynamic model at each η_s. The local Lipschitz constants (γ, γ_e) are then computed at each temporal point t:

$$\gamma(t) = \|\Delta^*\,(\eta^*(t))\|_2\,, \qquad \gamma_e(t) = \|\Delta_e^*\,(\eta^*(t))\|_2 \tag{6.46}$$

where

$$\eta^*(t) = \arg \max_{\eta \in \{\eta_s\}_{s=1}^{N_s}} \Gamma(\eta) \tag{6.47}$$

The outer optimization problem (6.47) is selecting the sample η_s that produces the maximum value of the inner optimization problem (6.44) across the N_s samples.

This spatiotemporal sampling procedure has been observed to be accurate enough for practical problems with a reasonable number of points N_s (typically on the order of 10^3 or 10^4 for an entire trajectory). Note that because it is possible to solve N_s independent optimization problems, each of which is quite small, this sampling procedure can be done very quickly and is parallelizable.

6.5.3 The Q-Problem

We now describe the Q-problem, which is the subproblem that is obtained by fixing the local multiplier matrices M and M_e in Problem 6.1 and solving for the variables $\{Q(t), Y(t), \tau(t), \tau_e(t), \alpha(t), \lambda(t)\}$.

Theorem 6.7. *For fixed multiplier matrices M and M_e, the output controller $\xi = \mathbf{K}(t)y(t)$ satisfies the conditions in (6.21) if and only if there exists $Q \succeq 0$, $Y \in \mathbb{R}^{n_u \times n_x}$ and scalars $\widetilde{\tau}, \widetilde{\tau}_e, \widetilde{\alpha}, \widetilde{\lambda} > 0$ such that for all $t \in [t_0, t_f]$,*

$$\begin{pmatrix} \mathbf{Q}A^\intercal + A\mathbf{Q} - \dot{\mathbf{Q}} + \mathbf{Q} + B\mathbf{Y} + \mathbf{Y}^\intercal B^\intercal & T_3 \\ * & T_4 \end{pmatrix} \preceq 0, \quad \gamma_e^2\, \widetilde{\boldsymbol{\lambda}} - \widetilde{\boldsymbol{\tau}_e} \leq 0, \quad R\widetilde{\boldsymbol{\alpha}} - \mu\widetilde{\boldsymbol{\lambda}} \leq 0 \tag{6.48}$$

where $\mathbf{F}_{cl} = F\mathbf{Q} + G\mathbf{Y}$, and

$$T_3 = \begin{bmatrix} \widetilde{\boldsymbol{\tau}}\, E & \widetilde{\boldsymbol{\tau}_e} B & \gamma F_{cl}^\intercal & \mathbf{Q}C_\mu^\intercal & s\mathbf{Q} \end{bmatrix}, \quad T_4 = \boldsymbol{diag}\{-\widetilde{\boldsymbol{\tau}} I,\ -\widetilde{\boldsymbol{\tau}_e} I,\ -\widetilde{\boldsymbol{\tau}} I,\ -\widetilde{\boldsymbol{\alpha}} I,\ -\widetilde{\boldsymbol{\lambda}} I\} \tag{6.49}$$

Proof. Consider the differential matrix inequality (6.39d). Due to the form of the local multiplier matrix M_γ in (6.41), we know that $M_{11} \succeq 0$. In this case, the matrix inequality (6.39d) is equivalent to $\boldsymbol{\lambda}R - \boldsymbol{\alpha}\mu \leq 0$ and the following

$$\begin{pmatrix} \mathbf{Q}A^\intercal(t) + A(t)\mathbf{Q} - \dot{\mathbf{Q}} + \mathbf{Q} + B(t)\mathbf{Y} + \mathbf{Y}^\intercal B^\intercal(t) & \mathcal{T}_1 \\ * & \mathcal{T}_2 \end{pmatrix} +$$
$$\begin{pmatrix} \gamma\mathbf{F}_{cl} \\ 0_{2\times 1} \\ \gamma\widetilde{G} \end{pmatrix} (\tau I) \begin{pmatrix} \gamma\mathbf{F}_{cl} \\ 0_{2\times 1} \\ \gamma\widetilde{G} \end{pmatrix}^\intercal + \begin{pmatrix} \mathbf{Q}C_\mu^\intercal \\ 0_{3\times 1} \end{pmatrix} (\alpha I) \begin{pmatrix} C_\mu\mathbf{Q} \\ 0_{1\times 3} \end{pmatrix}^\intercal + \begin{pmatrix} s\mathbf{Q} \\ 0_{3\times 1} \end{pmatrix} (\lambda I) \begin{pmatrix} s\mathbf{Q} \\ 0_{1\times 3} \end{pmatrix}^\intercal \preceq 0$$

where $\mathcal{T}_1 = \begin{bmatrix} E & B & 0 \end{bmatrix}$, and $\mathcal{T}_2 = \mathbf{diag}\Big(\begin{bmatrix} -\tau I & -\tau_e I & \gamma_e^2 \tau_e I - \lambda I \end{bmatrix} \Big)$

If E and $B \neq 0$ then we must have $\tau, \tau_e, \lambda, \alpha > 0$. Hence we can use a Schur complement to arrive at the equivalent condition

$$\begin{pmatrix} \mathbf{Q}A^\intercal(t) + A(t)\mathbf{Q} - \dot{\mathbf{Q}} + \mathbf{Q} + B(t)\mathbf{Y} + \mathbf{Y}^\intercal B^\intercal(t) & \mathcal{T}_3 \\ * & \mathcal{T}_4 \end{pmatrix} \preceq 0, \qquad \gamma_e^2 \tau_e I - \lambda I \leq 0$$

where

$$\mathcal{T}_3 = \begin{bmatrix} E & B & \gamma \mathbf{F}_{cl}^\intercal & \mathbf{Q}C_\mu^\intercal & s\mathbf{Q} \end{bmatrix}, \quad \mathcal{T}_4 = \mathbf{diag}\Big(\begin{bmatrix} -\tau I & -\tau_e I & -\tau^{-1} I & -\alpha^{-1} I & -\lambda^{-1} I \end{bmatrix} \Big).$$

Let $\widetilde{\tau} := \tau^{-1}$, $\widetilde{\tau}_e := \tau_e^{-1}$, $\widetilde{\alpha} := \alpha^{-1}$ and $\widetilde{\lambda} := \lambda^{-1}$. Pre- and post-multiplying the above matrix-inequality by $\mathbf{diag}\Big(\begin{bmatrix} I & \widetilde{\tau} & \widetilde{\tau}_e & I & I & I \end{bmatrix} \Big)$ gives the proof.

□

The continuous-time Q-problem is summarized below in Problem 6.2. It is assumed that a nominal trajectory and system definition are all given.

Problem 6.2. (Q-Problem). Given local multiplier matrices $M \in \mathcal{M}_{\phi,\Omega}$ and $M_e \in \mathcal{M}_{\widetilde{\phi},\mathcal{E}_{Q_e}}$, find the matrix-valued functions of time $Q(t)$, $Y(t)$ and scalars $\widetilde{\tau}(t), \widetilde{\tau}_e(t), \widetilde{\alpha}(t), \widetilde{\lambda}(t)$ that solve the following optimization problem

$$\max_{\substack{\widetilde{\tau}(t),\widetilde{\tau}_e(t),\widetilde{\alpha}(t),\widetilde{\lambda}(t),\\ Q(\cdot),Y(\cdot)}} \quad \log\det Q\left(t_0\right) \tag{6.50a}$$

$$\textbf{subject to} \quad 0 \preceq Q \preceq Q_{\max},\ 0 \leq \widetilde{\tau}(t),\ 0 \leq \widetilde{\tau}_e(t),\ 0 \leq \widetilde{\alpha}(t),\ 0 \leq \widetilde{\lambda}(t), \tag{6.50b}$$

$$\gamma_e^2\, \widetilde{\boldsymbol{\lambda}} - \widetilde{\boldsymbol{\tau}}_{\boldsymbol{e}} \leq 0, \quad R\widetilde{\boldsymbol{\alpha}} - \mu\widetilde{\boldsymbol{\lambda}} \leq 0 \tag{6.50c}$$

$$\begin{pmatrix} \mathbf{Q}A^\intercal + A\mathbf{Q} - \dot{\mathbf{Q}} + \mathbf{Q} + B\mathbf{Y} + \mathbf{Y}^\intercal B^\intercal & T_3 \\ * & T_4 \end{pmatrix} \preceq 0 \tag{6.50d}$$

$$\begin{pmatrix} \left(1 + s\left(1+\rho\right)\right)^{-2} \mathbf{Q} & \mathbf{Y}^\intercal \\ \mathbf{Y} & R_{\max} \end{pmatrix} \succeq 0 \tag{6.50e}$$

where T_3 and T_4 are defined in (6.49).

In order to solve the Q-problem numerically, we make the following assumption regarding the continuous-time problem data.

Assumption 6.8. *Let $M(t)$ denote any of the matrices $Q(t), Q_{\max}(t), Y(t), R_{\max}(t), A(t)$ or $B(t)$. We assume that $M(t)$ can be expressed or approximated as the convex combination*

$$M(t) = \sum_{i=1}^{n_M} \sigma_i(t) M_i \tag{6.51}$$

for some integer $n_M > 1$, constant matrices M_i, and interpolating functions $\sigma_i(t) \geq 0$ such that $\sum_{i=1}^{n_M} \sigma_i(t) = 1$.

By using the *temporal matrix decomposition methods* discussed in [7], the DMI in (6.50d) can be written as a set of $\frac{1}{2} n_M \left(n_M + 1\right)$ LMIs. These LMIs constitute a set of sufficient conditions for the satisfaction of the DMI (6.50d). The constraints (6.50d) can be written as a set of LMIs by using (6.51) and constraining each sum and in the resulting expression individually.

6.5.4 Algorithm Summary

Having defined the $Q-$ and $M-$problems independently, we now collect them together in order to design a convergent algorithm that is capable of solving Problem 6.1. As it is discussed in [7], the premise of the $\gamma-$iteration is that we can decrease the upper bound, $Q_{\max}$, until we are able to synthesize a funnel $Q \approx Q_{\max}$. We measure how close a given solution of the $Q-$problem is to achieving the upper bound of $Q \approx Q_{\max}$ by using the fill ratio, defined by

$$\kappa = \min_{i=1,\ldots,n_x} \left(\frac{\text{proj}_i \, \mathcal{E}_Q}{\text{proj}_i \, \mathcal{E}_{Q_{\max}}} \right)^{1/2} \tag{6.52}$$

Algorithm 4 describes the γ-iteration used to solve the quadratic funnel synthesis problem. For the proof of convergence, see [7].

Algorithm 4 The γ-iteration designed to solve the quadratic funnel synthesis problem.

Require: A nominal trajectory $\{\overline{x}(t), \overline{u}(t)\}_{t=t_0}^{t_f}$, the system matrices for (6.16), a tolerance $\kappa_{tol} \in (0, 1)$.

1: solve (6.34) for $Q_{\max}$ and $R_{\max}$
2: set $M_\gamma \leftarrow 0$ and $M_{\gamma_e} \leftarrow 0$
3: solve the Q-problem to get $\{Q^{(0)}, Y^{(0)}\}$ ▷ Problem 6.2
4: update $Q_{\max}^{(0)} \leftarrow Q^{(0)}$ and $Y_{\max}^{(0)} \leftarrow Y^{(0)}$
5: **for** $k = 1, \cdots$ **do**
6: solve the M-problem to get $M_\gamma^{(k)}$ and $M_{\gamma_e}^{(k)}$ using $\{Q_{\max}^{(k-1)}, Y_{\max}^{(k-1)}\}$ ▷ see (6.47)
7: solve the Q-problem to get $\{Q^{(k)}, Y^{(k)}\}$ using $M_\gamma^{(k)}$ and $M_{\gamma_e}^{(k)}$ ▷ Problem 6.2
8: **if** $\kappa^{(k)}(t_0) \geq \kappa_{tol}$ **then** ▷ see (6.52)
9: Converged.
10: **break**
11: **else**
12: $Q_{\max}^{(k)} \leftarrow \zeta Q_{\max}^{(k-1)}$ and $Y_{\max}^{(k)} \leftarrow \zeta Y_{\max}^{(k-1)}$ ▷ contraction step, see [7]
13: **end if**
14: **end for**

6.6 Numerical Example

In this section, we present a motivating setup for end-to-end vision-based navigation and control of a spacecraft where we can showcase the performance of the perception error modeling and the robust controller synthesis developed in Section 6.4. For that, consider a target tracking problem, where the target spacecraft (so-called the target) is uncontrolled and possibly uncooperative. Thus, by using the image sensors available on the ego spacecraft (so-called ego), we adopt a computer vision approach to predict the target's relative position. developed pipeline is showcased in Figure 6.4.

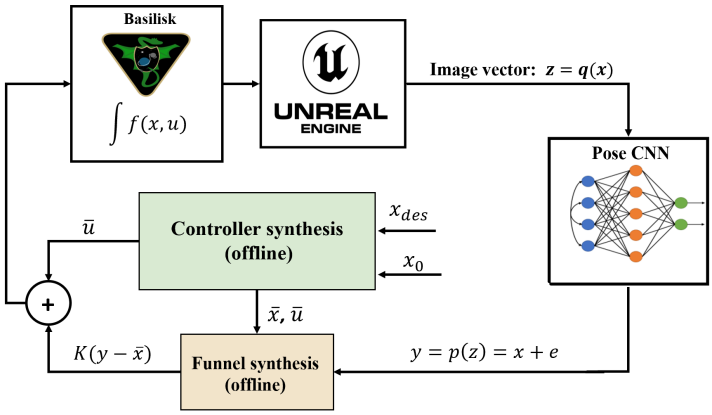


Figure 6.4: The simulation setup developed for emulating the real-world settings for applications of the proposed machinery. The astrodynamics simulator [1] and the developed visual simulator are replicating the actual system with possibility of generating high-quality image data. The state estimated from the image data using the CNN module is then fed into the RCVS controller module for robust feedback signal synthesis.

6.6.1 End-to-end Simulation Pipeline

In this section, we delineate the architecture and implementation details of the simulation pipeline employed to validate our methods. The simulation environment utilizes the Basilisk framework in conjunction with Unreal Engine to replicate the actual system, as described in Section 3.3. This setup emulates the space conditions in Low Earth Orbit (LEO). A sample snapshot of the environment is illustrated in Figure 6.5, while Figure 6.6 presents the overall software architecture. The Basilisk simulation process and the Unreal Engine game thread communicate via a socket-based library called ZeroMQ.

The Basilisk framework includes its own visualization tool and relies on ZeroMQ for its communication model. Consequently, ZeroMQ was selected for thread communication. This communication model employs a streaming approach, wherein the Basilisk process transmits the entire simulation state—including the position, velocity, and orientation of all celestial

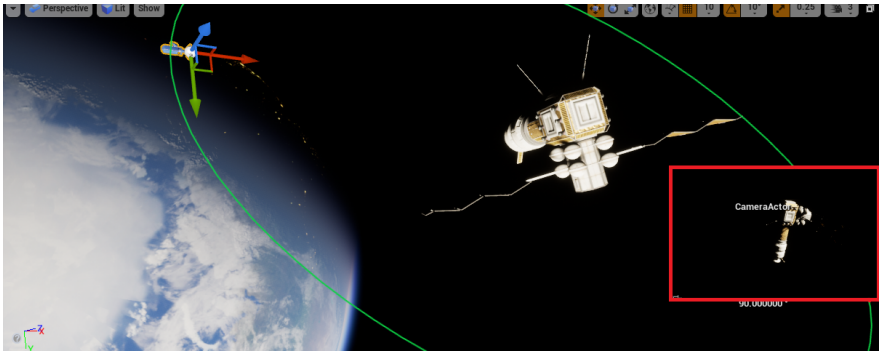


Figure 6.5: A snapshot of the spacecraft simulation setup showing the ego (denoted on the left with its attached body frame) orbiting around an uncontrolled satellite target (the white object in the middle) on a fixed flyby orbit (denoted by a green circle). The in-picture on the right shows the target in the ego's camera frame.

bodies—over the socket at each step. The visualization tool then reads these state messages and updates the locations of the celestial bodies accordingly.

To integrate the CNN process with the remainder of the pipeline, we utilize an indirect communication method. Each process waits for the generation of a file by another process before proceeding. Unreal Engine receives the current simulation state from Basilisk, updates the actor locations, and renders an image from the viewpoint of the Ego spacecraft. The CNN process then waits for this image file to be written, opens it, and performs detection. Subsequently, the CNN process writes the detected pose to a file, which is then read by the Basilisk process. The estimated pose is incorporated as input to the attitude control law, directing the ego spacecraft toward the target spacecraft. Although simplistic, this approach is effective for rapid prototyping and accommodates pose estimation at a lower rate (1 Hz) compared to the simulation rate (10 Hz).

The details of the CNN-based perception module can be found in [6].

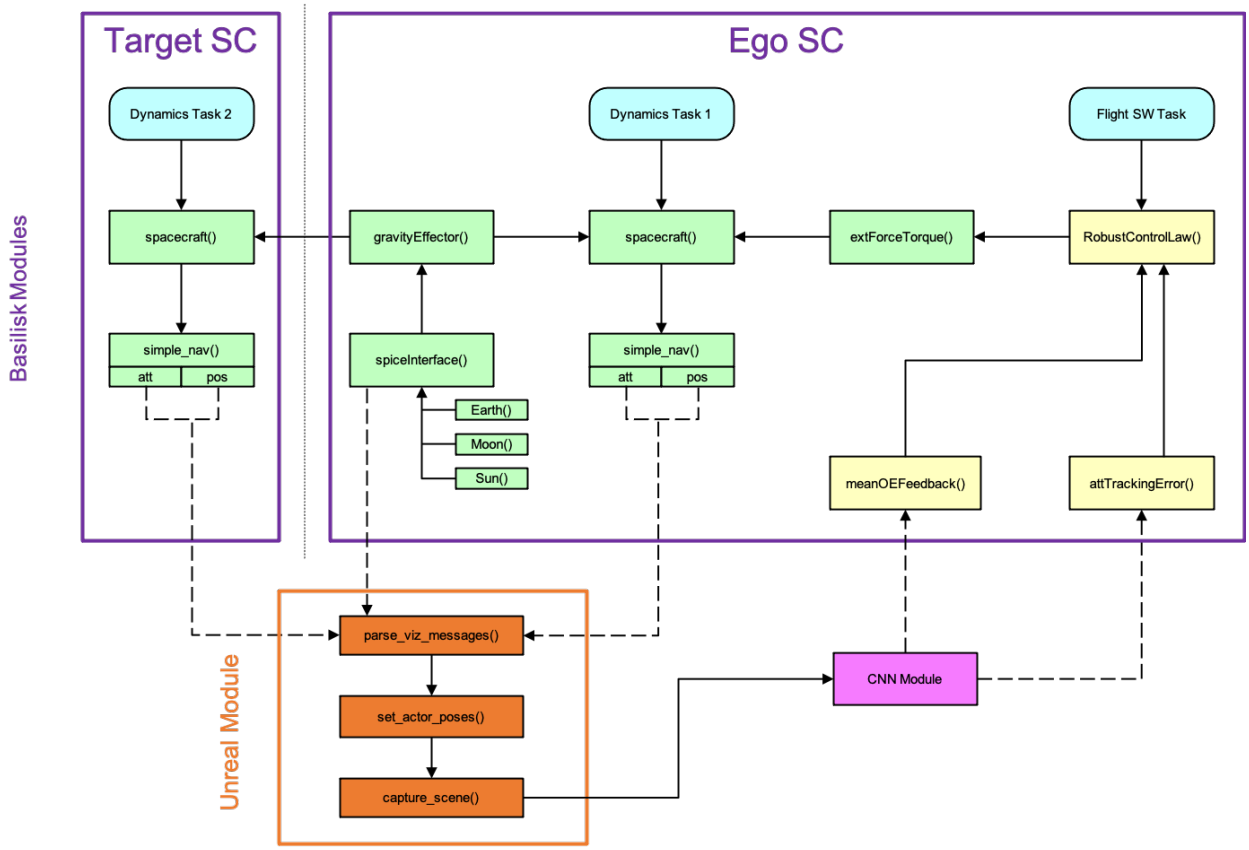


Figure 6.6: The end-to-end simulation pipeline, where all intermediate functions are shown.

6.6.2 System dynamics

Assuming that the ego is inserted in a passive elliptical orbit around the target, the mission starts when the ego's camera points away from the target's center of mass but has the target in its field of view. The goal is to drive the ego's attitude and angular rates such that the target's center of mass is placed at the center of the ego's camera frame. Thus, herein, we lay out the attitude dynamics of the ego using a modified Rodriguez parameters approach.

To obtain ego's attitude dynamics in the Earth fixed frame ($\mathcal{F}_{\mathcal{E}}$), we denote the positive definite inertial tensor by $J \in \mathbb{R}^{3\times3}$ and the angular velocity by $\boldsymbol{\omega}^E \in \mathbb{R}^3$. The torque input is denoted by $\boldsymbol{u} \in \mathbb{R}^3$ and thus, the rigid-body dynamics of the ego in the Earth fixed frame $\mathcal{F}_{\mathcal{E}}$, are given by:

$$J\dot{\boldsymbol{\omega}}^E = \left(J\boldsymbol{\omega}^E\right) \times \boldsymbol{\omega}^E + \boldsymbol{u}^E + \boldsymbol{d}_{\text{ext}}\,,$$

where $\boldsymbol{d}_{\text{ext}}$ represents the external torque acting on the ego, which is assumed to be negligible in this setting for brevity. Next, by using Modified Rodriguez parameters (MRP), $\boldsymbol{q}^E = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^\intercal \in \mathbb{R}^3$ with $\boldsymbol{q}^E = \boldsymbol{e}.\tan\frac{\phi}{2}$ (such that $\boldsymbol{e}$ is the unit vector denoting the axis of rotation and ϕ is the rotation angle), the attitude dynamics can be described as follows:

$$\frac{d}{dt}\begin{pmatrix} \boldsymbol{q}^E \\ \boldsymbol{\omega}^E \end{pmatrix} = f(\boldsymbol{q}^E, \boldsymbol{\omega}^E, \boldsymbol{u}) \;=\; \begin{pmatrix} \frac{1}{2}\left[\mathbf{I}\left(\frac{1-\left(\boldsymbol{q}^E\right)^T\boldsymbol{q}^E}{2}\right) + \boldsymbol{q}^E\left(\boldsymbol{q}^E\right)^T + S(\boldsymbol{q}^E)\right]\boldsymbol{\omega}^E \\ J^{-1}\left(\left(J\boldsymbol{\omega}^E\right) \times \boldsymbol{\omega}^E + \boldsymbol{u}\right) \end{pmatrix} \tag{6.53}$$

where $S(\boldsymbol{q}^E) \coloneqq \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$. Note that target pose estimation depends on both the relative orbital motion between ego and target, and the attitude dynamics of ego. The relative orbital motion follows the Hill-Clohessy-Wiltshire equations [128] that can be solved for an exact solution. We denote the ego's position with respect to the target's body frame of reference, by $r = [r_x,\ r_y,\ r_z]^\intercal \in \mathbb{R}^3$. Following the Hill-Clohessy-Wiltshire equations in phase magnitude form we have

$$\ddot{r}_x = \rho_x \sin(nt + a_x)\,, \quad \ddot{r}_y = \rho_y + 2\rho_x \cos(nt + a_x)\,, \quad \ddot{r}_z = \rho_z \sin(nt + a_x) \tag{6.54}$$

where $n = \sqrt{\mu/a^3}$ and a refers to the orbital radius of the target and μ is the standard gravitational parameter. In (6.54), (a_x, a_y, a_z) denote the semi major axes and (ρ_x, ρ_y, ρ_z) give the phase magnitude of the passive relative elliptical orbit. The formulation of phase magnitude and selection of initial conditions for the ego spacecraft in the reltive elliptical orbit have been detailed in [129]

For this problem, denote the relative states between target and ego spacecraft as $x(t) := \left[r,\ \dot{r},\ q^E,\ w^E\right]^\intercal \in \mathbb{R}^{12}$. We disregard the target's orientation, as it is not needed for the current problem. The relative dynamics are described using (6.54) and desired attitude is calculated using difference between relative target position and ego camera axis. This desired attitude is used as reference state for the attitude dynamics defined by (6.53). Thus the task of the observation module is to detect the target and determine its position in the ego's camera frame. The camera is fixed on the ego's body frame facing its principal x-axis. For simplicity of presentation, we assume the ego's body frame and its camera frame are identical.

Along the relative dynamics, we consider a sensory observation $z(t) \in \mathbb{R}^{m\times m}$ as

$$z(t) = \mathcal{Q}(x(t)) \tag{6.55}$$

that represents the images taken at relative state $x(t)$. $\mathcal{Q} : \mathbb{R}^{12} \to \mathbb{R}^{m\times m}$ is a nonlinear and potentially quite high dimensional mapping. The CNN-based pose estimation techniques are then adopted to represent a *perception map* $p : \mathbb{R}^{m\times m} \to \mathbb{R}^3$ that imperfectly recovers the relative position $x(t)$ in the camera frame; that is,

$$p(z(t)) = C\,x(t) + e_p(x(t)) \tag{6.56}$$

where e_p denotes the perception error that is dependent on the relative states. To model the perception map and study its error characteristics, as discussed in section 6.3, we design a nominal nonlinear controller for given initial relative states $x(0)$. Such nominal controller generates a nominal trajectory $\{\overline{x}(t), \overline{u}(t)\}_{t=t_0}^{t_f}$, depicted in Figure 6.7, as seen in the ego's camera frame. We then collect the initial training dataset $\mathcal{S}_0 = \{(x_d, z_d)\}_{d=1}^{N_0}$ from the nominal trajectory and train the neural network[5].

[5]See [6] for further details regarding the neural network setup.

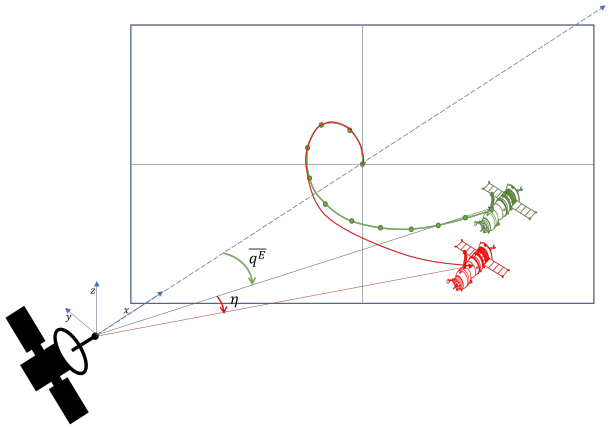


Figure 6.7: Nominal trajectory in green as seen in the camera frame. For some initial condition which deviate form the nominal as shown in red, we use the robust controller such that the error between the new state and nominal given by η is driven to zero.

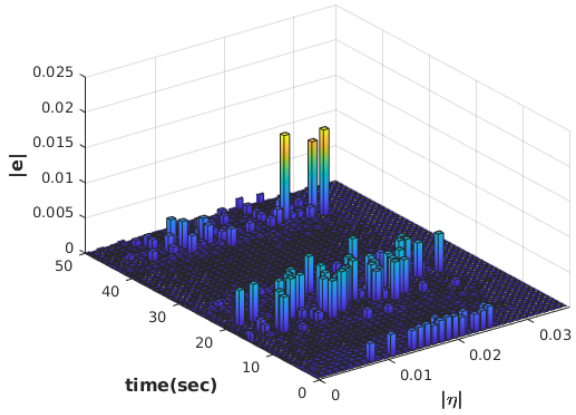


Figure 6.8: The error profile in the neighborhood of the nominal trajectory, for $T = 50$ seconds given 100 deviation samples per time stamp.

Based on the assumption that the ego is in a fixed flyby orbit around the target, we focus our efforts on designing the torque vector $\mathbf{u}$ that tracks the target position in the ego's

camera frame. Hence, starting off from the nominal trajectory with error, we define

$$\eta(t) := C\ x(t) - C\ \overline{x}(t) = \begin{pmatrix} q^E(t) \\ w^E(t) \end{pmatrix} - \begin{pmatrix} \overline{q^E}(t) \\ \overline{w^E}(t) \end{pmatrix}$$

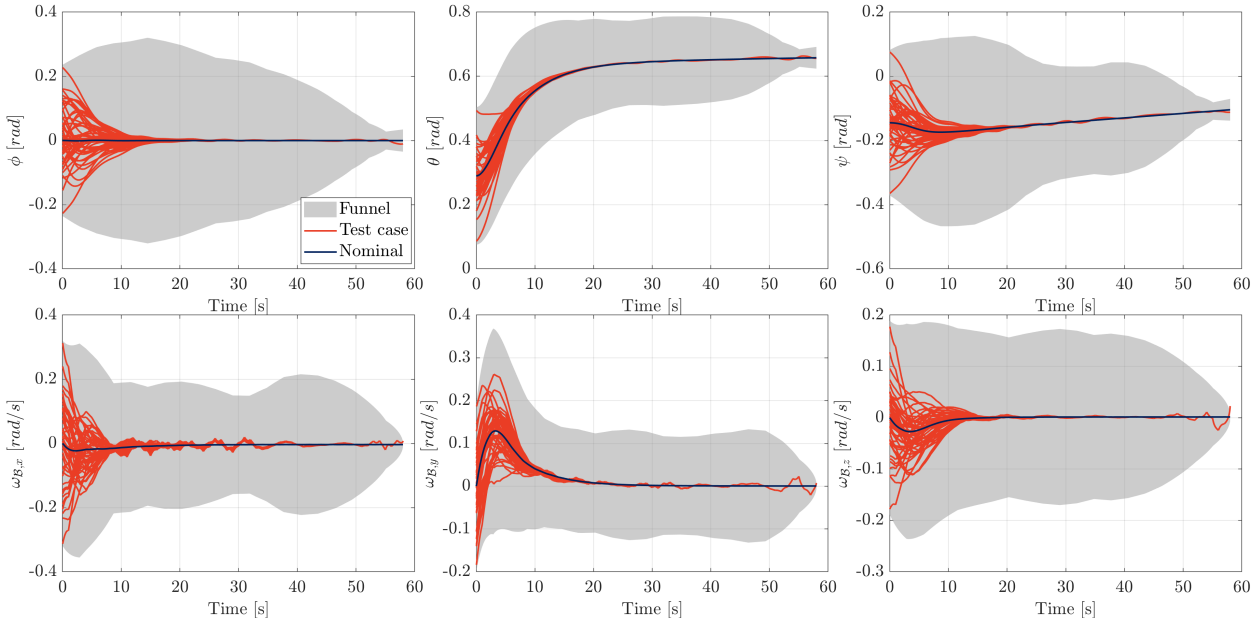
as the tracking error and $\xi = u - \bar{u}$ as the correction signal. Figure 6.8 showcases the error around the nominal trajectory with respect to time and η.

The matrices A and B are the partial derivatives of the attitude dynamics in (6.53) along the nominal trajectory, and the parameters F, G and E are constructed using a total of four ($n_h = 4$) nonlinear channels to be

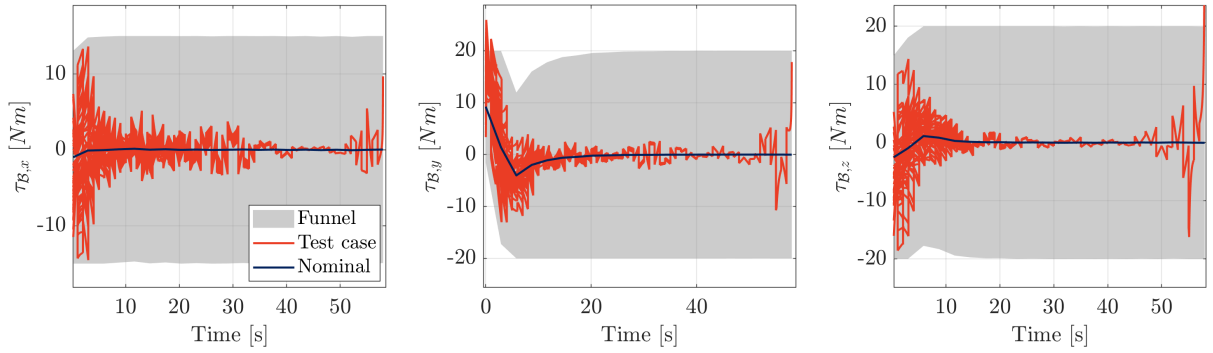
$$F = \begin{pmatrix} I_3 & 0_{3\times3} \\ I_3 & I_3 \\ I_3 & I_3 \\ I_3 & I_3 \end{pmatrix}, \qquad G = 0_{12\times6}, \qquad E = \begin{pmatrix} I_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_3 & I_3 & I_3 \end{pmatrix}.$$

Note that for this problem $n_x = 6$, $n_u = 3$ and $J = \mathbf{diag}\{900,\ 900,\ 900\}\ \ kg\,m^2$. The quadratic funnel synthesis problem in section 6.5, provides a robust controller that renders a UUB closed-loop system.

Figure 6.9 depicts the computed quadratic funnel. In Figure 6.9a, the ellipsoid $\mathcal{E}_Q$ is projected onto each state dimension and depicted as the shaded grey area. The red trajectories correspond to test cases for which an initial condition was randomly (uniformly) selected from the ellipsoid $\mathcal{E}_Q(t_0)$, and the nominal control and correction law was used to numerically integrate the equations of motion according to (6.40). Figure 6.9b shows the ellipsoid $\mathcal{E}_{R_{\max}}$ projected into each control dimension along with the corresponding control trajectories from each test case.



(a) In state space $\mathcal{E}_Q$



(b) In control space $\mathcal{E}_{R_{\max}}$

Figure 6.9: The quadratic funnel computed by the γ-iteration for the target tracking problem. The initial condition of each test case was randomly sampled from the funnel entry.

Part III

CONCLUSION AND FUTURE DIRECTIONS

This dissertation explored iterative data-driven control and perception-aware trajectory planning and control, providing applicable methods for some of the significant challenges in autonomous systems' control theory. The findings and methodologies developed in this dissertation open several avenues for future research.

Part I focused on iterative data-driven control for stabilization and performance enhancement in systems with unstable subsystems. In Chapter 4, we introduced the concept of ***"regularizability"***, a novel system-theoretic notion that quantifies the capacity for data-driven finite-time regulation of linear systems. This notion is in contrast to the analysis of the asymptotic behavior of the system, which is commonly characterized by stabilizability/controllability. We developed the Data-Guided Regulator (DGR), an online iterative feedback regulator for partially unknown, potentially unstable linear systems. DGR effectively utilizes streaming data from a single trajectory to regulate unknown systems and generate informative data for stabilization and system identification. Along the way, we have provided another system theoretic notion referred to as ***"instability number"*** to analyze DGR's performance and derive bounds on the system's trajectory over finite time intervals. The application of DGR to a highly maneuverable unstable aircraft demonstrated its integration with other data-driven methods, achieving improved performance through better numerical conditioning. Future directions include extending results to noisy data, considering unknown input matrices, addressing partial observation of the system's trajectory, and incorporating input constraints like rate limits.

Chapter 5 extended the notion of "regularizability" to nonlinear dynamical systems with unknown time-varying disturbances. We characterize ***"rapid-regularizability"*** for a linear

time-varying (LTV) representation of the nonlinear system. The unknown disturbances affecting the system states were modeled as a *high-dimensional* linear time-invariant (LTI) system with unknown dynamics. Furthermore, we assumed that the unknown disturbance dynamics are observable through measured system states. Subsequently, we proposed the Data-Guided Regulator for Adaptive Nonlinear Control algorithm (DG-RAN), which utilizes discrete time-series data from a single trajectory to achieve finite-time regulation of system states. DG-RAN efficiently addresses the limitations of existing adaptive control approaches by providing an iterative synthesis procedure that adapts to changing environments while generating informative data for disturbance dynamics identification. This research contributed to the integration of learning, optimization, and control by introducing a novel framework for designing adaptive data-driven controllers, enabling autonomous systems to adapt quickly to new environments while ensuring safety and control-theoretic guarantees.

- ***Future Directions for Part I:***

    1. *Noisy Data and Unknown Input Matrices:* Extending the DGR framework to handle noisy data and unknown input matrices will enhance its applicability in real-world scenarios where perfect data and complete system knowledge are rarely available. This involves developing robust estimation techniques and adaptive algorithms that can compensate for these uncertainties.
    2. *Partial Observation and Multi-scale Dynamics:* Investigating state regulation in scenarios with partial observation of the system's trajectory or systems with multi-scale dynamics presents a challenging yet essential extension. This requires integrating advanced state estimation methods and multi-scale modeling techniques to ensure effective control in complex environments.
    3. *Input Constraints and Practical Implementations:* Addressing practical input constraints, such as rate limits, within the DGR framework will make the method more applicable to real-world systems. Analyzing the resulting closed-loop trajectory

under these constraints will provide deeper insights into the practical feasibility and performance of the proposed control methods.

4. *Sample Complexity and Nonlinear Dynamics:* The DG-RAN method employs a uniformly random sampling technique to estimate local Lipschitz bounds for the higher-order terms of the system dynamics. A more precise sampling technique could improve these local bounds, thereby enhancing the controller's performance. Understanding the relationship between the sampling techniques and system nonlinearity is crucial, as the effects of disturbances on such nonlinearity can significantly impact the system's behavior. Further exploration of this relationship could lead to more accurate and robust control strategies by better capturing the complexities introduced by disturbances.

5. *Real-world Experiments and Validation:* Conducting real-world experiments to validate the performance of the proposed DG-RAN algorithm will be crucial for transitioning from theoretical models to practical applications. This involves implementing the algorithms on actual autonomous systems and evaluating their performance in dynamic and uncertain environments.

Part II addressed a problem in perception-aware trajectory planning and control, focusing on the practical application of deep learning algorithms in processing high-dimensional data for feedback control. In Chapter 6, we developed a robust controller synthesis method for autonomous vision-based system control, using a Convolutional Neural Network (CNN) as a perception module. We characterized the associated error along a nominal trajectory defined by a nonlinear controller and developed a robust control synthesis procedure based on the invariant funnel concept. This method robustly synthesized a feedback signal directly from image data, closing the control loop with an image sensor. We validated the proposed approach using an astrodynamics simulator and a photo-realistic simulator in a target spacecraft tracking problem. Future directions include expanding the approach to multi-agent formation control and reconfiguration applications, as well as incorporating data-driven

dynamics modeling into the image-based neural network architecture to reduce perception error.

- ***Future Directions for Part II:***

    1. *Physics-Informed Neural Networks:* Incorporating the object's dynamics into the perception module and integrating advanced state estimation methods can potentially reduce perception error. Leveraging data-driven techniques to simultaneously model the perception error could eliminate the dependency on the nominal trajectory and handle a wide range of operational conditions. This integration will enhance the robustness and accuracy of perception-aware control systems, providing more reliable feedback control in diverse environments.
    2. *Multi-Agent Formation Control and Reconfiguration:* Expanding the robust control synthesis approach to multi-agent formation control and reconfiguration will address the growing need for coordinated control in space missions. This includes developing decentralized control strategies and exploring the interplay between individual spacecraft dynamics and collective behavior.

Addressing these future research directions will further extend this dissertation's contributions, providing more robust, adaptive, and reliable control solutions for autonomous systems in various safety-critical applications.

Bibliography

[1] P. W. Kenneally, S. Piggott, and H. Schaub, "Basilisk: A flexible, scalable and modular astrodynamics simulation framework," *Journal of Aerospace Information Systems*, vol. 17, no. 9, pp. 496–507, 2020.

[2] N. Rahimi and M. Mesbahi, "Data-guided regulator for adaptive nonlinear control," in *AIAA SciTech 2024 Forum*, p. 0090, 2024.

[3] N. Rahimi, S. Talebi, A. Deole, M. Mesbahi, S. Bandyopadhyay, and A. Rahmani, "Robust controller synthesis for vision-based spacecraft guidance and control," in *AIAA SciTech 2022 Forum*, p. 2213, 2022.

[4] S. Talebi, S. Alemzadeh, N. Rahimi, and M. Mesbahi, "On regularizability and its application to online control of unstable lti systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6413–6428, 2021.

[5] S. Talebi, S. Alemzadeh, N. Rahimi, and M. Mesbahi, "Online regulation of unstable linear systems from a single trajectory," in *59th IEEE Conference on Decision and Control (CDC)*, pp. 4784–4789, 2020.

[6] J. Becktor, W. Seto, A. Deole, S. Bandyopadhyay, N. Rahimi, s. Talebi, M. Mesbahi, and A. Rahmani, "Robust vision-based multi-spacecraft guidance navigation control using cnn-based pose estimation," in *2022 IEEE Aerospace Conference*, pp. –, IEEE, 2022.

[7] T. P. Reynolds, D. Malyuta, M. Mesbahi, B. Açıkmeşe, and J. M. Carson, "Funnel synthesis for the 6-dof powered descent guidance problem," *AIAA SciTech 2021 Forum*, pp. 1–21, 2021.

[8] E. F. S. Boyd, L. El Ghaoui and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, vol. 3. Studies in Applied Mathematics, SIAM, 1994.

[9] B. Açıkmeşe, J. M. Carson III, and D. S. Bayard, "A robust model predictive control algorithm for incrementally conic uncertain/nonlinear systems," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 5, pp. 563–590, 2011.

[10] B. Açıkmeşe and M. Corless, "Stability analysis with quadratic lyapunov functions: Some necessary and sufficient multiplier conditions," *Systems & Control Letters*, vol. 57, no. 1, pp. 78–94, 2008.

[11] L. D'Alto and M. Corless, "Incremental quadratic stability," *Numerical Algebra, Control & Optimization*, vol. 3, no. 1, p. 175, 2013.

[12] T. P. Reynolds, *Computational Guidance and Control for Aerospace Systems*. PhD thesis, University of Washington, 2020.

[13] T. Söderström and P. Stoica, *System Identification*. Prentice-Hall, 1989.

[14] P. C. Young, *Recursive Estimation and Time-Series Analysis: An Introduction*. Springer Science & Business Media, 2012.

[15] M. Verhaegen and P. Dewilde, *Subspace Model Identification*. Springer, 1992.

[16] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach*. IEEE Press, 2001.

[17] S. Haykin, *Adaptive Filter Theory*. Pearson, 2001.

[18] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, 1960.

[19] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010.

[20] I. Jolliffe, *Principal Component Analysis*. Springer, 2002.

[21] D. Westwick and R. Kearney, *Identification of Nonlinear Physiological Systems*. IEEE Press, 2003.

[22] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, 1999.

[23] I. Leontaritis and S. Billings, "Input-output parametric models for non-linear systems part i: Deterministic nonlinear systems," *International Journal of Control*, 1985.

[24] J. P. Hespanha, *Linear Systems Theory*. Princeton University Press, 2018.

[25] G. Stein, "Respect the unstable," *IEEE Control Systems Magazine*, vol. 23, no. 4, pp. 12–25, 2003.

[26] I. Khalil, J. Doyle, and K. Glover, *Robust and optimal control*. Prentice-Hall, 1996.

[27] R. P. Sree and M. Chidambaram, *Control of Unstable Systems*. Alpha Science Int'l Ltd., 2006.

[28] S. Skogestad, K. Havre, and T. Larsson, "Control limitations for unstable plants," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 485–490, 2002.

[29] D. G. Taylor, P. V. Kokotovic, R. Marino, and I. Kanellakopoulos, "Adaptive regulation of nonlinear systems with unmodeled dynamics," in *1988 American Control Conference*, pp. 360–365, IEEE, 1988.

[30] M. de Oliveira, J. Bernussou, and J. Geromel, "A new discrete-time robust stability condition," *Systems & Control Letters*, vol. 37, no. 4, pp. 261 – 265, 1999.

[31] M. de Oliveira, J. Geromel, and L. Hsu, "LMI characterization of structural and robust stability: the discrete-time case," *Linear Algebra and its Applications*, vol. 296, no. 1, pp. 27 – 38, 1999.

[32] G. F. Franklin, J. D. Powell, A. Emami-Naeini, and J. D. Powell, *Feedback control of dynamic systems*, vol. 4. Prentice-Hall Upper Saddle River, 2002.

[33] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*. Courier Corporation, 2013.

[34] K. J. Åström and B. Wittenmark, *Adaptive Control*. Courier Corporation, 2013.

[35] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *International Conference on Intelligent Robots and Systems*, pp. 4906–4913, IEEE, 2012.

[36] V. Kumar, E. Todorov, and S. Levine, "Optimal control with learned local models: Application to dexterous manipulation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 378–383, IEEE, 2016.

[37] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1714–1721, IEEE, 2017.

[38] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, 2016.

[39] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[40] H. Kim and A. Y. Ng, "Stable adaptive control with online learning," in *Advances in Neural Info. Processing Systems*, pp. 977–984, 2005.

[41] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2020.

[42] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *18th European Control Conference (ECC)*, pp. 307–312, 2019.

[43] S. Baros, C.-Y. Chang, G. E. Colon-Reyes, and A. Bernstein, "Online data-enabled predictive control," *arXiv preprint arXiv:2003.03866*, 2020.

[44] H. J. Van Waarde, J. Eising, H. L. Trentelman, and M. K. Camlibel, "Data informativity: a new perspective on data-driven analysis and control," *IEEE Transactions on Automatic Control*, pp. 1–1, 2020.

[45] Y. Yu, S. Talebi, H. J. van Waarde, U. Topcu, M. Mesbahi, and B. Açıkmeşe, "On controllability and persistency of excitation in data-driven control: Extensions of Willems' Fundamental Lemma," in *60th IEEE Conference on Decision and Control (CDC)*, 2021 (to appear).

[46] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1702–1717, 2021.

[47] R. Lozano, P. Castillo, P. Garcia, and A. Dzul, "Robust prediction-based control for unstable delay systems: Application to the yaw control of a mini-helicopter," *Automatica*, vol. 40, no. 4, pp. 603–612, 2004.

[48] D. P. Bertsekas, "Approximate policy iteration: A survey and some new methods," *Journal of Control Theory and Applications*, vol. 9, no. 3, pp. 310–335, 2011.

[49] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[50] S. M. Kakade, "A natural policy gradient," in *Advances in Neural Information Processing Systems*, pp. 1531–1538, 2002.

[51] M. Fazel, R. Ge, S. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," in *International Conference on Machine Learning*, pp. 1467–1476, PMLR, 2018.

[52] S. Alemzadeh and M. Mesbahi, "Distributed Q-learning for dynamically decoupled systems," in *American Control Conference (ACC)*, pp. 772–777, 2019.

[53] K. Ozcaldiran and F. L. Lewis, "On the regularizability of singular systems," *IEEE Transactions on Automatic Control*, vol. 35, no. 10, pp. 1156–1160, 1990.

[54] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, 2009.

[55] P. Jagtap, G. J. Pappas, and M. Zamani, "Control barrier functions for unknown nonlinear systems using gaussian processes," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3699–3704, 2020.

[56] N. Agarwal, B. Bullins, E. Hazan, S. Kakade, and K. Singh, "Online control with adversarial disturbances," in *International Conference on Machine Learning*, pp. 111–119, PMLR, 2019.

[57] S. Lale, K. Azizzadenesheli, B. Hassibi, and A. Anandkumar, "Adaptive control and regret minimization in linear quadratic gaussian (LQG) setting," *arXiv preprint arXiv:2003.05999*, 2020.

[58] A. Cohen, T. Koren, and Y. Mansour, "Learning linear-quadratic regulators efficiently with only $\sqrt{T}$ regret," in *International Conference on Machine Learning*, pp. 1300–1309, PMLR, 2019.

[59] M. Simchowitz and D. Foster, "Naive exploration is optimal for online lqr," in *International Conference on Machine Learning*, pp. 8937–8948, PMLR, 2020.

[60] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[61] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[62] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.

[63] M. P. Deisenroth, C. E. Rasmussen, and D. Fox, "Learning to control a low-cost manipulator using data-efficient reinforcement learning," *Robotics: Science and Systems*, pp. 57–64, 2012.

[64] M. P. Deisenroth, G. Neumann, J. Peters, *et al.*, "A survey on policy search for robotics," *Foundations and Trends® in Robotics*, vol. 2, no. 1–2, pp. 1–142, 2013.

[65] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, pp. 621–635, Springer, 2018.

[66] P. W. Kenneally, S. Piggott, and H. Schaub, "Basilisk: a flexible, scalable and modular astrodynamics simulation framework," *Journal of Aerospace Information Systems*, vol. 17, no. 9, pp. 496–507, 2020.

[67] A. Wagenmaker and K. Jamieson, "Active learning for identification of linear dynamical systems," in *Conference on Learning Theory*, pp. 3487–3582, PMLR, 2020.

[68] E. Nozari, Y. Zhao, and J. Cortés, "Network identification with latent nodes via autoregressive models," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 2, pp. 722–736, 2017.

[69] M. Sharf and D. Zelazo, "Network identification: A passivity and network optimization approach," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 2107–2113, IEEE, 2018.

[70] Z. Hou, H. Gao, and F. L. Lewis, "Data-driven control and learning systems," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4070–4075, 2017.

[71] M. Sedghi, M. Geo, and G. Atia, "A multi-criteria approach for fast and robust representative selection from manifolds," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020.

[72] M. K. S. Faradonbeh, A. Tewari, and G. Michailidis, "Finite-time adaptive stabilization of linear systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3498–3505, 2019.

[73] S. Dean, S. Tu, N. Matni, and B. Recht, "Safely learning to control the constrained linear quadratic regulator," in *American Control Conference (ACC)*, pp. 5582–5588, IEEE, 2019.

[74] A. Alaeddini, S. Alemzadeh, A. Mesbahi, and M. Mesbahi, "Linear model regression on time-series data: non-asymptotic error bounds and applications," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 2259–2264, 2018.

[75] T. Sarkar, A. Rakhlin, and M. A. Dahleh, "Finite time LTI system identification," *Journal of Machine Learning Research*, vol. 22, pp. 26:1–26:61, 2021.

[76] J. Berberich, A. Koch, C. W. Scherer, and F. Allgöwer, "Robust data-driven state-feedback design," in *American Control Conference (ACC)*, pp. 1532–1538, 2020.

[77] S. Oymak and N. Ozay, "Non-asymptotic identification of LTI systems from a single trajectory," in *American Control Conference (ACC)*, pp. 5655–5661, IEEE, 2019.

[78] S. Fattahi, N. Matni, and S. Sojoudi, "Learning sparse dynamical systems from a single sample trajectory," in *IEEE 58th Conference on Decision and Control (CDC)*, pp. 2682–2689, 2019.

[79] A. Tsiamis and G. J. Pappas, "Finite sample analysis of stochastic system identification," in *IEEE 58th Conference on Decision and Control (CDC)*, pp. 3648–3654, 2019.

[80] L. Ljung, *System Identification*. Wiley Online Library, 2001.

[81] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Courier Corporation, 2012.

[82] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Foundations of Computational Mathematics*, 2019.

[83] S. Tu and B. Recht, "The gap between model-based and model-free methods on the linear quadratic regulator: An asymptotic viewpoint," in *Proceedings of the 32nd Conference on Learning Theory*, vol. 99, pp. 3036–3083, PMLR, 25–28 Jun 2019.

[84] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *Proceedings of 1994 American Control Conference*, vol. 3, pp. 3475–3479, IEEE, 1994.

[85] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[86] A. D. González, A. Chapman, L. Dueñas-Osorio, M. Mesbahi, and R. M. D'Souza, "Efficient infrastructure restoration strategies using the recovery operator," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 12, pp. 991–1006, 2017.

[87] K. Manohar, E. Kaiser, S. L. Brunton, and J. N. Kutz, "Optimized sampling for multiscale dynamics," *Multiscale Modeling & Simulation*, vol. 17, no. 1, pp. 117–136, 2019.

[88] D. S. Naidu and A. J. Calise, "Singular perturbations and time scales in guidance and control of aerospace systems: A survey," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 6, pp. 1057–1078, 2001.

[89] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 2012.

[90] J. Simon and S. K. Mitter, "A theory of modal control," *Information and Control*, vol. 13, no. 4, pp. 316–353, 1968.

[91] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.

[92] N. E. Friedkin and E. C. Johnsen, "Social influence and opinions," *Journal of Mathematical Sociology*, vol. 15, no. 3-4, pp. 193–206, 1990.

[93] C. D. Meyer, Jr, "Generalized inversion of modified matrices," *SIAM Journal on Applied Mathematics*, vol. 24, no. 3, pp. 315–323, 1973.

[94] J. T. Bosworth, *Linearized Aerodynamic and Control Law Models of X-29A Airplane and Comparison with Flight Data*, vol. 4356. NASA, 1992.

[95] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice-Hall, Inc., 1996.

[96] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[97] M. O'Connell, G. Shi, X. Shi, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural-fly enables rapid learning for agile flight in strong winds," *Science Robotics*, vol. 7, no. 66, p. eabm6597, 2022.

[98] A. B. Açikmeşe and M. Corless, "Robust tracking and disturbance rejection of bounded rate signals for uncertain/non-linear systems," *International Journal of Control*, vol. 76, no. 11, pp. 1129–1141, 2003.

[99] J.-J. Slotine and J. Coetsee, "Adaptive sliding controller synthesis for non-linear systems," *International Journal of Control*, vol. 43, no. 6, pp. 1631–1651, 1986.

[100] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987.

[101] S. S. Sastry and A. Isidori, "Adaptive control of linearizable systems," *IEEE Transactions on Automatic Control*, vol. 34, no. 11, pp. 1123–1131, 1989.

[102] I. Kanellakopoulos, P. V. Kokotovic, and R. Marino, "An extended direct scheme for robust adaptive nonlinear control," *Automatica*, vol. 27, no. 2, pp. 247–255, 1991.

[103] G. Shi, K. Azizzadenesheli, M. O'Connell, S.-J. Chung, and Y. Yue, "Meta-adaptive nonlinear control: Theory and algorithms," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[104] N. M. Boffi, S. Tu, and J.-J. E. Slotine, "Regret bounds for adaptive nonlinear control," in *Learning for Dynamics and Control*, pp. 471–483, PMLR, 2021.

[105] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Foundations of Computational Mathematics*, vol. 20, no. 4, pp. 633–679, 2020.

[106] S. Lale, K. Azizzadenesheli, B. Hassibi, and A. Anandkumar, "Logarithmic regret bound in partially observable linear dynamical systems," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20876–20888, 2020.

[107] S. Kakade, A. Krishnamurthy, K. Lowrey, M. Ohnishi, and W. Sun, "Information theoretic regret bounds for online nonlinear control," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15312–15325, 2020.

[108] Y. Li, X. Chen, and N. Li, "Online optimal control with linear dynamics and predictions: Algorithms and regret analysis," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[109] C. Yu, G. Shi, S.-J. Chung, Y. Yue, and A. Wierman, "The power of predictions in online control," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1994–2004, 2020.

[110] Y. Lin, Y. Hu, G. Shi, H. Sun, G. Qu, and A. Wierman, "Perturbation-based regret analysis of predictive control in linear time varying systems," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[111] Y. Abbasi-Yadkori and C. Szepesvári, "Regret bounds for the adaptive control of linear quadratic systems," in *Proceedings of the 24th Annual Conference on Learning Theory*, pp. 1–26, JMLR Workshop and Conference Proceedings, 2011.

[112] H. Mania, S. Tu, and B. Recht, "Certainty equivalence is efficient for linear quadratic control," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[113] E. Hazan, S. Kakade, and K. Singh, "The nonstochastic control problem," in *Algorithmic Learning Theory*, pp. 408–421, PMLR, 2020.

[114] B. T. Lopez and J.-J. E. Slotine, "Universal adaptive control of nonlinear systems," *IEEE Control Systems Letters*, vol. 6, pp. 1826–1830, 2021.

[115] T. Reynolds, D. Malyuta, M. Mesbahi, B. Acikmese, and J. M. Carson, "Funnel synthesis for the 6-dof powered descent guidance problem," in *AIAA SciTech 2021 Forum*, p. 0504, 2021.

[116] T. P. Reynolds, M. Szmuk, D. Malyuta, M. Mesbahi, B. Açıkmeşe, and J. M. Carson III, "Dual quaternion-based powered descent guidance with state-triggered constraints," *Journal of Guidance, Control, and Dynamics*, 2020.

[117] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 55–81, 2012.

[118] F. Alhwarin, A. Ferrein, and I. Scholl, "Ir stereo kinect: Improving depth images by combining structured light with ir stereo," in *13th Pacific Rim International Conference on Artificial Intelligence*, pp. 409–421, Springer, 2014.

[119] S. Dean, N. Matni, B. Recht, and V. Ye, "Robust guarantees for perception-based control," *Proceedings of Machine Learning Research*, pp. 1–21, 2019.

[120] H. K. Khalil, *Nonlinear Systems*, vol. 3. Prentice-Hall, 2002.

[121] P. F. Proenca and Y. Gao, "Deep learning for spacecraft pose estimation from photorealistic rendering," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 6007–6013, 2020.

[122] S. Sharma, C. Beierle, and S. D'Amico, "Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks," in *2018 IEEE Aerospace Conference*, pp. 1–12, 2018.

[123] S. Sonawani, R. Alimo, R. Detry, D. Jeong, A. Hess, and H. B. Amor, "Assistive relative pose estimation for on-orbit assembly using convolutional neural networks," *arXiv preprint arXiv:2001.10673*, 2020.

[124] A. Harvard, V. Capuano, E. Y. Shao, and S.-J. Chung, "Spacecraft pose estimation from monocular images using neural network based keypoints and visibility maps," in *AIAA SciTech 2020 Forum*, p. 1874, 2020.

[125] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, 2017.

[126] S. M. LaValle, J. J. Kuffner, B. Donald, *et al.*, "Rapidly-exploring random trees: Progress and prospects," *Algorithmic and Computational Robotics: New Directions*, vol. 5, pp. 293–308, 2001.

[127] Y. Mao, D. Dueri, M. Szmuk, and B. Açıkmeşe, "Successive convexification of non-convex optimal control problems with state constraints," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4063–4069, 2017.

[128] W. Clohessy and R. Wiltshire, "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.

[129] K. Alfriend, S. R. Vadali, P. Gurfil, J. How, and L. Breger, *Spacecraft Formation Flying: Dynamics, Control and Navigation*, vol. 2. Elsevier, 2009.