Control Methodologies for Systems with Set-Valued Uncertainties

Aditya Deole

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

Doctor of Philosophy

University of Washington

2025

Reading Committee:
Mehran Mesbahi, Chair
Kristi Morgansen
Amir Taghvaei

Program Authorized to Offer Degree:
Aeronautics and Astronautics Engineering

University of Washington

Abstract

Control Methodologies for Systems with Set-Valued Uncertainties

Aditya Deole

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

This dissertation develops control methodologies for systems with set-valued uncertainties in modeling and estimation, with applications spanning spacecraft navigation and neuromodulation. The work is organized into two major parts. The first part addresses estimation-related uncertainties in vision-guided navigation and their integration into planning and control. The second part focuses on modeling uncertainty in neuronal systems, presenting a controller design and model inference framework for neuromodulation.

In the context of spacecraft navigation, we design a pose-estimation pipeline supported by a photorealistic simulation environment for satellite rendezvous operations. A Machine Learning (ML)-based platform is developed to detect the pose of a target spacecraft, and the simulation environment is used to generate test and validation data with a minimal simulation-to-reality (sim2real) gap. The platform also serves as a tool for modeling ML-based uncertainties, thereby enabling robust controller design. Building on this foundation, two approaches are proposed for incorporating ML-based estimation into navigation systems. The first introduces a controller design methodology that constructs invariant funnels for slope-bounded uncertainty models around nominal trajectories. The second employs a passivity-based framework to characterize uncertainties that define a family of feasible controllers. Furthermore, we demonstrate that multi-agent consensus, viewed as an interconnection of passive agents, can enhance estimation performance in distributed settings.

We further investigate estimation-aware trajectory design for improving the performance of state-dependent sensors such as perception maps. A class of state-dependent, set-valued output uncertainty models is formalized as state-to-output uncertainty set maps. An observability-based metric is introduced to quantify the estimator's sensitivity to output perturbations, and this metric is optimized to generate trajectories that improve estimation performance. Extensions of this framework to multi-agent trajectory planning are also presented.

The final part of the dissertation develops a feedback control framework for neuromodulation. By analyzing neuronal system trajectories during experimental sessions, we show that average neuronal dynamics in closed-loop scenarios can be approximated as a linear parameter-dependent system, with parameter-dependent internal processes. For a fixed parameter, the trial-averaged dynamics exhibit closed-loop linear behavior. A proportional–integral (PI) feedback controller is demonstrated to effectively track reference signals over a finite horizon, outperforming feedforward control in both tracking accuracy and disturbance rejection, while also reducing trial-to-trial variability. Moreover, in a "reward-induced" brain state with more consistent parameters, a sample-based approach is shown to enable controller optimization.

Together, these contributions advance the integration of machine learning, robust control, and trajectory optimization in the presence of set-valued uncertainty, providing new methodologies for controlling uncertain dynamical systems in both engineering and biological domains.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to express my gratitude, first and foremost, to my Ph.D. advisor, Professor Mehran Mesbahi, for his unwavering support throughout my doctoral journey. Professor Mesbahi's mentorship has not only facilitated my academic achievements but has also significantly contributed to my personal growth. I would like to acknowledge and express my gratitude to my collaborators, Shahriar Talebi, Newsha Rahimi, Jonathan Becktor, Beniamino Pozzan, and Spencer Kraisler, for the amazing opportunities and fruitful teamwork we have shared. In particular, I extend special thanks to Amir Rahmani and Saptarshi Bandyopadhyay 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, Justin Chang, Tom Meisen and Ayaz Ahmed. 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. I have started to feel like something of a scientist myself.

DEDICATION

To my parents Sunanda and Avinash Deole.

Part I

VISION BASED ESTIMATION AND NAVIGATION

Chapter 1

MACHINE LEARNING FOR VISION GUIDED NAVIGATION

The first part of this thesis focuses on investigating controller and trajectory design for systems that incorporate uncertainties arising from Machine Learning (ML)-based models. Specifically, we explore techniques suitable for safety-critical environments where state estimation is derived primarily from perception maps. These maps are described as an image to state mapping generated by feeding the image captured by cameras to the ML platform that predicts the state. Our goal is to develop and demonstrate the use of these methods that ensure robust performance despite the inherent uncertainties associated with ML-driven perception. We will begin with describing the setup required for designing, training, testing and validating the perception maps. The key components; the simulation platform and the ML architecture, are described in this chapter.

1.1 Simulation Platforms

Object detection is a cornerstone task in computer vision, and is utilized in critical applications such as autonomous driving, robotics, and surveillance. These applications demand robust models capable of accurately identifying and localizing objects in diverse, real-world environments. However, developing such models requires large-scale, annotated datasets—resources that are often prohibitively expensive and time-consuming to collect.

Real-world data acquisition faces significant challenges, including high costs for sensor hardware, labor-intensive manual annotation, and limited control over environmental variables such as lighting, weather, and object placement/orientation. Furthermore, real-world datasets frequently lack diversity or fail to capture rare or edge-case scenarios—such as uncommon traffic situations or occlusions in surveillance—which are essential for ensuring

model robustness.

To address these challenges, photorealistic simulation platforms have emerged as transformative tools. By generating synthetic datasets with high-fidelity rendering, these platforms provide scalable and cost-effective alternatives to real-world data collection. They allow precise control over scene parameters, facilitating the simulation of varied conditions—such as adverse weather, complex lighting, and dynamic object interactions—without physical constraints. Moreover, these simulators often include integrated annotation tools that automatically generate ground-truth labels (e.g., bounding boxes, segmentation masks, and depth maps), significantly accelerating dataset creation and ensuring consistent annotations.

The demand for photorealistic simulations is further driven by the growing complexity of object detection tasks. Modern deep learning models, including convolutional neural networks and transformers, require vast amounts of data to generalize effectively across domains. Synthetic data from photorealistic simulators can augment real-world datasets and enhance model performance through domain randomization—where variations in textures, lighting, and object appearances bridge the sim-to-real gap. Furthermore, simulations enable safe and repeatable testing of safety-critical applications, such as autonomous vehicles, where real-world experimentation may be risky or impractical. Platforms like CARLA and AirSim (now GRID) facilitate the simulation of rare crash scenarios or sensor failures, providing valuable data to improve model reliability.

Understanding these simulation platforms is essential for designing our Unreal Engine-based simulator presented later in this thesis. Table 1.1 summarizes commonly used photorealistic simulation platforms, comparing their rendering quality, annotation capabilities, and domain-specific applicability.

This section surveys commonly used photorealistic simulation platforms in object detection research, focusing on their rendering quality and applicability to specific domains. This is a growing domain of research, and new platforms are constantly being designed and upgraded by the community.

Table 1.1 presents a list of simulation platforms used for data set generation and visual-

ization. Photorealistic simulation platforms for object detection include UnrealCV [164], a plugin for Unreal Engine 4 offering depth maps and object annotations with domain randomization to reduce the sim-to-real gap via customizable shaders; AirSim [182], a Microsoft-developed simulator for vehicles and drones with realistic physics and sensor simulations (e.g., LiDAR, cameras) to mimic real-world dynamics; Unity Perception Toolkit [201], which provides procedural generation and bounding box/segmentation labeling with domain randomization for robotics and retail; SYNTHIA [175], a Unity-based dataset for urban driving with varied weather and lighting, aiding sim-to-real transfer despite stylized rendering; BlenderProc [55], an open-source Blender pipeline for indoor scenes with texture randomization to minimize the sim-to-real gap; CARLA [63], an Unreal Engine-based simulator for autonomous driving with 2D/3D bounding boxes and LiDAR data, enhancing sim-to-real transfer through high-fidelity urban environments; and NVIDIA's Isaac Sim [133], which uses Omniverse for ray-traced rendering and physics-based simulations, ideal for industrial applications with robust domain randomization to bridge visual and physical gaps.

Table 1.1: Comparison of Photorealistic Simulation Platforms

Platform	Engine	Labels	Domain	Realism	Open Source	API Support
UnrealCV	UE4	BBox, Depth	General CV	High	Yes	Python
AirSim	UE4	BBox, Depth	Vehicles, Drones	High	Yes	Python, C++
Unity Perception	Unity	BBox, Segm.	Robotics, Retail	Medium	Yes	C#
SYNTHIA	Unity	Segm., BBox	Driving	Medium	Partial	Limited
BlenderProc	Blender	BBox, Segm.	Indoor CV	High	Yes	Python
CARLA	UE4	2D/3D BBox	Driving	High	Yes	Python
Isaac Sim	Omniverse	BBox, Point Cloud	Industrial	Very High	Partial	Python

Sim-to-Real Transfer

Simulation platforms are vital for domain adaptation in object detection, particularly in autonomous driving, where platforms such as CARLA [63] train models for pedestrian and vehicle detection using synthetic urban data. The sim-to-real gap—where synthetic data differs from real-world data in visual appearance (e.g., textures, lighting) and physical behavior

(e.g., sensor noise)—can reduce model accuracy. For example, SYNTHIA's [175] stylized rendering may limit real-world performance. Mitigation strategies include domain randomization (e.g., Unity Perception, BlenderProc), fine-tuning with real data (e.g., CARLA), hybrid datasets (e.g., AirSim), and advanced rendering techniques such as Isaac Sim's ray tracing. Despite challenges like computational costs and randomization noise, these platforms significantly enhance model robustness for real-world applications.

1.2 UE-based Simulation Platform

As part of ongoing research at the Rain Lab, we are developing a simulation engine based on the Unreal Engine 5 (UE5) platform. Our survey of existing platforms highlighted the need for a modular design tailored to our specific scenario of interest: the satellite rendezvous problem. Consequently, we chose to build a custom simulation environment from the ground up.

This visualization engine is a critical component in evaluating the performance of our vision-based estimation system. The agent's state, defined by nonlinear system dynamics, is propagated by the dynamics module and shared across the simulation network. This state information is then consumed by the Unreal Engine visualization module, which renders the scene with the agent embedded in the environment. A virtual camera captures photorealistic images as proxies for state measurements, with adjustable levels of realism.

These camera images serve as inputs to an ML-based estimator, which outputs the agent's relative state estimate. This estimate is then processed by an outlier rejection filter followed by a dynamic filter—either a Moving Horizon Filter or a Luenberger Observer, depending on the application. We interpret this entire estimation loop as a nonlinear mapping from state to output, formalized as the *perception map*:

$$h : \mathbb{R}^n \to \mathbb{R}^m,$$

as illustrated in Figure 1.1. The overall architecture of the simulation platform is depicted in Figure 1.2, while Figure 1.3 showcases example photorealistic visualizations.

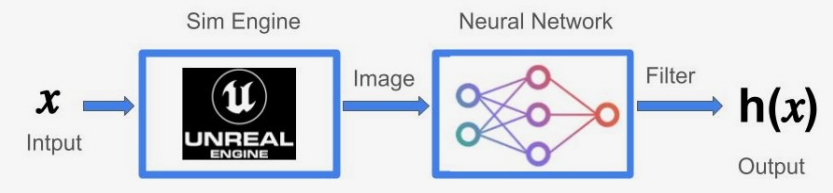


Figure 1.1: Perception map

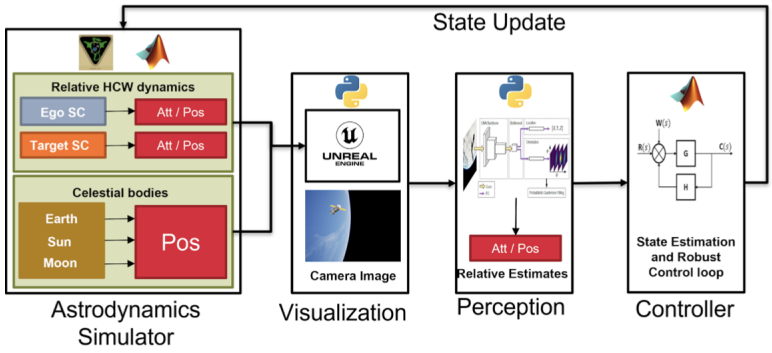


Figure 1.2: Simulation platform schematic.

We further incorporate features of realism consistent with mission scenarios and add sensor-related noise to support both the training and testing phases of the ML estimation loop. Figure 1.4 illustrates some of these visual phenomena, including bloom effects, chromatic aberration, and lens flares, which simulate realistic sensing challenges.

We focus on using features of realism consistent with the mission scenarios and add sensing-related noise for the training and testing phases of the ML loop.

1.3 Overview of Object Pose Detection Methods

Object pose detection aims to estimate the 6 degrees of freedom (6DoF) pose—rotation and translation—of objects in images. This capability is critical for tasks in robotics, autonomous driving, and augmented reality. The advent of deep learning has led to significant advances

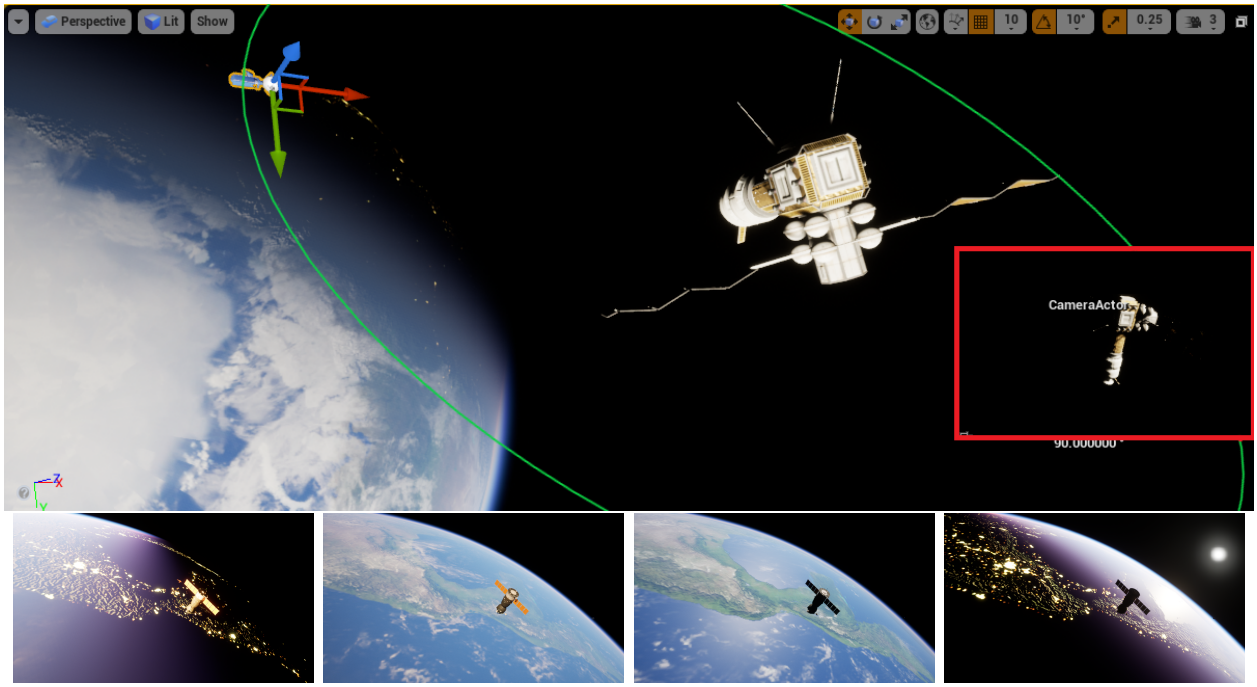


Figure 1.3: Photorealistic Simulation Engine Designed at Rain Lab

Figure 1.4: Left: Scenarios showing effects of bloom from the surface when camera is suddenly switched on. Middle: Chromatic aberration due to defective lens. Right: Flares due to light scattering

in monocular RGB-based 6DoF pose estimation.

Fan et al. [67] categorize pose detection methods into:

- **Instance-level detection:** Estimates pose of known objects using CAD models.

- **Category-level detection:** Handles unseen objects by generalizing across categories.
- **Pose tracking:** Refines pose estimates over time using video input.

Instance-level methods using RGB input can be broadly categorized into five groups:

- **Direct regression:** Predicts pose parameters directly (e.g., PoseCNN [227]), but is sensitive to occlusion.
- **Keypoint-based:** Detects 2D keypoints (e.g., PVNet [157]) and solves Perspective-n-Point (PnP) problem where camera pose is determined using positions on n points [154]. Robust to occlusion but dependent on keypoint visibility.
- **Dense correspondence:** Estimates per-pixel 2D-3D mapping (e.g., DenseFusion [223]), at high computational cost.
- **Refinement-based:** Iteratively improves initial pose estimates (e.g., DeepIM [121]).
- **Self-supervised:** Trained on synthetic data without annotations (e.g., Self6D [221]), but limited by domain gaps.

Table 1.2: Comparison of Representative RGB-Based Methods (LINEMOD)

Method	**2D Proj. (%)**	**ADD(-S) (%)**	**5°5cm (%)**
BB8 [165]	89.3	62.7	69.0
PVNet [157]	99.0	86.3	–
CDPN [122]	98.1	89.9	94.3
DeepIM [121]	97.5	88.6	85.2
Self6D [221]	–	58.9	–

Among these, keypoint-based and refinement methods are particularly robust to occlusions. In contrast, dense correspondence and self-supervised approaches show promise but still face challenges related to computational efficiency and domain generalization. Ongoing research focuses on unifying these pipelines and enhancing real-world performance.

Recently, algorithms for 6D pose estimation for objects from images have been explored in the ML community [230, 58, 120, 155, 202]. As accuracy of segmentation algorithms improves through ongoing research in ML domain, the accuracy of the pose estimators is bound to increase.

In the context of space-based object detection, a specialized class of algorithms was developed for the ESA Pose Estimation Challenge. This competition focused on satellite relative pose estimation using both real and synthetic imagery [49, 163, 183, 188, 94].

Building upon prior research, this work adopts a framework inspired by the pose and position estimation pipeline introduced in [32]. This algorithm was developed in collaboration with and at JPL for the Robust Data-driven Vision-based multi-Spacecraft Guidance and Control (RDVS) project collaboration.

This pipeline consists of three separate operations: Target segmentation, keypoint estimation, and attitude and position estimation. Mask R-CNN [97] is used for segmentation, HRNets [222] for keypoint estimation, and the perspective-n-point algorithm [79, 119, 205] for final pose estimation.

1.4 CNN-based Pose Estimation using Key-points and Perspective-n-Point

This section details a framework inspired by the pose and position estimation pipeline from [32], developed in collaboration with JPL under the Robust Data-driven Vision-based multi-Spacecraft Guidance and Control (RDVS) project. For this specific scenario we have a spacecraft (Ego) that uses a RGB camera to capture images of an uncontrolled spacecraft (Target). The ML algorithm is designed to estimate pose of this Target The pipeline comprises three main steps: (1) Target segmentation, (2) key-point estimation, and (3) attitude and position estimation using PnP.

1.4.1 Segmentation

The initial step is to segment the Target from the Ego viewpoint using Mask R-CNN [97], a mask-based segmentation network with a ResNet backbone. We adopt the PyTorch implementation. Given an input image, the network predicts a mask, bounding box, and class labels (Target or background). The predicted mask generates a region of interest (ROI) by expanding the bounding box to fully include the Target. This approach enables high-resolution inputs for subsequent stages by cropping unnecessary background.

Training is performed using the AdamW optimizer [130] with Cosine Annealing Warm Restarts [129], an initial learning rate of 1×10^{-4}, a batch size of 8, and training on an RTX-3090 GPU. The loss is defined as:

$$L = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}}, \tag{1.1}$$

following the Mask R-CNN formulation [97].

The training dataset comprises 24,000 images generated using Unreal Engine 4, simulating conditions in Low Earth Orbit (LEO) with varying sun angles, Earth/Moon/Target positions, and Target orientations. Sample segmentation data is shown in Figure 1.5.

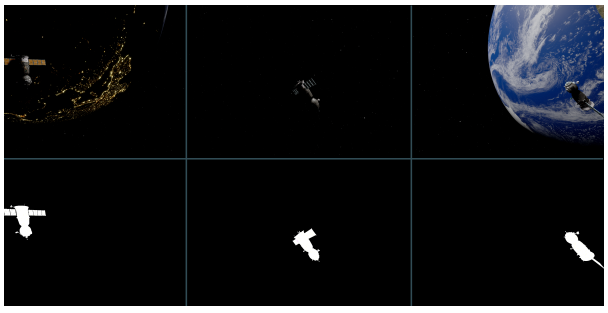


Figure 1.5: Samples of segmentation training data generated in Unreal Engine in LEO with varying sun angles and positions of Earth/Moon/Target.

1.4.2 Key-point Estimation

With the segmentation result, the image is cropped and resized to the ROI. The goal is to predict key points of the Target, such as corners of solar arrays and other distinct features, using HRNet [222].

The network is trained using AdamW [130] with Cosine Annealing Warm Restarts, an initial learning rate of 1×10^{-5}, a batch size of 16, and an RTX-3090 GPU. The loss function is the mean squared error:

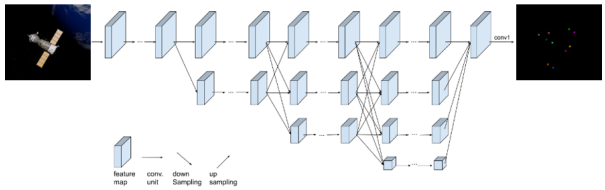
$$L_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} \big(y_i - f(x_i)\big)^2, \tag{1.2}$$

where y_i is the ground truth, and $f(x_i)$ is the predicted key-point.

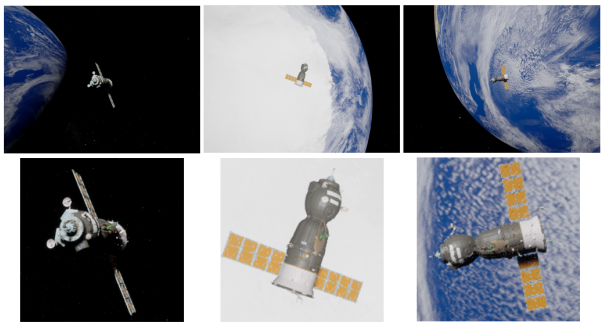
Each ground-truth keypoint is represented as a 2D Gaussian distribution centered at its true location with a standard deviation of 1 pixel. The training dataset consists of 83,000 synthetic images generated in Unreal Engine 4, simulating LEO conditions, varying sun angles, and Target/Ego positions. The relative distance between Ego and Target ranges from 15–45 meters, which affects the resolution of cropped ROIs, as shown in Figure 1.6b.

1.4.3 Perspective-n-Point

Given predicted 2D key-points and known 3D Target points, the pose is estimated using the RANSAC-based PnP algorithm from OpenCV [28], with SQPnP [205] as the solver. The algorithm discards outlier keypoints to compute an initial pose estimate, which is subsequently refined using Levenberg–Marquardt (LM) optimization. This step minimizes the re-projection error—the sum of squared distances between observed key-points and the projected object points—yielding a final estimate of the Target's position and orientation.



(a) HRNet architecture adapted from [32].



(b) Key-point training data examples showing resolution at different distances.

Figure 1.6: CNN pipeline components.

1.4.4 Pipeline Results

The position loss is defined as the ℓ_2 norm between predicted translation $\mathbf{t}_{\mathrm{p}}$ and ground truth translation $\mathbf{t}_{\mathrm{gt}}$:

$$L_{\mathrm{T}} = \sum_{i=1}^{m} \frac{\left\|\mathbf{t}_{\mathrm{p}}^{(i)} - \mathbf{t}_{\mathrm{gt}}^{(i)}\right\|}{\left\|\mathbf{t}_{\mathrm{gt}}^{(i)}\right\|}. \tag{1.3}$$

The pose loss is the angle between the predicted quaternion q_{p} and the ground truth quaternion q_{gt}:

$$L_{\mathrm{R}} = 2 \cdot \arccos\left(|q_{\mathrm{p}}^{(i)\top} q_{\mathrm{gt}}^{(i)}|\right). \tag{1.4}$$

The total loss is given by:

$$L_{\mathrm{total}} = L_{\mathrm{T}} + L_{\mathrm{R}}. \tag{1.5}$$

The test set includes samples with distances from 15–35 meters between Ego and Target and sun angles uniformly distributed over the sphere, excluding direct sun angles in the Ego field of view.

Table 1.3: Results of the total loss L_{total} on two test sets using 1 MP and 12 MP images.

Metric	1 MP	12 MP
95% CI μ_{E_T} (m)	0.037	0.020
99% CI μ_{E_T} (m)	0.064	0.038
95% CI μ_{E_R} (rad)	0.149	0.103
99% CI μ_{E_R} (rad)	0.192	0.146

As shown in Table 1.3, higher-resolution input significantly improves the pipeline's accuracy. The system yields precise position estimates and reliable attitude predictions across both test sets. However, errors occur when the key-point network mirrors key-points or fails

to detect enough points. A certainty score is computed for each key-point and aggregated to a total certainty score. If this is low, the pipeline passes this information to the controller along with the pose estimate.

Dataset Generation

Datasets are generated using Unreal Engine 4 simulations of the Target spacecraft in LEO, with the Ego spacecraft in a relative Hill's frame. The Target is positioned 10–40 meters away with varying angles (0–60 degrees). Additional exploration images uniformly sample the feasible relative states (Target position and rotation relative to Ego) in the camera's field of view. Environmental factors—especially sun angle and Earth positions—are sampled uniformly as they significantly affect pose estimation accuracy. Noise factors are added to the dataset to include camera aberrations, as illustrated in Figure 1.4.

In summary, our in-house simulation engine provides a flexible, modular, and photo-realistic platform tailored to our research needs. It supports the integration of ML-based estimators with robust testing under varied conditions, thus bridging the gap between synthetic and real-world data.

1.5 Conclusion

As of early 2025, the field of 6-DoF pose estimation is undergoing a notable shift toward model-free and reference-based methods, enabling accurate prediction of object pose and size using just a single RGB-D anchor image—most notably showcased by *Any6D* [117]. At the same time, *BOP Challenge 2024* has expanded the benchmark suite to include real-world, model-free tasks and practical object-detection scenarios, pushing for faster onboarding and stronger generalization across unseen conditions [147]. Similarly, *SuperPose* [54] integrates robust tracking, mask-free initialization, and the use of foundation models like SAM2 and LightGlue to maintain continuous real-time pose estimation—eliminating the need for manual retraining or prior segmentation.

Parallel research trends highlight the rise of distribution-aware approaches such as Corr-

2Distrib [220], which explicitly model pose ambiguity; efficiency-oriented methods like FAST GDRNPP [238] and 6DOPE-GS [235], which enable real-time deployment in robotics and AR; and foundation-guided pipelines such as SAM-6D [128] and ZS6D [196], which leverage large vision transformers for zero-shot generalization. End-to-end pipelines like YOLOPose V2 [146] further showcase the effectiveness of transformer-based architectures for multi-object settings.

Overall, the trajectory of 6D pose estimation is toward generalizable, foundation-model-driven solutions that minimize reliance on object-specific training data, achieve robustness under challenging real-world conditions, and use reliable benchmarks such as BOP as yardsticks for progress. This trend suggests that the future of pose estimation will increasingly prioritize scalability, adaptability, and seamless integration into broader perception systems.

Despite these promising advances, several limitations remain when considering 6D pose estimation for safety-critical applications such as autonomous manipulation or navigation. First, most deep learning–based methods provide no formal guarantees on correctness, stability, or worst-case performance. While approaches like Corr2Distrib introduce distributional reasoning, they still lack calibrated uncertainty estimates necessary for risk-aware decision-making. Second, robustness under domain shift remains a key challenge: models trained and benchmarked on datasets like LINEMOD, YCB-V, or BOP often degrade significantly in unseen real-world conditions involving clutter, occlusion, or lighting changes.

Another limitation concerns symmetry and object ambiguity: many methods produce multiple plausible pose hypotheses, which can lead to unsafe downstream actions if not properly resolved. Efficiency-oriented pipelines (e.g., FAST GDRNPP, 6DOPE-GS) further highlight the trade-off between speed and accuracy—real-time capability is achieved at the cost of degraded pose precision in challenging cases. Moreover, foundation-model-driven methods (e.g., SAM-6D, SuperPose) inherit biases and blind spots from their pretraining corpora, raising concerns about reliability when deployed on out-of-distribution objects.

Finally, progress in this field remains largely benchmark-driven. While the BOP Challenge [147] has pushed toward more practical and model-free tasks, current evaluation pro-

tocols rarely capture safety-critical metrics such as worst-case error bounds, robustness to adversarial conditions, or calibrated confidence estimates. For robotics safety, this implies that pose estimates must be augmented with filtering, uncertainty-aware planning, or set-based estimation before they can be reliably deployed in real-world scenarios.

In our subsequent work, we present error characterization approaches without considering the specific architecture used for the perception map. For the map to be integrated into the robust control setup, we claim that the map must follow the criterion stated in the subsequent sections. Namely, we will show the use of two specific approaches: first, a slope bound in the neighborhood of a known trajectory and second, a generalizable sector-bounded nonlinearity. These approaches are adjacent to other conventional methods used for uncertainty characterization; therefore, they are applicable to a wide variety of problems. Note that to make sure that the perception map adheres to the characterization, some external filters like outlier rejection may be required.

Chapter 2

ROBUST CONTROLLER SYNTHESIS FOR VISION-BASED NAVIGATION

This chapter presents approaches for integrating machine learning (ML)-based estimation into state feedback control. We focus on a scenario where the Target spacecraft is tracked using a pose estimation algorithm such as one introduced in the previous chapter. We first describe a method that models perception uncertainty as a state-dependent and slope-bounded set around a nominal trajectory. A funnel synthesis method is then employed to design a feedback controller as defined in by Taylor et al in [172]. By leveraging slope bounds, we characterize the set of worst-case stabilizable states as a sequence of ellipsoids around a nominal trajectory (referred to as a funnel [172]). A state feedback controller is designed such that any state within the maximal funnel can be exponentially stabilized to an invariant set around the nominal trajectory.

In a second approach, we demonstrate that a passivity-based model can be used for perception, enabling the design of a simplified passivity-based controller for trajectory tracking. This approach is similar to some works seen in recent literature where high order control Lyapunov barrier functions are used in the context of non-linear optimal control [36].

2.1 Introduction

In this section, we examine vision-guided navigation for satellite rendezvous operations. The central challenge lies in incorporating state-dependent uncertainties from the perception map and designing a controller that is robust enough to track an optimal trajectory. The chapter discusses two approaches that have been investigated for robust vision-based navigation for Ego-Target tracking scenarios. The first approach applies funnel synthesis, where the

perception map is modeled as a slope-bounded uncertainty around the nominal trajectory. In the second approach, we treat perception nonlinearities as sector-bounded nonlinearities, a type of passivity short-model, that leads to a feedback control design. We then categorize the set of feasible controllers and demonstrate simulation-based control optimization. We also show that multi-agent sensing can improve tracking performance for the Ego-Target problem.

Close proximity operations for spacecraft have been extensively studied and tested in recent years [194, 25, 65, 123]. Applications such as docking and on-orbit services heavily rely on cooperation between spacecrafts and require a high level of certainty on exchanged information (ex. relative position) to avoid catastrophic failures. Through an array of sensor measurements, these design processes rely on observer models like Extended Kalman Filter (EKF) to generate state estimations; as such, these techniques involve full or partial knowledge of Target spacecraft [72]. The Target may not cooperate in situations such as asteroid capture or orbital debris removal. In these cases, feature recognition over the Target surface has been widely adopted. The problem is then broken into long-range and short-range estimation depending on the accuracy of feature recognition algorithms [190].

Incorporating rich perceptual sensing modalities such as cameras has the potential to convey more information than simple, single-output sensor devices. Despite the maturity of deep learning algorithms and CNN in many computer vision tasks, the use of ML has only recently become common for state estimation problems in space applications [31, 184, 95, 189, 162]. These new methods utilize high-resolution images to train a neural network for Target-specific state estimation. The use of such tools can be challenging in harsh lighting conditions and against highly textured backgrounds (e.g., Earth) [78, 11]. 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 a challenging task. This becomes particularly critical when the neural network serves as a perception module within a closed-loop control system for autonomous navigation—which requires specific safety guarantees.

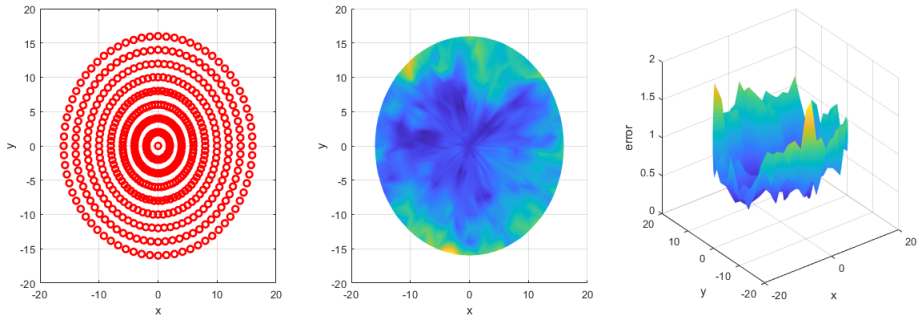


Figure 2.1: The measurement error profile (center and right) showing the maximum errors around each training data point (shown in red in the left image). This scenario captures relative state of Target with respect to camera while randomly sampling Target orientation. See [20] for more details.

In traditional deep learning, regression models typically provide only point estimates without expressing uncertainty, and classification models often output normalized score vectors—such as softmax probabilities—that fail to reflect their own uncertainty. To address this limitation, researchers in the control community are increasingly integrating machine learning with robust and optimal control techniques. This hybrid approach aims to enhance trustworthiness and reliability, especially in applications like autonomous navigation. In particular, in [50] the authors consider controlling a known linear dynamical system for which partial state information can only be extracted from complex observations. We use the approach defined in [50] to design a virtual sensor by learning both a perception map (i.e., a map from (imagery) observations to a linear function of the state) and a bound on its estimation errors. Under suitable smoothness assumptions (Lemma 1 (Closeness implies generalization) [51]), "uniformly" bounded errors are guaranteed within a neighborhood of the training data. This uncertainty model enables the synthesis of a robust controller that keeps the system's state trajectories within the bounds encountered during training. However, in space applications, the information retrievable from imagery can vary dramatically depending on the system's relative pose—for instance, as shown in a real-world example

Figure 2.1.

Therefore, a uniform analysis results in excessively conservative error modelings. Moreover, the error profile typically does not follow a known distribution pattern and thus, assuming distribution profiles (such as Gaussian) would not be realistic. Here, instead, we aim to provide an error modeling approach that can capture its dependence on the relative states of the system and extract guaranteed upper-bounds that are suitable for employment in the imminent robust control synthesis.

2.2 Funnel Synthesis for Slope-Bounded Perception Based Uncertainty

Explicit trajectory generation methods suffer from two fundamental drawbacks. First, they are inherently specific to the problem data, and any change requires re-solving the optimal control problem from scratch. Second, these methods lack convergence guarantees for solving the associated nonconvex optimal control problems. No known algorithm can guarantee convergence to a solution for a general nonconvex optimal control problem from an arbitrary initial guess. As a result, there is no theoretical justification for solving such problems in real-time—although empirical evidence suggests that reliable local convergence is often achieved in practice [172]. This motivates the study of alternative methods that can theoretically guarantee real-time feasibility.

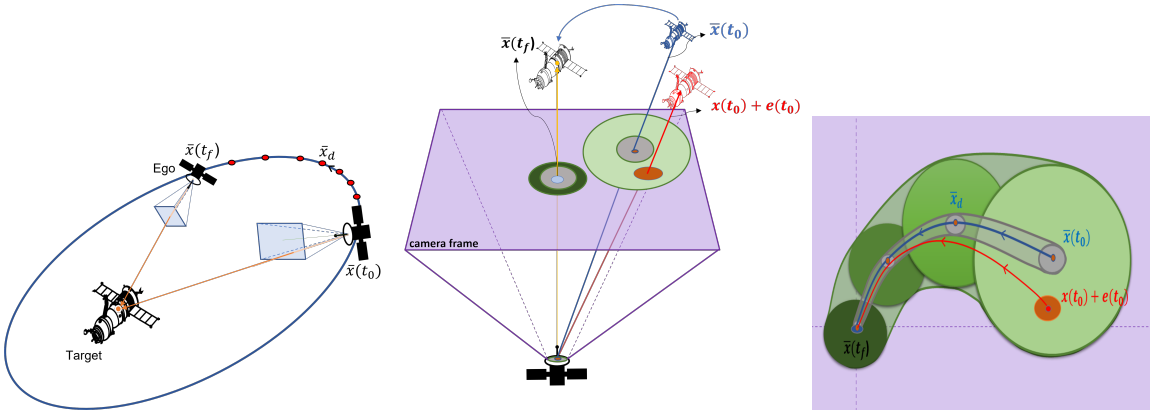
In this work, we aim to combine contemporary techniques from machine learning and robust control to design a nonlinear controller that synthesizes feedback signals directly from image sensors. We illustrate this approach with the following scenario: the Ego spacecraft must navigate to a desired position and orientation relative to an uncooperative Target spacecraft, while ensuring that the Target remains within its camera's field of view (see Figure 2.2a). This task requires estimating the Target's relative state in the Ego's body frame from a sequence of image observations. Our method extends the analysis of [50] by introducing a virtual sensor that jointly learns (i) a perception map—that is, a mapping from raw observations to a linear function of the Target state—and (ii) a state-dependent bound on the associated estimation error.

In contrast to [50], in our setup, factors such as the distance to the Target and the sun's position significantly affect the quality of information extractable from imagery. We therefore model the perception error as a function of the relative state between the Ego and Target. This approach has been described in Section 2.3.2.

Next, we utilize implicit trajectory optimization methods to define a feasible set of trajectories in the state-control space. Then, based on a nominal trajectory and the concept of uniform ultimate boundedness (UUB) [109], we extend the invariant funnel synthesis technique [172] to account for state-dependent measurement noise. In this chapter, the estimator is assumed to have no information about the state-dependent characteristics of noise. Therefore, the estimator is assumed to be a Kalman Filter designed for worst-case noise covariance. In the next chapter, we will discuss estimation-aware planning where the performance of generic estimators can be improved. On the contrary, if full prior knowledge of state-dependence noise characteristics is available, then adaptive filtering techniques can be employed [186]. This framework enables incorporating the perception error model and its associated upper bounds directly into the controller design.

The result is a composite robust control strategy that generates control inputs directly from visual observations, while handling uncertainties in both the nonlinear dynamics and the noisy measurements.

Figure 2.2 illustrates the orbital trajectory of the Ego spacecraft around the Target in Hill's frame, shown in (a). The Ego's camera frame (purple plane) is shown in (b), which also depicts the nominal trajectory starting from $\bar{x}(t_0)$ (blue) with the Target initially off-center in the camera view, and ending at the terminal state $\bar{x}(t_f)$ (black), where the Target is aligned with the center of the Ego's camera. The actual initial state $x(t_0)$ (red) lies within the invariant funnel at $t = 0$ (light green ellipse). Finally, (c) shows both the nominal (blue) and actual (red) trajectories contained within the invariant funnel (green ellipses).



(a) (b) (c)

Figure 2.2: Representation of Ego-Target tracking and funnel synthesis.

2.3 *Robust Controller Synthesis*

We will now go through system definition and controller design in the following sections starting with the definitions for modeling the perception uncertainties.

2.3.1 Structured Nonlinear System Model with State Dependent Perception Error

Considering the perception-based uncertainty for the following system. Define a (relative) state $x(t) \in \mathbb{R}^{n_x}$ (between Ego and Target) as

$$x(t) := x_{\text{Ego}}(t) - x_{\text{Target}}(t); \tag{2.1}$$

which follows nonlinear dynamics given by

$$\dot{x}(t) = f(x(t), u(t)), \quad t \in [t_0, t_f]\,, \tag{2.2}$$

$$z(t) = \mathcal{Q}(x(t)), \tag{2.3}$$

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 perception map measures the Target 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 Eq. (2.3). Suppose there exists a *perception map* $p : \mathbb{R}^m \to \mathbb{R}^l$ that imperfectly predicts the relative state $x(t)$, that is

$$p(z(t)) = Cx(t) + e_p(x(t)), \tag{2.4}$$

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 are further discussed in the succeeding section.

Assume that for the tracking task a nominal trajectory is given by applying the input $\bar{u}(t)$. Then, for online tracking using the perception map, an input policy defined by $\xi(t)$, is synthesized, that deals with both dynamics and estimation uncertainties rooted in the modeling and composed perception process, respectively. Let $\{\bar{x}(t), \bar{u}(t)\}_{t=t_0}^{t_f}$ be the nominal trajectory that satisfies the dynamics Eq. (2.2) for some initial condition, x_0, and define

$$\eta(t) := x(t) - \bar{x}(t) \quad \text{and} \quad \xi(t) := u(t) - \bar{u}(t). \tag{2.5}$$

By assuming that f is at least once differentiable, rewrite Eq. (2.2) in terms of the variables in Eq. (2.5) by using a first-order Taylor series expansion around the nominal trajectory as follows:

$$\dot{\eta}(t) = A(t)\eta(t) + B(t)\xi(t) + g(x(t), u(t)), \tag{2.6}$$

where $A(t)$ and $B(t)$ are the partial derivatives of f evaluated along the nominal trajectory, and $g(x(t), u(t))$ represents the higher order (nonlinear) terms[1]. The dynamical system in

[1] An alternative dynamics modeling and online control purely based on data will be considered as a future direction, see e.g., [198, 176]

Eq. (2.6) can then be equivalently expressed using structured nonlinearities as follows

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

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 [173], 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 Eq. (2.4) as

$$y(t) \coloneqq p(z(t)) - C\bar{x}(t) = C\eta(t) + e(\eta(t)), \tag{2.7}$$

where $e(\eta(t))$ denotes the state-dependent measurement noise—to be characterized later in Section 2.3.2. Hence, we arrive at the following (nonlinear) system dynamics and (state-dependent) noisy observations:

$$\begin{aligned}
\dot{\eta}(t) &= A(t)\eta(t) + B(t)\xi(t) + \sum_{i=1}^{N_w} E_i w_i(t) \\
h_i(t) &= F_i\eta(t) + G_i\xi(t), && i = 1, \ldots, N_w \\
w_i(t) &= \phi_i\left(h_i(t)\right), && i = 1, \ldots, N_w, \\
y(t) &= C\eta(t) + e(\eta(t)).
\end{aligned} \tag{2.8}$$

2.3.2 Modeling Sensor Uncertainties

Suppose the initial training dataset is $\mathcal{S}_0 = \{(x_d, z_d)\}_{d=1}^{N_0}$ which is 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 a wide variety of traditional supervised methods [20].

Now we perform local uncertainty analysis on optimal trajectories. Consider the assumption that defines the slope bound model that must hold for subsequent analysis. We adopt this setup from Dean et al [51], while adding state dependence w.r.t. the trajectory states.

Assumption 2.3.1. *The mapping $p \circ \mathcal{Q} - C$ is locally s-slope bounded with a radius of r around each training data point, i.e., for each $(x_d, z_d) \in \mathcal{S}_0$, there exists a bounded real number $s = s(x_d)$ such that*

$$\|p \circ \mathcal{Q}(x) - Cx - (p \circ \mathcal{Q}(x_d) - Cx_d)\| \leq s(x_d) \|(x - x_d)\|, \quad \forall x \in B_r(x_d). \tag{2.9}$$

Assumption 2.3.2. *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 defined around a nominal trajectory and the model is dependent on trajectory state x, which avoids 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

$$\mathcal{X}_r(\bar{x}(t)) \coloneqq \Big\{ x \in B_r(\overline{x}(t)) \;\Big|\; \|p(\widetilde{z}_d) - C\widetilde{x}_d\| + s(t) \|\overline{x}(t) - \widetilde{x}_d(t)\| \leq l(t), \\ \widetilde{x}_d(t) \coloneqq \arg\min_{x_d \in \mathcal{S}_0} \|x - x_d\| \Big\}, \tag{2.10}$$

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

$$s(t) \coloneqq s(\widetilde{x}_d(t)), \quad l(t) \coloneqq rs(t) + \gamma(t). \tag{2.11}$$

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 [20] for more details). Finally, we can precisely define how densely the nominal trajectory should be sampled for the training dataset as follows:

Definition 2.3.1. *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 $x \in B_r(\overline{x}(t))$ 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 [50] 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 2.3.1 and 2.3.2.

Now, we are well-equipped to characterize the perception error—altering the output of the system as modeled in Eq. (2.7)—in the following lemma.

Lemma 2.3.3. *Suppose the training dataset is sampled densely enough from the nominal trajectory (per Definition 2.3.1) with a radius r and Assumptions 2.3.1 and 2.3.2 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 Eq. (2.8) 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 a smaller shift $l(t)$. However, the slope-bound and training errors are inherent to the learning problem at hand and do not necessarily improve by just sampling more densely. Also note that for simplicity, we assume $C = I$ to represent an unbiased observable system and a state feedback controller. Therefore in subsequent analysis we use $y = \eta + e(\eta)$.

2.3.3 Robust Controller Synthesis

Now this error model can be incorporated into the funnel synthesis setup, namely Lemma 2.3.3. Consider the class of controllers that, at each time, t, can synthesize a feedback correction signal $\xi(t) = K(t)y(t)$ for some matrix-valued function of time $K(t) \in \mathbb{R}^{n_u \times n_y}$ where $n_y = n_u$.

The closed-loop system representation obtains the following form:

$$\begin{aligned}
\dot{\eta}(t) &= \left(A(t) + B(t)K(t)\right)\eta(t) + B(t)K(t)e\left(\eta(t)\right) + \sum_{i=1}^{N_w} E_i w_i(t) \\
h_i(t) &= \left(F_i + G_i K(t)\right)\eta(t) + G_i K(t) e\left(\eta(t)\right), && i = 1, \cdots, N_w \\
w_i(t) &= \phi_i\left(h_i(t)\right), && i = 1, \cdots, N_w, \\
\|e(\eta(t))\| &\leq s(t)\|\eta(t)\| + l(t).
\end{aligned} \tag{2.12}$$

As discussed in the previous section, the perception error is bounded in the vicinity of the nominal trajectory. We use this fact to model the term $K(t)e(\eta(t))$ in the dynamics of $\eta(t)$ as a structured nonlinearity as proposed by [173]. The funnel design can handle any possible perception error $e(t)$ that satisfies the perception upper bound model.

Modeling the Perception Error in Feedback as Added Nonlinearity

The slope-bounded uncertainty is formalized as an LMI, synthesizing the feedback gain K. In particular, as proposed by [173] the error, $e(t)$, is modeled as a nonlinear channel and the maps to all uncertainties as:

$$w_e(t) \coloneqq \widetilde{\phi}\left(e(t)\right) = K(t)\, e(\eta(t)), \quad \text{for some} \quad \widetilde{\phi} : \mathbb{R}^{n_y} \to \mathbb{R}^{n_u}. \tag{2.13}$$

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

$$h_i(t) = \left(F_i + G_i K(t)\right)\eta(t) + \widetilde{G}_i e(t), \qquad i = 1, \cdots, N_w \tag{2.14}$$

where the constant matrix $\widetilde{G}_i \in \mathbb{R}^{n_x \times n_y}$ serves as a 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 Eq. (2.12)

together with Eq. (2.13) and Eq. (2.14) as

$$\begin{aligned}\dot{\eta}(t) &= A_{cl}(t)\eta(t) + B(t)w_e(t) + Ew(t),\\ h(t) &= F_{cl}(t)\eta(t) + \widetilde{G}e(t),\\ w(t) &= \phi(h(t)),\\ w_e(t) &= \widetilde{\phi}\left(e(t)\right)\end{aligned} \tag{2.15}$$

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

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

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 $A_{\text{cl}}(t) := A(t) + B(t)K(t)$ and $F_{\text{cl}}(t) := F + GK(t)$.

Funnel Synthesis

The funnel synthesis techniques are based on the notion of quadratic stability as defined in [177] and [5, 18, 48], the latter of which offers necessary and sufficient conditions for stability based on quadratic Lyapunov functions. As described in [172], funnel synthesis makes use of a Lyapunov function (as opposed to the maximum principle-based techniques) to seek out nearby feasible trajectories. We restate the funnel definition from [172] as

Definition 2.3.2. *(Funnel). A funnel, denoted by $\mathcal{F}(t)$, is a time-varying set in state and control space that is both invariant and contained inside a feasible region.*

Specifically, the term funnel synthesis refers to the algorithmic procedure designed to compute a funnel. The invariance property of a funnel means that if a particular initial condition is inside the entry of the funnel (at some initial time t_0), then the entire subsequent trajectory remains inside the funnel as well. Stated mathematically, if $\left(\eta\left(t_0\right), \xi\left(t_0\right)\right) \in \mathcal{F}(t_0)$ then $(\eta(t), \xi(t)) \in \mathcal{F}(t)$ for all $t \geq t_0$. 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. Funnel synthesis guarantees convergence behavior. For real-time implementation, the funnel based controllers are deployed as state-dependent controllers which can be stored on hardware.

Particularly, consider the scalar-valued function $V : \mathbb{R}^{n_x} \to \mathbb{R}$ defined by

$$V(\eta(t)) = \eta(t)^\top Q(t)^{-1}\eta(t) \tag{2.16}$$

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 $\eta^\top Q^{-1}\eta \leq 1$, which is also the equation of a non-degenerate n_x-dimensional ellipsoid (nonzero, finite dimensions in all directions). We denote the ellipsoid defined by the positive definite matrix Q and centered at the origin as

$$\mathcal{E}_Q \coloneqq \left\{\eta \in \mathbb{R}^{n_x} \mid \eta^\top Q^{-1}\eta \leq 1\right\} = \left\{Q^{1/2}v \mid \|v\|_2 \leq 1\right\} \tag{2.17}$$

where $Q^{1/2}$ is the matrix square root that can be computed using Cholesky decomposition.

The assumption that $\xi = Ky$ used to form the closed-loop system Eq. (2.15) thus results in the following implication:

$$\text{if} \quad y \in \mathcal{E}_{Q_y} \quad \Rightarrow \quad \xi \in \mathcal{E}_{KQ_yK^\top}, \tag{2.18}$$

a fact that can be proven easily via Schur complements when K is full row-rank [173]. Before we proceed, we need to derive an upper bound for Q_y, given the information about the perception error. This bound is restated here as described by Rahimi et al in [168].

Theorem 2.3.4. *[168] 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*

$$Q_y = (1 + s\,(1 + \rho))^2\; Q, \qquad Q_s := s^2 Q, \quad \rho := \sqrt{l\lambda_{\max}\,(Q_s^{-1})} \tag{2.19}$$

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

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 as follows.

Definition 2.3.3. *[172] (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*

$$\mathcal{F} \coloneqq \mathcal{E}_Q \times \mathcal{E}_{KQ_yK^\top} \subseteq \mathcal{X}_\mathcal{F} \times \mathcal{U}_\mathcal{F}. \tag{2.20}$$

As the closed-loop system— described in Eq. (2.15)—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[2] closed-loop system. We utilized Theorem 4.18 in [109] to design a controller K such that the closed loop system Eq. (2.15), 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 Eq. (2.15) is detailed as follows:

$$\alpha_1(\|\eta(t)\|) \leq V(\eta(t)) \leq \alpha_2(\|\eta(t)\|), \tag{2.21}$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial \eta}\dot{\eta}(t) \leq -V(\eta(t)), \qquad \forall\, t \in [t_0, t_f], \qquad \forall \eta \ni \|C_\mu \eta\| \geq \mu \geq 0, \tag{2.22}$$

$$\forall h \in \Omega, \quad \text{and} \quad \forall e \in \mathcal{E}_{Q_e} \ni \|e(t)\| \leq s(t)\|\eta(t)\| + l(t),$$

[2]See Definition 4.6 in [109] for more details on UUB concept.

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 is the input to the nonlinear terms in system Eq. (2.12), must satisfy the set inclusion:

$$h \in \Omega, \quad \Omega = \mathcal{E}_{F_{cl} Q F_{cl}^\top} \oplus \mathcal{E}_{\widetilde{G} Q_e \widetilde{G}^\top} \tag{2.23}$$

where $\oplus$ is the Minkowski sum between two sets. In order to utilize the condition Eq. (2.22) 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 Eq. (2.15)—in such a way that they are consistent with the quadratic form of the lyapunov function $V(\eta(t))$. To this end, we use the idea of local multiplier matrices, as adopted in [172].

Definition 2.3.4. *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 domain of sets Ω if*

$$\mathcal{T}(h) := \begin{bmatrix} h \\ \phi(h) \end{bmatrix}^\top M \begin{bmatrix} h \\ \phi(h) \end{bmatrix} \geq 0, \quad \forall\, h \in \Omega. \tag{2.24}$$

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

$$\mathcal{M}_{\phi,\Omega} := \left\{ M \in \mathbb{S}^{(n_h+n_w)} \mid \mathcal{T}(h) \geq 0, \quad \forall\, h \in \Omega \right\}. \tag{2.25}$$

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 as

$$\mathcal{T}_e(e) := \begin{bmatrix} e \\ \widetilde{\phi}(e) \end{bmatrix}^\top M_e \begin{bmatrix} e \\ \widetilde{\phi}(e) \end{bmatrix} \geq 0, \quad \forall\, e \in \mathcal{E}_{Q_e}. \tag{2.26}$$

The stability condition is now posed as condition (2.22) such that (2.24) and (2.26) hold for $M \in \mathcal{M}_{\phi,\Omega}$ and $M_e \in \mathcal{M}_{\widetilde{\phi},\mathcal{E}_{Q_e}}$.

The subsequent procedure to check for feasibility of the funnels and the detailed synthesis of funnels and the controller is described in [173, 167].

Solution Strategy for Funnel Synthesis Problem

The solution strategy has been detailed in [172, 167] where an iterative method, called the γ-iteration has been described, to solve the quadratic funnel synthesis problem introduced earlier. Since the full problem is nonlinear and nonconvex, a direct solution is intractable. The γ-iteration alternates between two subproblems:

- **The M-problem**: With the funnel parameters fixed (matrices Q, Y, and scalar λ), this subproblem computes a local Lipschitz constant, γ, and constructs a corresponding local multiplier matrix, M_γ.

- **The Q-problem**: With M_γ fixed, this subproblem solves a convex optimization problem over Q, Y, and λ to maximize the funnel size while satisfying the system's dynamic and constraint feasibility conditions.

A contraction mechanism is introduced to shrink the maximum allowable funnel size, $Q_{\max}$ when convergence is not achieved, guided by a fill ratio, κ, that measures how well the current funnel fills the allowable space. The algorithm terminates when κ exceeds a specified threshold.

This procedure for funnel synthesis methods described here has been detailed in **Algorithm 1** [167], a convergent method that guarantees a feasible quadratic funnel in a finite number of iterations under reasonable assumptions.

2.4 Funnel synthesis for Target Tracking Problem

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, we adopt a computer vision approach to predict the Target's relative position. We use a high-fidelity astrodynamic simulator [108] along with a photo-realistic simulator to replicate the actual system. This environment emulates the space conditions in

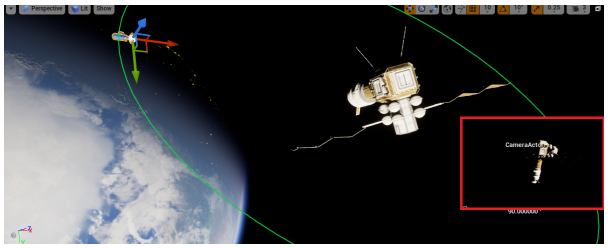


Figure 2.3: Simulation platform showing the Ego (denoted on the left with its attached body frame) orbiting around an uncontrolled satellite Target (the white object in the middle) on a fixed flyby orbit (denoted by a green circle). The in-picture on the right shows the Target in the Ego's camera frame.

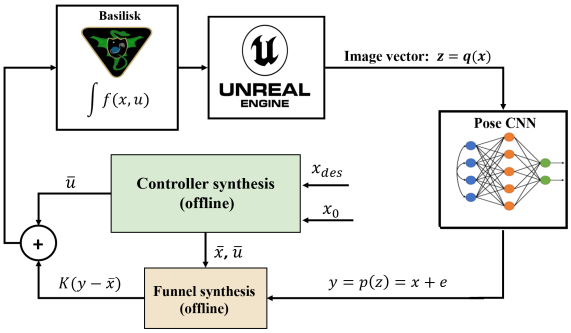


Figure 2.4: The simulation setup developed for emulating the real-world settings for applications of the proposed machinery. The astrodynamics simulator [108] and the developed visual simulator are replicating the actual system with the possibility of generating high-quality image data. The state estimated from the image data using the CNN module is then fed into the controller module for robust feedback signal synthesis.

LEO. A sample snapshot of the environment is illustrated in Figure 2.3, and the developed pipeline is showcased in Figure 2.4.

The details of the CNN-based perception module and the overall space flight simulation setup can be found in [20].

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(MRP) 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:

$$J\dot{\boldsymbol{\omega}}^E = \left(J\boldsymbol{\omega}^E\right) \times \boldsymbol{\omega}^E + \boldsymbol{u}^E + \boldsymbol{d}_{\text{ext}}\,, \tag{2.27}$$

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 MRP, $\boldsymbol{q}^E = [q_1 \;\; q_2 \;\; q_3]^\top \in \mathbb{R}^3$ 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:

$$\frac{d}{dt}\begin{bmatrix} \boldsymbol{q}^E \\ \boldsymbol{\omega}^E \end{bmatrix} = f(\boldsymbol{q}^E, \boldsymbol{\omega}^E, \boldsymbol{u}) \;=\; \begin{bmatrix} \frac{1}{2}\left(I\left(\frac{1-\left(\boldsymbol{q}^E\right)^T\boldsymbol{q}^E}{2}\right) + \boldsymbol{q}^E\left(\boldsymbol{q}^E\right)^T + S(\boldsymbol{q}^E)\right)\boldsymbol{\omega}^E \\ J^{-1}\left(\left(J\boldsymbol{\omega}^E\right) \times \boldsymbol{\omega}^E + \boldsymbol{u}\right) \end{bmatrix} \tag{2.28}$$

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 [39] 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 $r = [r_x,\; r_y,\; r_z]^\top \in \mathbb{R}^3$. Following the Hill-Clohessy-Wiltshire equations in phase magnitude form we have

$$\ddot{r}_x = \rho_x \sin(nt + a_x)\,, \quad \ddot{r}_y = \rho_y + 2\rho_x \cos(nt + a_x)\,, \quad \ddot{r}_z = \rho_z \sin(nt + a_x) \tag{2.29}$$

where $n = \sqrt{\mu/a^3}$, a refers to the orbital radius of the Target and μ is the standard gravitational parameter. In Eq. (2.29), (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 relative elliptical orbit have been detailed in [10].

For this problem, denote the relative states between Target and Ego spacecraft as $x(t) := \left[r,\; \dot{r},\; q^E,\; w^E\right]^\top \in \mathbb{R}^{12}$. We disregard the Target's orientation, as it is not needed for the

current problem. The relative dynamics are described using Eq. (2.29) and desired attitude is calculated using difference between relative Target position and Ego camera axis. This desired attitude is used as the reference state for the attitude dynamics defined by Eq. (2.28). 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 outward along positive x-axis. For simplicity of presentation, we assume the Ego's body frame and its camera frame are identical.

Along with the relative dynamics, we consider the sensory observation $z(t) \in \mathbb{R}^{m\times m}$ as defined in Eq. 2.3 which represents the high dimensional image and the perception map p as defined in Eq 2.4 that maps the image to relative state measurement $x(t)$ with perception uncertainty $e_p(x(t))$.

To model the perception map and study its error characteristics, as discussed in Section 2.3.2, 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 2.5, 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[3].

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 u that tracks the Target position in the Ego's camera frame. Hence, starting off from the nominal trajectory with error, we define

$$\eta(t) := C\ x(t) - C\ \overline{x}(t) = \begin{bmatrix} q^E(t) \\ w^E(t) \end{bmatrix} - \begin{bmatrix} \overline{q^E}(t) \\ \overline{w^E}(t) \end{bmatrix}$$

as the tracking error and $\xi = u - \bar{u}$ as the correction signal. Figure 2.6 showcases the error around the nominal trajectory with respect to time and η.

The matrices A and B are the Jacobian matrices of the attitude dynamics in Eq. (2.28) along the nominal trajectory, and the parameters F, G and E are constructed using a total

[3]See [20] for further details regarding the neural network setup.

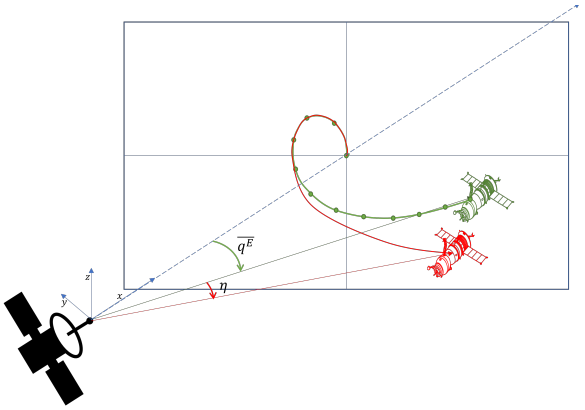


Figure 2.5: Nominal trajectory in green as seen in the camera frame. For some initial condition which deviates form the nominal as shown in red, we use the robust controller such that the error between the new state and nominal given by η is driven to zero.

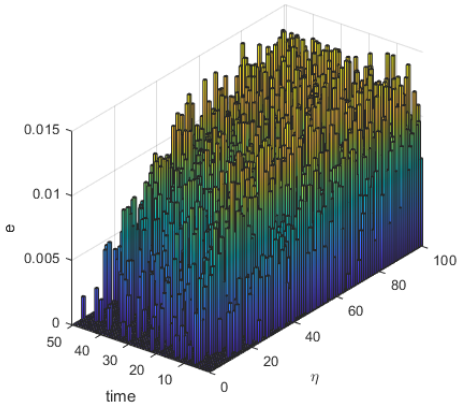
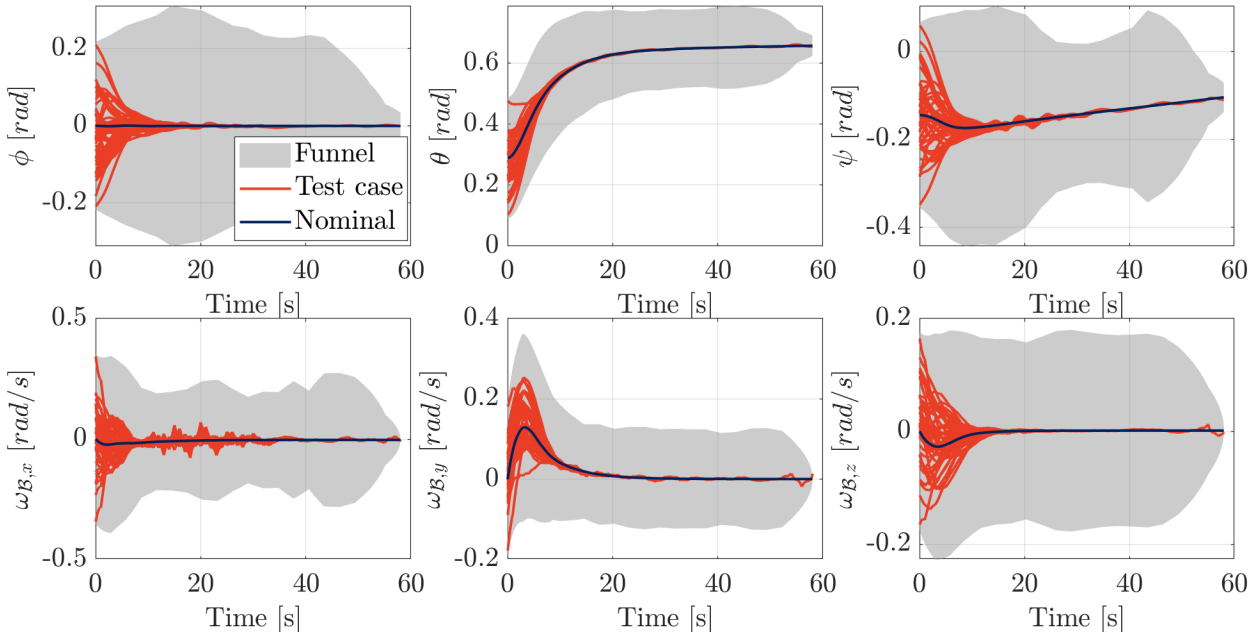


Figure 2.6: The error profile in the neighborhood of the nominal trajectory for $T = 50$ seconds (as shown with time axis) with 100 deviation samples per time stamp.

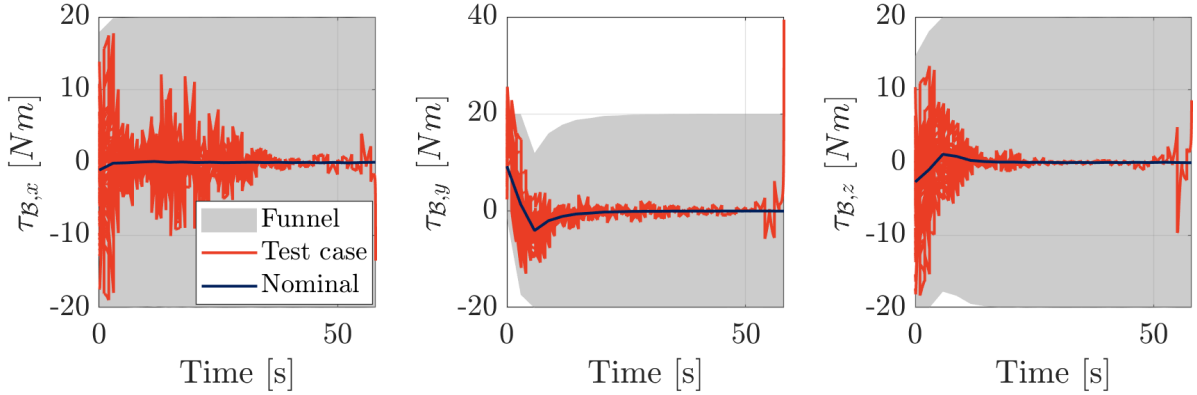
of four ($n_h = 4$) nonlinear channels to be

$$F = \begin{bmatrix} I_3 & 0_{3\times3} \\ I_3 & I_3 \\ I_3 & I_3 \\ I_3 & I_3 \end{bmatrix}, \qquad G = 0_{12\times6}, \qquad E = \begin{bmatrix} I_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_3 & I_3 & I_3 \end{bmatrix}.$$

Note that for this problem $n_x = 6$, $n_u = 3$ and $J = \mathbf{diag}\{900,\ 900,\ 900\}\ \ kg\,m^2$. The Figure 2.7 depicts the computed quadratic funnel. In Figure 2.7a, 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 were used to numerically integrate the equations of motion. The Figure 2.7b 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 2.7: 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.

In this section, we presented the funnel synthesis approach for designing a feedback controller such that a set around a given nominal trajectory remained invariant. This approach

can be used in a wide variety of applications in robotics and navigation where uncertainty quantification of sensors is non-trivial. We use a sensor model that exhibits slope-bounded behavior in the neighborhood of a nominal trajectory. In the next sections will discuss the alternative passivity-based approach to capture perception uncertainty.

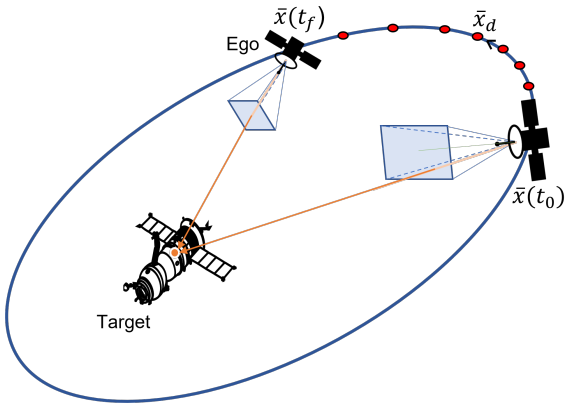


Figure 2.8: Tracking a Target satellite in a passive relative orbit. The Ego spacecraft follows a trajectory that brings the Target into the center of its camera frame while cruising in the passive orbit.

2.5 Passivity-Based Control for Vision-Guided Navigation

In this section, we present an alternative approach for attitude tracking using ML based perception. We do not condition the controller to a nominal trajectory, but instead find a general controller for attitude tracking that can adapt to a nonlinear output. Here we present:

- A passivity-based controller that operates with uncertain observations and guarantees a passivity-short stable system.
- A method for designing an optimal control law under passivity constraints using a quadratic cost and zeroth-order optimization via simulation.
- An extension of the controller to incorporate information from neighboring, generally passive, agents.

Target tracking problems in this setting have been extensively studied through various approaches [185]. Robust control based on Lyapunov theory has been well explored [145],

particularly using attitude-specific Lyapunov functions to derive stability conditions. Extensions of this approach include passivity-based methods, which we examine in this work. Passivity facilitates interconnection among subsystems and enables analysis of the overall system's stability [208, 210, 207, 66].

Using the passivity framework, we address nonlinear uncertain observations produced by the perception map. Traditionally, passivity-based control design under noise or uncertainty leads to high-gain controllers [118]. Tsiotras has studied stability and optimality under this framework, obtaining optimal solutions through Hamilton-Jacobi theory and a modified control law with quadratic cost [209]. In contrast, our approach considers a constrained problem, and we derive linear controllers for the guidance loop. Given a quadratic cost in both state and input, we optimize over a feasible set of control gains.

In satellite attitude control, the Ego spacecraft is described by two independent dynamical systems that are both passive and lossless. Our use of passivity is motivated by the fact that feedback interconnections of passive systems—under appropriate conditions such as observability and strict passivity—result in a passive and stable overall system [109]. We build upon this property to interconnect multiple passive systems and propose a consensus-based process to improve state estimation in distributed architectures.

Another motivation for using passivity is that a generalized notion of passivity can be used to describe the sensor characteristics. Our Ego spacecraft uses neural networks that take images as input and output the relative location of the Target spacecraft in the camera frame. A detailed overview of the simulation setup and neural network architecture is provided in [20], along with an analysis of environment-specific sensor behavior. Through extensive analysis of the CNN's input-output characteristics, we observe that object pose measurements exhibit state-dependent errors and are influenced by environmental factors such as illumination and celestial background.

Here, instead of defining the error profile around a trajectory, we propose characterizing the neural network's generalized error model via passivity. This model can be used independently of a nominal trajectory. Intuitively, we observe that in the absence of adversarial

noise or extreme outliers, the measured direction of the Target position aligns approximately with the true Target position. This directionality suggests that the error induced by the perception map—i.e., the energy injected—is bounded. In an ideal case, where no error is present, the perception map would be lossless.

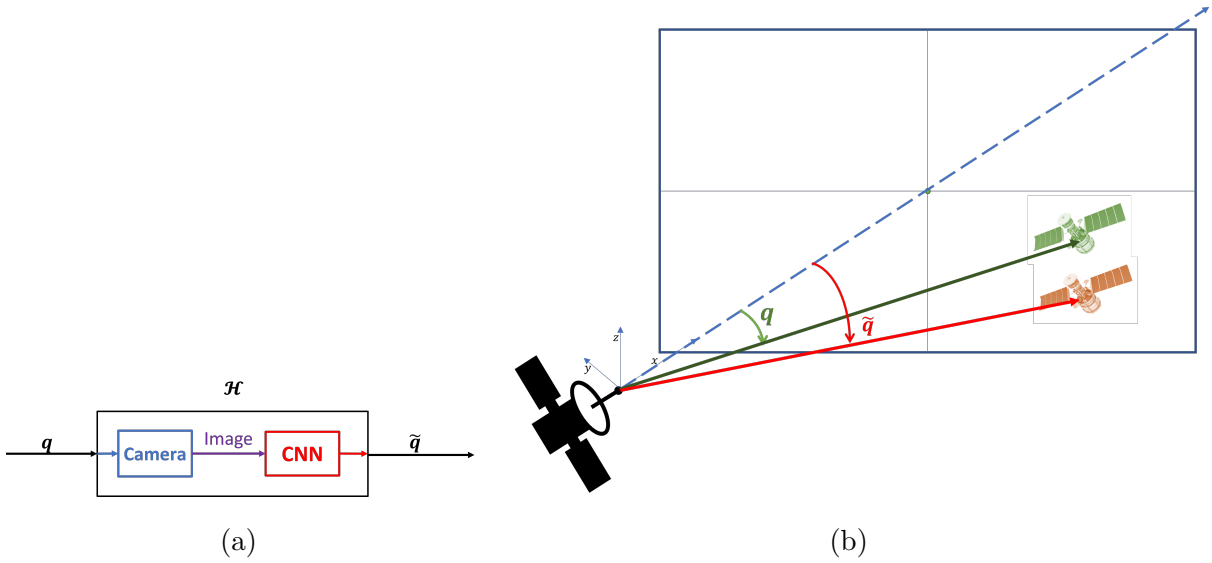
To validate this behavior, we analyze the perception map's feasible state space. Empirically, our simulations show that the output is sector-bounded along the input direction, outside an ellipse around the origin. Thus, the input-output relationship demonstrates a generalized passivity-short nature. We formalize these passivity conditions and error bounds from the CNN's empirical analysis and exploit them in control design.

Once we establish a feedback system with the perception map, the entire system can be made passive by selecting appropriate input-output pairs and a suitable storage function. For a multi-agent setup, we associate each Ego spacecraft (in passive orbit around the Target) with a passive system. These passive systems can then operate in a distributed fashion: each Ego spacecraft shares its relative attitude measurements with a candidate Ego in its own frame. This exchange enables the candidate to converge to a consensus estimate, improving accuracy over what would be achieved using a single Ego's measurements. Since consensus generally improves estimation, it allows tighter error bounds and enables more aggressive control for the distributed case.

In the following sections, we present the problem formulation, the simulation framework, and the setup for multi-agent state estimation.

2.6 Passivity-Based Error Modeling Setup

We first describe the error modeling for the perception map (pose CNN) by analyzing its input-output characteristics. To define the perception map, $\mathcal{H}$, we consider the relative dynamics between the Ego and Target spacecraft. Given a relative Target position, $x_t \in \mathbb{R}^3$, in the Ego's camera frame, we define a desired Ego attitude, $q \in \mathbb{R}^3$ (represented by Modified Rodrigues Parameters (MRPs)), that aligns the Target with the center of the camera frame.

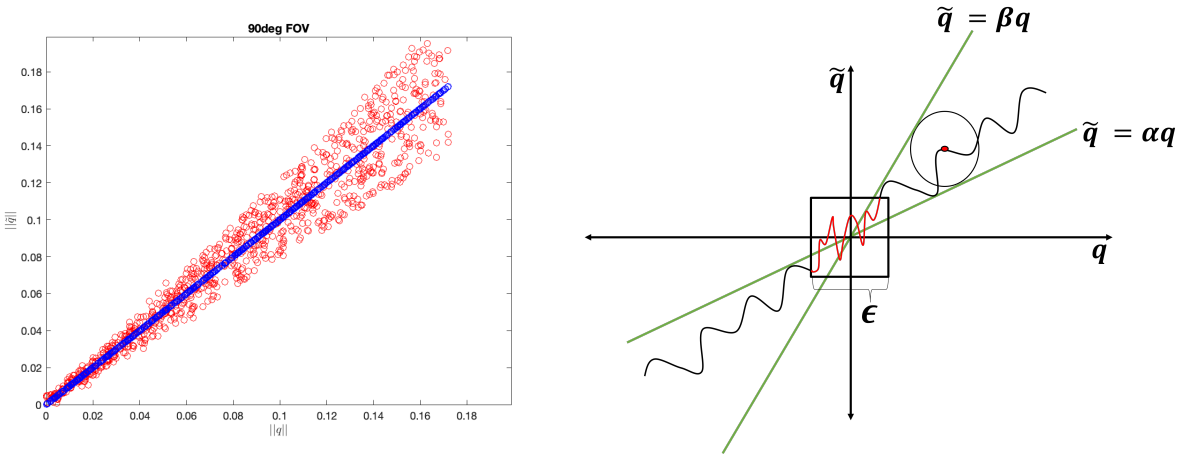


(a) (b)

Figure 2.9: (a) The perception map maps the true relative attitude q to its estimated measurement, $\tilde{q}$. (b) Depiction of the true Target position in the Ego camera frame (green). Here, q represents the relative attitude required to align the Target with the camera center. The measured position is shown in red, and the corresponding relative attitude estimate is $\tilde{q}$.

Throughout this section, the relative attitude, q, represents the tracking error between the Target's position in the Ego's camera frame and the camera center. When $q = 0$, the Target is perfectly aligned with the center, as illustrated in Fig. 2.9b. We define the nonlinear attitude dynamics in terms of $\dot{q}$ and $\dot{\omega}$, where ω is the angular velocity. Fig. 2.9a provides a schematic of the perception pipeline: for a given true relative attitude, q, an image, $z(q)$, is generated, and the CNN produces a noisy estimate, $\tilde{q}$, of the relative attitude from this image.

We define the perception map as $\tilde{q} = \mathcal{H}(q)$, where $\tilde{q}$ is the noisy estimate of the true relative attitude, q. To generalize this mapping over the feasible state space of relative poses between Ego and Target, we generate tracking trajectories using a PID controller.



(a) (b)

Figure 2.10: (a) Empirical data showing the relationship between the norm of the true relative attitude, $||q||$, and the measured value, $||\tilde{q}||$, filtered for outliers. Larger $||q||$ results in higher deviations in $||\tilde{q}||$, indicating state-dependent perception error and passivity loss near the origin. (b) Error modeling using an ϵ-ball around the origin; sector-bounded passivity holds outside this region.

We initialize multiple trajectories across the relative attitude state space, corresponding to various Target positions in the Ego's camera frame. Celestial parameters, such as Earth-relative position and solar angles, are randomized across these trajectories. This dataset supports the empirical characterization of the perception map's input-output properties (see Fig. 2.10a).

We model the perception map as shown in Fig. 2.10b. The plot reveals that sector-bounded passivity holds outside a neighborhood of the origin. Additionally, the measurement error norm $||\tilde{q} - q||$ is bounded due to mission constraints. The following set of assumptions encapsulate the sector-bounded error model,

Assumption 2.6.1. ***Sector bounded error model*** *Given a perception map* $\mathcal{H} : \mathbb{R}^3 \to \mathbb{R}^3$ *, such that* $\tilde{q} = \mathcal{H}(q)$*, the following properties hold*

- *Boundedness:* $\|\tilde{q} - q\| \leq \delta$
- *Sector bounds:* $\alpha q^\top q \leq \tilde{q}^\top q \leq \beta q^\top q$ *if* $\|q\| \geq \epsilon$ *for some* $\alpha, \beta, \epsilon > 0$

The perception map is defined as sector bounded if Assumption 2.6.1 holds. This map can be interpreted as a *passive* map. A general nonlinear system is defined as passive as follows:

Definition 2.6.1. *[109] A system* $\dot{x} = f(x, u)$ *with output* $y = h(x, u)$ *is said to be passive if there exists a differentiable positive semidefinite function* $V(x)$ *(called the storage function) such that:*

$$u^\top y \geq \dot{V} = \frac{\partial V}{\partial x} f(x, u), \quad \forall (x, u) \in \mathbb{R}^n \times \mathbb{R}^m$$

Moreover the system is said to be:

- *lossless if* $u^\top y = \dot{(V)}$
- *strictly-passive if if* $u^\top y \geq \dot{V} + \phi(x)$ *for some positive definite function* ϕ.

Furthermore, an input-output map,$y = f(u)$, can be defined as passive if $u^\top y > 0$ [109]. Having defined passivity, we will now discuss controller design for attitude stabilization.

2.6.1 Passivity-Based Feedback Control Setup

We now describe the relative attitude dynamics, q, with respect to the desired tracking attitude, along with the angular rate error ω. The goal is to develop a stable tracking controller using feedback from the perception map. Additionally, we aim to model the closed-loop system as passive, with respect to a chosen input-output pair and storage function, V.

We begin with two passive subsystems: the attitude dynamics and the angular rate dynamics, modeled as follows [207]:

$$J\dot{\omega} = (J\omega) \times \omega + u \tag{2.30}$$

$$\dot{q} = G(q)\omega \tag{2.31}$$

where $G(q) = \frac{1}{2}\left[I_3\left(\frac{1-q^Tq}{2}\right) + S(q) + qq^T\right]$, and $S(q)$ is the skew-symmetric matrix:

$$S(q) = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$

These equations describe the Ego spacecraft's relative dynamics with respect to the reference attitude generated from Target tracking. The control task is to stabilize the attitude, q, to zero using measurements from the perception map. Since the attitude dynamics are passive, we can design a passive interconnection with the ML-based perception module, which exhibits sector-bounded passivity. Feedback interconnections of passive systems are well studied and guarantee stability under suitable conditions [109].

2.7 Passivity for Attitude Stabilization

Passivity between subsystems in attitude stabilization problems has been extensively studied [207, 64]. We build on the our results on passivity-based controllers with perception errors, using the proposition stated by Tsiotras [207]:

Proposition 2.7.1. *[207]*

- *System (2.30) is passive with input u and output ω.*
- *System (2.31) is passive with input ω and output q.*

We can further use stability results from Tsiotras [207] to show that the feedback interconnection of system (2.30) and (2.31) results in a stable system and that the system (2.30) itself is passive.

Proposition 2.7.2. *[207] The system (2.30) with $u = -k_\omega\omega - \nu$ is strictly passive with input ν and output ω.*

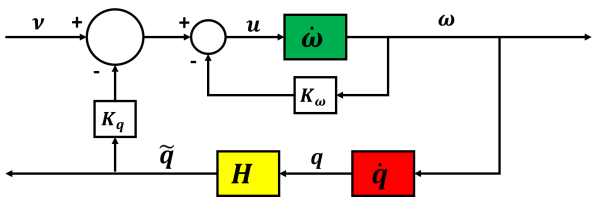


Figure 2.11: Block diagram showing feedback interconnection: the strictly passive attitude rate subsystem and the perception map acting as a generalized passive module.

Next, let the output q of the attitude dynamics system be the input ν to System (2.30). The full interconnection is then defined by the control law $u = -k_\omega \omega - k_q q$. This results in a stable interconnection.

Theorem 2.7.3. *[207] The control law $u = -k_\omega \omega - k_q q$, with $k_\omega, k_q > 0$, globally stabilizes Systems (2.30) and (2.31) for a radially unbounded storage function.*

Here the lyapunov function was defined by $V = \frac{1}{2}\omega J \omega + 2k_q \ln(1 + q^\top q)$. We will further use this result to find feasible gains k_ω, k_q for noisy perception.

2.8 Passivity-Based Control for Nonlinear Observations

Let us now look at the attitude stabilization system under nonlinear perception model. We assume that the perception map has been modeled as a sector-bounded nonlinearity as shown in Figure 2.10b. The sector bounding parameters $\alpha, \beta > 0$ are determined empirically.

2.8.1 Feedback Interconnection Under a Noisy Observer

As discussed in Theorem 2.7.3, the system in Eq. (2.30) is strictly passive with input ν and output ω when the control law is given by $u = -k_\omega \omega - \nu$. Now, we define the system as described in Figure 2.11 where output is generated by perception map,$\mathcal{H}$. The control input becomes $u = -k_\omega \omega - k_q \tilde{q}$, where $\tilde{q}$ is a noisy measurement of the true relative attitude q. We aim to develop a systematic method for handling this noisy feedback.

Building upon Theorem 2.7.3, we propose that a passivity-based controller(PBC) of the form $u = k_\omega \omega + k_q \tilde{q}$.

Building upon Theorem 2.7.3, we propose that a passivity-based controller(PBC) of the form $u = k_\omega \omega + k_q \tilde{q}$. The lyapunov function proposed by Tsiotras [207], is defined as $V = \frac{1}{2}\omega J \omega + 2k_q \ln(1 + q^\top q)$, where we use output feedback given by the perception map. Taking the derivative of lyapunov function with time we get:

$$\dot{V} = u^T \omega + k_q q^T \omega \tag{2.32}$$

$$= (-k_\omega \omega - k_q \tilde{q})^T \omega + k_q q^T \omega \tag{2.33}$$

$$= -k_\omega ||\omega||^2 - k_q (\tilde{q} - q)^T \omega \tag{2.34}$$

Here to ensure stability we need $\dot{V} \leq 0$. By using the perception map model we know that the passivity-short behavior imposes a lower bound on q as characterized in Assumption 2.6.1, where we have, $\|q\| > \epsilon$, for some $\epsilon > 0$. Moreover, to impose input constraints and saturation we want the controller gains k_ω, k_q to be bounded. This imposes a lower bound norm of ω for the stability condition given by Eq. (2.34) to hold. We can observe that from Eq. (2.34):

$$k_\omega \omega^T \omega + k_q (\tilde{q} - q)^T \omega \geq 0$$

$$\frac{k_\omega}{k_q} \omega^T \omega \geq (\tilde{q} - q)^T \omega \tag{2.35}$$

using Cauchy–Schwarz inequality, and error bond condition we can write: $(\tilde{q} - q)^T \omega \leq \|\tilde{q} - q\| \|\omega\| \leq \delta \|\omega\|$. Using this condition on Eq. (2.35), we can write a sufficient condition for Eq. (2.34) to hold by writing

$$\frac{k_\omega}{k_q} \omega^T \omega \geq \delta \|\omega\|$$

$$\|\omega\| \geq \Omega \tag{2.36}$$

$$\tag{2.37}$$

where Ω is design parameter that is an upper bound of $\delta k_q/k_\omega > 0$. The condition in Eq. (2.36), imposed a *lower bound* on ω given an upper bound on the error norm $||\tilde{q} - q||$ exists. Also, the passivity-short behavior of perception map imposed a lower bound condition on q, ie $\|q\| \geq \epsilon$. We define the set of states following the lower bound conditions as *zero-state*, $\mathcal{O}$, and the control policy is defined such that for any state in the zero-state set the control is zero. Therefore, the modified controller is given by

$$u = \begin{cases} 0 & , (q, \omega) \in \mathcal{O} \\ k_\omega \omega + k_q \tilde{q} & , (q, \omega) \notin \mathcal{O} \end{cases} \tag{2.38}$$

where

$$\mathcal{O} = \{q, \omega | \ \|q\| \geq \epsilon, \|\omega\| \geq \Omega\}. \tag{2.39}$$

We reformulate the Lyapunov inequality Eq (2.34) as an LMI by defining an internal state vector:

$$x = \left[(\tilde{q} - q)^\top, \omega^\top, q^\top, 1\right]^T$$

The stability condition Eq (2.34) can be expressed in matrix form as:

$$\begin{bmatrix} \tilde{q} - q \\ \omega \end{bmatrix}^T \begin{bmatrix} 0 & \frac{k_q}{2} \\ \frac{k_q}{2} & k_\omega \end{bmatrix} \begin{bmatrix} \tilde{q} - q \\ \omega \end{bmatrix} \geq 0 \tag{2.40}$$

$$\tag{2.41}$$

We can rewrite this condition as the following LMI:

$$\begin{bmatrix} \tilde{q} - q \\ \omega \\ q \\ 1 \end{bmatrix}^T \begin{bmatrix} 0 & \frac{k_q}{2} & 0 & 0 \\ \frac{k_q}{2} & k_\omega & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{q} - q \\ \omega \\ q \\ 1 \end{bmatrix} \geq 0 \tag{2.42}$$

This inequality can be compactly written as $x^T A_0 x \succeq 0$, defining the stabilizable set of states:

$$\mathcal{X} = \{x \mid x^T A_0 x \succeq 0\}, \quad (q, \omega)) \notin \mathcal{O} \tag{2.43}$$

The zero-state set conditions in (2.43) can be equivalently expressed using internal state x as:

$$x^T A_1 x \succeq 0, \quad x^T A_2 x \succeq 0$$

as stated by condition $T1$ and $T2$ in Table 2.1. We now define the perception map based

Table 2.1: List of Constraint set and Stability definitions as matrix inequalities corresponding to internal state x.

T0	**Stability:** $\dot{V} \leq 0$	$k_\omega \omega^T \omega + k_q(\tilde{q} - q)^T \omega \geq 0$	$x^T A_0 x \succeq 0$, where $A_0 = \begin{bmatrix} 0 & k_q/2 & 0 & 0 \\ k_q/2 & k_\omega & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
T1	**Avoiding zero-state:** $\|\|q\|\| \leq \epsilon$	$x^T A_1 x \succeq 0$	$A_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\epsilon \end{bmatrix}$
T2	**Avoiding zero-state:** $\|\|\omega\|\| \leq \Omega$	$x^T A_2 x \succeq 0$	$A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\Omega \end{bmatrix}$
T3	**Sector upper bound:** Eq. (2.44)	$x^T A_3 x \succeq 0$	$A_3 = \begin{bmatrix} 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 0 \\ -1/2 & 0 & \beta - 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
T4	**Sector lower bound:** Eq. (2.45)	$x^T A_4 x \succeq 0$	$A_4 = \begin{bmatrix} 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1 - \alpha & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
T5	**State bound:** $\|\|q\|\| \leq \gamma$	$x^T A_5 x \succeq 0$	$A_5 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & \gamma \end{bmatrix}$
T6	**Error bound:** $\|\|\tilde{q} - q\|\| \leq \delta$	$x^T A_6 x \succeq 0$	$A_6 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta \end{bmatrix}$

constraints as a expression of the internal state x.

2.8.2 Constraint Set for the Perception Map

To design a robust controller in the presence of measurement noise, we define constraints on the perception map, $\mathcal{H}$, which outputs a noisy estimate $\tilde{q} = \mathcal{H}(q) = q + e$, as shown in

Fig. 2.10b. Empirical simulation results provide the input-output characteristics of this map (Fig. 2.10a), obtained through exhaustive sampling over the neural network's input space.

The behavior of $\tilde{q}$, the perception output, is characterized by the following conditions:

1. **Passivity-short near equilibrium:** Near the origin (i.e., $q \approx 0$), the perception map exhibits non-negligible residual error. Thus, we model $\mathcal{H}$ as passivity-short with respect to the input-output pair $(q, \tilde{q})$. We require:

$$q^T \tilde{q} \geq 0 \quad \text{under the condition} \quad ||q|| > \epsilon.$$

   Here epsilon is the lower bound on q as shown by the box in Figure 2.10b, outside of which the sector bounded non-linearity conditions hold true. The value $\epsilon > 0$ is evaluated empirically from data set generated using the simulator.

2. **Sector-boundedness:** The neural network's input-output relationship is passive and sector-bounded. Tighter sector bounds imply smaller estimation error and capture state-dependence. While a prior study [167] modeled state-dependent error near nominal trajectories, here we consider all trajectories by enforcing:

$$\beta q^T q \geq \tilde{q}^T q \geq \alpha q^T q,$$

   which yields:

$$-(\tilde{q} - q)^T q + (\beta - 1) q^T q \geq 0 \tag{2.44}$$

$$(\tilde{q} - q)^T q + (1 - \alpha) q^T q \geq 0 \tag{2.45}$$

3. **State and error upper bounds:** Mission constraints impose:

$$||\tilde{q} - q|| \leq \delta, \quad ||q|| \leq \gamma.$$

able 2.1 summarizes the constraint sets expressed as matrix inequalities in the quadratic form described in the previous section. These constraints are enforced under the assumption

that outliers and adversarial noise are mitigated using an averaging filter. This is necessary because the ML-based perception map does not inherently satisfy Assumption 2.6.1. Such maps are particularly susceptible to outlier noise, and adversarial perturbations are a well-documented challenge in machine learning [217]. Moreover, the high dimensionality of image-based inputs makes dense sampling of the input space infeasible during training. Consequently, the training set only represents a sparse subset of the input space, and inputs sampled outside this subset may yield outputs that deviate from the nominal behavior. To address this, we apply an averaging filter combined with an outlier rejection algorithm, which helps maintain the system response close to the expected output, even in the presence of noise and uncertainty. This approach ensures robust enforcement of the constraints while accounting for the inherent limitations of the learned perception map.

2.8.3 Feasibility Analysis

We defined all the constraints in Section 2.8.2, with respect to internal state, x, as matrix inequalities shown in Table 2.1. We will refer to these constraints using table references T0 to T6.

Now, we want to find feasible controller gains (k_q, k_ω) such that when all the constraints of perception map, (T3,T4,T6), and state constraints, (T1,T2,T5), are satisfied, then the stability condition (2.43) is also satisfied.

$$x^T A_i x \succeq 0 \Rightarrow x^T A_0 x \succeq 0 \quad \text{for } i = 1, \ldots, 6. \tag{2.46}$$

then Using the S-procedure[4], we eliminate the state x and reformulate this implication as an LMI in terms of positive multipliers $\lambda_i > 0$. The S-procedure states that if there exist positive λ_i such that;

$$A := A_0 - \sum_{i=1}^{6} \lambda_i A_i \succeq 0 \tag{2.47}$$

[4]We use S-Procedure to convert quadratic inequalities into LMIs as described in Section 2.6.3 by Boyd et al [26]

then the condition (2.46) holds true.

Given all the parameters $(\delta, \Omega, \gamma, \epsilon, \alpha, \beta)$, that define the constraints sets, we need to find a feasible controller k_p, k_ω. Equivalently, we can find a feasible PSD matrix A defined in condition (2.47) as a function of k_p, k_ω, λ_i.

We use an SDP solver to characterize feasible λ_i and identify corresponding control gains. The controller is defined as feasible if a solution to the following problem exists:

$$\begin{aligned} \min \quad & c \\ \text{s.t.} \quad & A \succeq 0, \\ & k_p > 0, \\ & k_\omega > 0, \\ & \lambda_i > 0, \quad i = 0, \ldots, 6 \end{aligned} \tag{2.48}$$

Here c is a placeholder cost function, and the solver searches for feasible A. Expanding A, we get:

$$A = \begin{bmatrix} \lambda_5 & k_q/2 & (\lambda_2 - \lambda_3)/2 & 0 \\ k_q/2 & k_\omega - \lambda_6 & 0 & 0 \\ (\lambda_2 - \lambda_3)/2 & 0 & -\lambda_1 - \lambda_2(\beta - 1) - \lambda_3(1 - \alpha) + \lambda_4 & 0 \\ 0 & 0 & 0 & \lambda_1\epsilon - \lambda_4\gamma - \lambda_5\delta + \lambda_6\Omega \end{bmatrix} \succeq 0.$$

where we observe the parameterization of A on gains and λ_i.

Given that the solution to (2.48) exists, the condition (2.46) is satisfied. This gives us the basis for the main Theorem of this work defined as follows:

Theorem 2.8.1. *(PBC for perception based nonlinearity) The output feedback controller Eq. (2.38), asymptotically stabilizes the system (2.30)-(2.31) with nonlinear perception map $\mathcal{H}$ to zero state set $\mathcal{O}$ if there exists PSD matrix A defined by condition 2.47.*

Proof. We have discussed the steps required for constructing this proof in this section. Let us now formalize the procedure.

Given the system (2.30)-(2.31) with nonlinear perception map $\mathcal{H}$, and constraint parameters $(\delta, \epsilon, \gamma, \Omega, \alpha, \beta)$, we define a zero-state set as given by Eq. (2.39). Now consider a candidate lyapunov function such that:

$$V = \left\{ \begin{array}{ccc} \frac{1}{2}\omega J\omega + 2k_q \ln(1+q^\top q), & \text{if} & (q,\omega) \notin \mathcal{O} \\ 0, & \text{if} & (q,\omega) \in \mathcal{O} \end{array} \right\}. \tag{2.49}$$

We take the time derivative as:

$$\dot{V} = \left\{ \begin{array}{ccc} -k_\omega ||\omega||^2 - k_q(\tilde{q}-q)^T\omega, & \text{if} & (q,\omega) \notin \mathcal{O} \\ 0, & \text{if} & (q,\omega) \in \mathcal{O} \end{array} \right\}. \tag{2.50}$$

Here V is continuously differentiable in $\mathcal{O}$ and is defined as $V = 0$ in $\mathcal{O}$. We construct V, such that on the boundary of zero set, $\dot{V}$ is smooth. We also assume that the annulus[5] where $\dot{V}$ transitions from $\dot{V} = 0$ to $\dot{V} = -k_\omega ||\omega||^2 - k_q(\tilde{q}-q)^T\omega$ is arbitrarily small, therefore the form of $V := g(V(q,\omega))$ in the annulus is not explicitly stated. Also the control is set $u = 0$ in the annulus so that system is well behaved around zero state.

We can now state that by solving the problem (2.48), we can find feasible controller gains k_q, k_ω such that $\dot{V} < 0$ for all $(q,\omega) \notin \mathcal{O}$ and $\dot{V} = 0$ in zero state. Use Lasalle's Theorem stated as Theorem 4.4 [109] we can state that the controller (2.38) asymptotically stabilizes the system to the zero state. $\square$.

Proposition 2.8.2. *For the feedback interconnection with $u = -k_q\tilde{q} - k_\omega\omega$, and $A \succeq 0$, the system is passive.*

Proof. Consider the composite Lyapunov function:

$$V(\omega, q) = V_1(\omega) + k_q V_2(q) = \frac{1}{2}\omega^T J\omega + 2k_q \ln(1+q^T q)$$

[5]buffer zone around zero state set

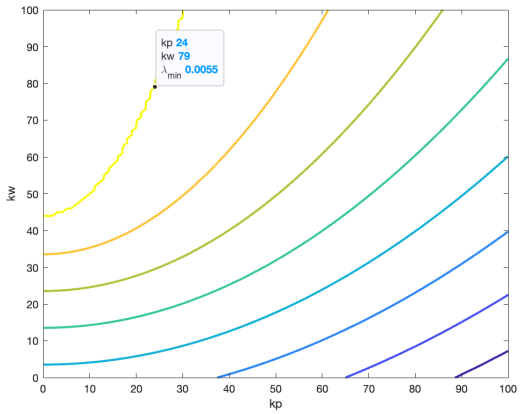


Figure 2.12: Feasible set for controller gains (yellow region), where $A \succeq 0$

With control law $u = \nu - k_q\tilde{q} - k_\omega\omega$, we get:

$$\begin{aligned}\dot{V} &= \omega^T J\dot{\omega} + 4k_q\frac{q^T\dot{q}}{1+q^Tq} \\ &= \omega^T u + k_q q^T\omega \\ &= -k_\omega\omega^T\omega - k_q(\tilde{q}-q)^T\omega + \nu^T\omega \\ &\leq \nu^T\omega \end{aligned} \tag{2.51}$$

Hence, the system is passive from input ν to output ω. □

On the Use of Linear Error. Note that the linear error term $(\tilde{q} - q)$ is a valid representation only when the deviation between the true and measured attitudes is small. In our simulations, this deviation is typically three orders of magnitude smaller than the nominal attitude state. Hence, the linear approximation is appropriate for representing the estimation error near the origin.

2.8.4 Zeroth-order Optimization using Sim Engine

The previous analysis characterizes a feasible set of controller gains that meet stability and passivity constraints. We now want to optimize gains over the feasible set to facilitate tasks like fuel saving and smooth trajectory generation. We may be required to optimize the controller defined in Eq. 2.38, if constraints such as torque limitations or time sensitivity of tracking exist in the mission scenario. Thus, we present an example where we use a quadratic cost that penalizes state error and input energy. Specifically, the cost for initial states q_0, ω_0 is given by:

$$J = \int_0^T q_t^T Q q_t + \omega_t^T W \omega_t + u_t^T R u_t \, dt \tag{2.52}$$

$$\tag{2.53}$$

where Q, R are PSD matrices that penalize state and input states with reference to optimality. We now pick optimal controller gains minimizing this cost over feasible initial states, such that the passivity and stability constraints are satisfied. Thus the problem can be defined as:

$$\begin{aligned}
\arg_{k_p,k_w} \min_{[q_0\sim\mathcal{D}(0,\Sigma_q),\omega_0\sim\mathcal{D}(0,\Sigma_\omega)]} \quad & \int_0^T q_t^T Q q_t + \omega_t^T W \omega_t + u_t^T R u_t \, dt \\
\text{s.t.} \quad & \dot{q}_t = G(q_t)\omega_t \\
& J\dot{\omega}_t = -S(\omega_t)J\omega_t + u \\
& \tilde{q}_t = \mathcal{H}(q_t) \\
& u_t = k_\omega \omega_t + k_q \tilde{q}_t \\
& A \succeq 0
\end{aligned}$$

This is a nonlinear optimization problem where we assume that the time horizon, T, is larger than the time required to stabilize. The initial states for q, ω are sampled from a distribution

of feasible initial conditions for the attitude stabilization problem as defined by the mission constraints. We sample initial conditions such that the perception map generates a valid result, i.e., the Target spacecraft is in the visible range of Ego. The constraint, $A \succeq 0$, ensures the stability and passivity constraints defined in Table 2.1.

The feasible region is visualized in Fig. 2.12, where controller gains in the yellow region yield $A \succeq 0$.

Algorithm 1 :Naive multi-point Zeroth-order Optimization on constraint set

Initialize $x_0 \sim \mathcal{D}(0, \Sigma), K_0 = [k_q, k_\omega], \epsilon$

while $C_i - C_{i-1} \geq \epsilon$ **do**

  for $j = 1, .., N$ **do**

    $U_j \sim \mathcal{B}_r$ ▷ sample over boundary of norm ball with radius r

    $K_j = K_{i-1} + U_j$ ▷ re-sample if K_i is infeasible

    for $t = 0, .., T$ **do** ▷ repeat for M trajectories

      $$u_t = -K^T \begin{bmatrix} \tilde{q}_t \\ \omega_t \end{bmatrix}$$

      $$\dot{q}_t = G(q_t)\omega_t$$

      $$J\dot{\omega}_t = -S(\omega_t)J\omega_t + u_t$$

      $$C_j = C_j + q_t^T Q q_t + \omega_t^T W \omega_t + u_t^T R u_t$$

    end for

    $C_i = \min C_j$

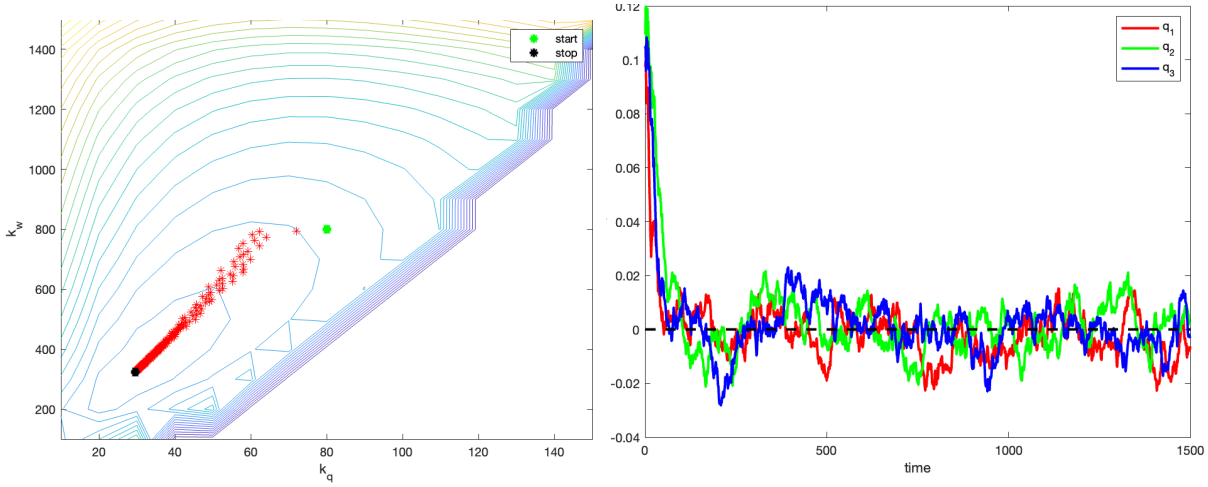
    $K_i \rightarrow \arg\min_{K_j} C_j$

    $i = i + 1$

  end for

end while

We now use a variation of zeroth-order optimization [127] method for computing optimal



(a) Quadratic cost on a set of trajectories initialized from $x_0 \sim \mathcal{D}(0, \Sigma)$. The trajectories are simulated for a time horizon T with passivity based control law and ML based estimation. For feasible sets of control gains we obtain an L-smooth mapping from gains to the cost function. Here we also show iterative updates of Zeroth-order optimization. The plot shows infeasible region on the bottom right. The initial control gain shown in green and optimal controller in black and iterations are plotted in red.

(b) Attitude trajectory under noisy observation. The MRP attitude states are stabilized to a reference $[0, 0, 0]$ i.e., camera center.

Figure 2.13: Trust region algorithm analysis for different observability exploration conditions used as termination conditions.

controller gains. The Algorithm [1] uses sample-based gradient information to iterate over minimizing gains k_p, k_ω. We initialize M trajectories with a initial feasible control law $u = K_0 [\omega, q]^T$, where $K_0 = [k_\omega, k_q]$. For each iteration, a cumulative cost is computed for the M

trajectories we initialized. The Figure 2.13a shows the sample space for the cumulative cost which is smooth in the region of feasible gains selected here, given that number of sampled trajectories M is large [127]. For a given iteration, we randomly sample controller gains around K_i, the controller at iteration i, in a ball of radius r, which defines the learning rate. The controller updates in the decreasing gradient direction, similar to a stochastic gradient algorithm. We also iteratively reduce the learning rate, r, as the iterations converge.

We use the optimal controller gains in the simulation pipeline, and the attitude can be observed to have been stabilized to within the zero-state set, as seen in Figure 2.13b.

2.9 Multi-Agent Consensus

We have established a zeroth-order optimal attitude stabilization system for Ego-Target tracking problem, which is passivity-short under noisy observers. The setup can now be extended from single agent pairing to the case of distributed estimation.

We examine a problem setup where multiple Ego spacecraft are in orbit around a single uncontrolled Target spacecraft. Each of the Ego-Target pairs has the passivity-based attitude stabilization controller to keep alignment. A fully connected communication network exists between each Ego agent and they can share their estimated relative Target locations and their own global orientation and position with respect to Earth fixed frame.

Denote the measurement derived from consensus on desired Ego attitude derived from all agents as $\tilde{q}_c$. We substitute the measurement in our previous analysis and use the control input given by $u_1 = -k_q\tilde{q}_c - k_\omega\omega$.

Thus the stability constraint set can now be written as:

$$\dot{V}_1 = -k_\omega\omega_1^T\omega_1 - k_q(\tilde{q}_c - q_1)^T\omega_1 \ . \tag{2.54}$$

Using the setup for feasibility analysis we can replace $\tilde{q}_1$ with $\tilde{q}_c$ by giving reduced bounds

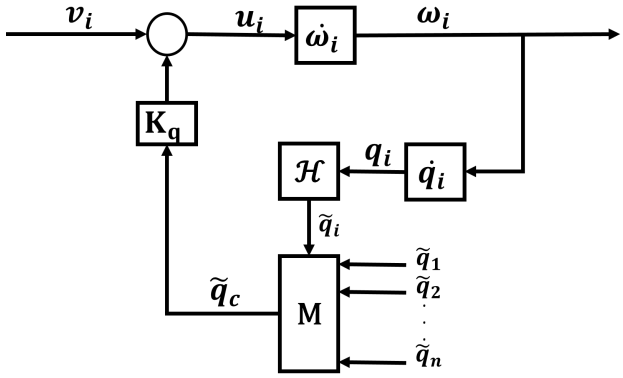


Figure 2.14: Feedback interconnection of multi-agent Ego-Target systems. Here system i is the primary Ego-Target pair. Measurements from all n passive agents are used as feedback to system i. The mixing matrix M is representation for consensus based measurement $\tilde{q}_c$.

on $\tilde{q}_c - q_1$. The constraints from Section 2.8.2 follow here with $\tilde{q}_c$ with new sector bounds α', β' and new error bound δ' and new feasibility set is defined such that $A' \succeq 0$. Tighter bounds increase the size of the feasible control set and we can design a controller for a stricter quadratic cost by weighting the control penalty lower to achieve an aggressive controller, which is currently not achievable as larger feasible set limits aggressive control. Design of the consensus based controller can now be achieved using the same procedure discussed from Section 2.7-2.8.4 by replacing $\tilde{q}_1$ with $\tilde{q}_c$.

2.9.1 Weighted Averaging Consensus approach

We define relative Ego attitude measurement for first Ego-Target pair as $\tilde{q}_1$, and the information shared from neighboring Ego spacecrafts in the frame of reference of the first spacecraft as $\tilde{q_2}, \ldots, \tilde{q_n}$ for n agents. Taking a weighted average for observation on the Ego attitude we can write:

$$\tilde{q}_{\text{avg}} = \frac{1}{n} \sum_{i=1}^{n} \alpha_i \tilde{q}_i \tag{2.55}$$

This sum converges to a weighted mean in the convex hull defined by $\tilde{q}_i$, for $\alpha_i > 0 \ \forall i$

and $\sum \alpha_i = 1$.[6] In the scenario where the confidence on a measurement of a single agent is significantly higher than others, we would prefer to trust that particular measurement as opposed to an average of all measurements. Thus, we also look at a max consensus approach.

2.9.2 Dynamic Max consensus protocol

We show the use of a max consensus protocol in distributed pose estimation. The averaging consensus cannot offer analytical bounds on deviation from the true state as we are using a sector bounded error model with non-Gaussian distribution. Using weighted averaging, we give more emphasis to agents with higher confidence in their measurements. This max consensus is a sort of variation on weighted averaging where we pick only the measurement with the highest confidence as the observation.

Here we use a confidence parameter, α_j, on the relative distance measurements made by a CNN in j^{th} Ego's frame. Transforming the measurement to the i^{th} agent, the measurements become $q_{ij} = \mathcal{F}_{ij}(q_j)$. Now we do a consensus on max confidence for each agent and pick the measurement with the highest confidence, namely,

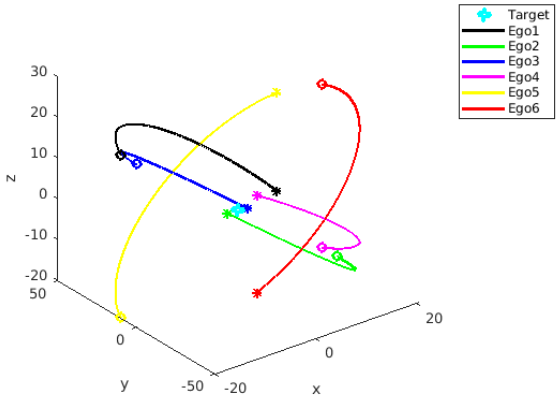
$$\tilde{q}_i = \tilde{q}_{ij} \quad \text{where} \quad j = \arg \max_{\forall j \in \{1,2,..,n\}} \alpha_j \tag{2.56}$$

This consensus methodology also applies to different network configurations.

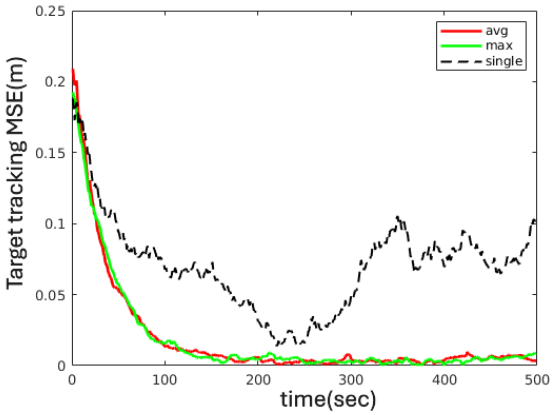
2.10 Simulation and Numerical Experiment

The simulation setup used here is based on a multi-agent Ego-Target tracking problem. We initiate a trajectory for a tumbling Target satellite with six Ego agents in passive relatively elliptical orbits around the Target. The relative orbits of each Ego agent are shown in Fig-2.15a. The Ego satellites have a camera attached to the body x-axis and are independently

[6]Other averaging schemes that are consistent with the geometry of MRPs can also be considered



(a) Ego trajectories around Target shown in blue.



(b) Comparison of attitude tracking error over a trajectory. Tracking for an agent with poor estimation without consensus compared to weighted averaging and max consensus based estimation.

tracking the Target using the passivity-based controller. The onboard attitude and angular velocity estimation is done through an EKF under the assumption of Gaussian measurement noise for the states. The details of assumptions on the EKF filter are listed in Section 2.10.1.

The Figure 2.15b shows a comparison of consensus algorithms compared to single-agent tracking. We show that weighted averaging consensus and dynamic consensus both show reduced tracking error as compared to single-agent tracking. The Ego agents are initialized such that Target is off-center from each of their camera viewpoints, and the primary task for each Ego is to stabilize their attitude such that Target is acquired at their camera center. The single-agent tracking shows tracking information from only the primary Ego, which shows some state dependence of perception error due to changing orbital conditions. In the case of both the averaging and the dynamic consensus, we see that the tracking performance for primary Ego improves as information is shared across the network, as shown by red and green traces in Figure 2.15b. The multi-agent setup is simulated under a fully connected network

thus all agents communicate their state and measurement information to every other agent in global reference frame. The details of the simulation framework, are given in [20]

2.10.1 Simulation with Noisy measurements

Due to onboard sensing with noisy sensors we need to state assumptions for the Ego state estimation. An on-board Extended Kalman Filter(EKF) [1] is implemented on each Ego spacecraft to estimate ω. We simply state the following assumption

Assumption 2.10.1. *Given the system Eq. 2.30, we assume there exist known, additive-gaussian process and measurement noise, $w_t \sim \mathcal{N}(0, Q)$ and $v_t \sim \mathcal{N}(0, R)$ respectively. An EKF is applied to this system which results a state estimate $\hat{\omega}$. We claim that the estimation error is bounded using Theorem 3.1 from [170].*

Additional assumptions required for Assumption 2.10.1 to hold, as stated in Theorem 3.1 from [170]:

Assumption 2.10.2. *The following assumptions are stated for boundedness:*

1. *There exist positive reals $\alpha, \beta, \gamma_1, \gamma_2 > 0$, for each time $t > 0$ such that:*

$$||A_t|| \leq \alpha$$
$$||C_t|| \leq \beta$$
$$\gamma_1 I \leq P_t \leq \gamma_2 I$$

   where $A_t = \partial f/\partial x, C_t = \partial h/\partial x$ are the linearization jacobians for system Eq. 2.30.

2. *A_t is non singular for all t*

3. *There are positive real numbers $\epsilon_\phi, \epsilon_\chi, \kappa_\phi, \kappa_\chi > 0$ such that:*

$$||\phi(x_k, \hat{x_k})|| \leq \epsilon_\phi ||x_k - \hat{x_k})||^2 \quad with \quad ||x_k - \hat{x_k})|| \leq \kappa_\phi$$
$$||\chi(x_k, \hat{x_k})|| \leq \epsilon_\chi ||x_k - \hat{x_k})||^2 \quad with \quad ||x_k - \hat{x_k})|| \leq \kappa_\chi \quad ,$$

   Functions ϕ, χ have been defined in (10) [170].

We define parameters such that Assumption 2.10.2 is satisfied and Assumption 2.10.2 holds true. Thus we have a system where estimation errors are bounded.

2.11 Discussion

We presented a passivity-based controller under non-linear observations synthesized from machine learning-based algorithms. The problem setup we analyzed was aimed at solving the Ego-Target satellite tracking using only a single camera. Our work presented in [20] discusses the ML-based algorithms we use for estimation and the simulation framework based on Unreal Engine to produce photo-realistic visualization and inputs.

The characteristics of our CNN-based estimator are represented as sector-bounded non-linearity. This modeling gives us a passivity-short framework, and we define the controller that works in a region defined by all feasible attitude trajectories for the Ego-Target pair. Among the set of feasible controllers, we pick an optimal controller that minimizes input energy and tracking error using a zeroth-order optimization approach from the data gathered through our simulation engine.

The passive nature of the system allows us to extend the framework to a multi-agent setup for passive systems. We use a consensus-based estimation from each agent, CNN-based measurement. Using passivity properties of each agent, we ensure that the feedback interconnection remains passive and the controller setup can be extended to the consensus case.

We demonstrate the use of a weighted averaging and a max-consensus algorithm based on confidence parameters assigned to the measurements by the perception map. The consensus algorithms show reduced tracking errors under the same controller due to better estimation, as seen in Figure 2.15b. In this work, we only look at a centralized approach using feedback from each agent in the network. The likely future direction would evaluate a distributed controller.

Extensions to this work include a generalized controller, for a broader domain of passivity short systems with sector bounded non-linearities. Moreover, adaptive controllers that can

exploit the state dependence of error models for CNN-based measurement are a pertinent future direction for this work.

Chapter 3

SET-VALUED OBSERVABILITY

Trajectory planning often involves the solution of optimization problems that efficiently guide a vehicle to a desired final state, while ensuring feasibility with respect to constraints on the vehicle's state and input [134]. This work addresses a trajectory planning problem that accounts for the vehicle's state estimation process—a consideration particularly critical in safety-critical systems where the accuracy of state estimation is essential.

Traditionally, in offline planning, the estimation process is assumed to be independent of state and control, causing the synthesized trajectory to potentially traverse regions with high measurement uncertainty. Meanwhile, in many practical applications, large measurement errors or disturbances adversely affect the quality of the corresponding state estimation; in such scenarios, estimation-aware trajectory design becomes indispensable [30]. In order to follow a trajectory, an agent often employs a state estimator, mapping the process measurements to full state estimates, provided that the system is observable [107, 92, 142, 106, 197]. The performance of this estimator is closely tied to the quality of the sensor measurements. If the sensor performance is state-dependent, then visiting states where the sensor has better performance can improve the corresponding state estimation process. One common approach to design optimal trajectories for improved state estimation is to incorporate a secondary objective into the original trajectory optimization problem [8, 143]. This augmented objective function must account for the quality of the estimation process with respect to errors in the measurement. Given knowledge of the measurement error distribution, a suitable metric for capturing the estimation quality can then be developed and embedded in the design objective. Performance metrics for evaluating the estimation performance include measures on error probabilities [23], mean squared errors [240, 113, 239], Fisher information [76, 112, 75],

and entropy and other metrics on the belief state. For example, a commonly used construct in settings with a Gaussian measurement noise is the Fisher Information Matrix (FIM) [112, 75]. In a nutshell, FIM quantifies the information content about an unknown state (or parameter) in a measurement. In fact in the matrix case, the inverse of FIM provides a lower bound on the expected error covariance, using the Cramer-Rao inequality [159, 169], i.e., $\mathbb{E}\{(\hat{x}-x)(\hat{x}-x)^T\} \geq \mathcal{I}^{-1}$, where $\mathcal{I}$ denotes the FIM, and x and $\hat{x}$ denote the state of interest and its estimate, respectively; this lower bound represents the error covariance achieved by an unbiased estimator. Various properties of FIM, such its determinant, spectrum, and trace, have been used in different applications [187].

The FIM-induced metrics utilized for quantifying the information content of a measurement scheme are effective for parameter estimation problems when the underlying state is static. On the other hand, measures such as posterior FIM have been studied to analyze state estimation for linear time-invariant (LTI) systems [150, 224]. Optimization-based approaches have also been developed to demonstrate the utility of error covariance metrics for trajectory planning problems [151, 166].

Another framework for designing estimation-aware trajectories is via observability-based metrics. Specifically, the Observability Gramian (OG) of a system, mapping initial conditions to an output trajectory sequence, can be used to assess how "sensitive" state estimation is to perturbations in output measurements. In this direction, prior studies have demonstrated that properties of the Empirical Gramian (EG) can be leveraged to improve conditioning for the estimation process [160, 17]. These works have shown that using EG generated from the model prediction, an optimization problem can be formulated to synthesize a qualitatively more "observable" trajectory. These trajectories have improved estimation performance based on the type of metric used on the EG [8, 7, 27, 82]. These studies primarily focus on applications with deterministic output maps with Gaussian noise measurements, that can be propagated forward, multiple roll-outs at a time [111, 116]. In addition, probabilistic models using covariance-based optimization have been examined in [84, 225]. Other motion planning approaches use Rapidly Exploring Random Trees to determine estimation-aware

trajectories [124]. And lastly, there are methods proposed in the literature that rely on application-specific correlations between estimation and the underlying state for trajectory design [4].

In this work, an alternative–yet complementary–perspective to the aforementioned lines of work has been provided, particularly for scenarios where measurements are *set-valued* and conventional methods for EG computation are infeasible due to Gaussian noise assumption. However, similar to metrics used EG-based approaches, here, a set distance metric is used to improve the conditioning of the estimation process over a trajectory. In the area of filtering for set-valued uncertainty, set-membership estimation has been investigated solely for the purpose of filtering, where estimation procedures for set-based outputs are analyzed [199, 200, 60]. To the best of our knowledge, studies on estimation-aware planning with set-based output uncertainties have been rather limited, signifying the need for "filtering" methods for scenarios where noise distributions are unknown yet bounded. This work specifically focuses on measurement spaces with state-dependent uncertainties that are often non-Gaussian, for example, as they arise in perception-aware planning. In this direction, an explicit approach for such planning problems for locally linearized systems is presented. Specifically, notions of finite-time observability for nonlinear systems with set-valued state-dependent output uncertainties are first presented. It is important to point out that in this setup, no specific assumption is made on the distribution for the uncertainties within the set; however, a constraint that the corresponding uncertainty set is compact (i.e., closed and bounded in a finite-dimensional Euclidean space) is enforced. Additionally, it is assumed that a metric (introduced in Section 3.2), on the size of the uncertainty set has a convexity property with respect to the state vector. This construction, suggests an efficient solution strategy for the corresponding optimization problem. In subsequent sections of this chapter, justifications for this assumption and its conservatism will be discussed. [1]

In this setup, the output map is represented using a bounded set-valued uncertainty over

[1]The size of a compact set can be defined by its diameter or formally as the maximum distance between any two points in the set.

a linear output measurement. Although the convexity of the size of this uncertainty with respect to the state is not always guaranteed in real-world applications, smooth variations in the uncertainty and its state dependence are often encountered in practice. In order to address such scenarios, an approximation is proposed for the uncertainty set, using an enveloping convex function that provides a tight upper bound on this set. Using the convex output map (exact or approximate), a metric is constructed for observability that correlates with the estimation quality. Optimizing this notion of observability is shown to result in better performance for the estimation process. The observability metric is introduced in Section 3.2 as a convex surrogate to manage uncertainty, subsequently incorporated into a generic optimal trajectory design problem, as presented in Section 3.5.1.

This chapter is also an extension of the previous work reported in [56], where a pipeline was proposed to design estimation-aware trajectories using nonlinear model predictive control via distinguishability-based notions. In this work, on the other hand, a framework for establishing guarantees for the existence of estimation-aware trajectories is examined. Specifically, it is shown that a convex optimization problem can be formalized to solve for the desired trajectory when certain conditions on output uncertainty sets are met.

3.1 Set-valued uncertainty measurement model

This section formalizes an observability-based metric that is subsequently used to design estimation-aware trajectories. In particular, this metric is developed such that maximizing its *lower bound* improves observability of the state, hence improving the measurement-driven estimation process. Subsequently, it is shown that this metric can be augmented to the objective of a baseline trajectory generation algorithm.

The model that is considered for the trajectory planning problem is built upon the discrete-time nonlinear system of the form,

$$x_{t+1} = f(x_t, u_t), \tag{3.1}$$

$$y_t = h(x_t), \tag{3.2}$$

$$y'_t = h(x_t) + \varepsilon_t, \tag{3.3}$$

with n_x, n_u and n_y as the state, input and output dimensions; in this case, $x_t \in \mathcal{X} \subset \mathbb{R}^{n_x}$ and $u_t \in \mathcal{U} \subset \mathbb{R}^{n_u}$ are feasible state and control inputs for $t = 0, 1, 2, \ldots, T$ and the term ε_t are set-valued, i.e., $\varepsilon_t \in \mathcal{E}$, as opposed to say, point-wise deterministic measurements or sampled from a known probability distribution. In particular, the output y'_t is sampled uniformly from a compact set centered around the nominal output $y_t = h(x_t)$. The functions $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ and $h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ are assumed to be differentiable.

The set of possible outputs with respect to the measurement uncertainty set $\mathcal{E}$ is defined as $\mathcal{Y}_x = h(x) \oplus \mathcal{E}$, where "$\oplus$" denotes the set addition;[2] $\mathcal{E}$ encodes the uncertainty in the observations due to noise, disturbances, or other unknown parameters effecting the output. In this work, the measurement uncertainty is characterized by maximal ellipsoids for which axial radii can be computed. The size of the uncertainty set is then defined as the maximum ellipsoidal radius. A "variation" in this notion of uncertainty is then utilized in the subsequent optimization formulation of estimation-aware planning. The maximal ellipsoid will bound the set-valued uncertainty observed in the model by a state-dependent set centered, around the nominal measurement $y = h(x)$. A general state-dependent ellipsoidal set centered at x_0 is defined as,

$$\mathcal{E}(x_0, Q(x_0)) = \left\{ x \mid (x - x_0)^\top Q(x_0)^{-1} (x - x_0) \leq 1 \right\}, \tag{3.4}$$

where $Q(x_0) \in \mathbb{S}^+$ and is dependent on x_0; occasionally, the short-hand notation $\mathcal{E}(x_0)$ for a generic ellipsoid centered at x_0. For $x_0 \in \mathcal{X}$ is adopted. The point-to-set map is defined as,

$$\mathcal{Y}_{x_0} := \mathcal{E}(y_0, Q(x_0)), \tag{3.5}$$

[2]The set generated by adding $h(x)$ to all elements of $\mathcal{E}$.

where $\mathcal{Y}_{x_0}$ denotes the ellipsoidal output uncertainty around $y_0 = h(x_0)$; note that the ellipsoidal output is a dependent on the corresponding state. Subsequently, y is used to denote the deterministic output corresponding to the state x, and $\mathcal{Y}_x$ as the corresponding output uncertainty set. The map from the state $x \in \mathcal{X}$ to the set $\mathcal{Y}_x$ defined in Eq. (3.5), is denoted simply by $\mathcal{Y}$ for brevity Ellipsoidal sets are often used for representing uncertainty sets [33]; in this work, these ellipsoids are used to represent and approximate uncertain set-valued output measurements in this work.

The key problem addressed in this work is the selection of the input sequence such that the corresponding state sequence remains "distinguishable" throughout its trajectory. Notions of distinguishability for characterizing the observability of a system have been examined by Hermann and Krener for a general class of deterministic systems [98]. In this work, these notions are modified to reason about the observability of uncertain set-valued output sequences. Note that in this setting, the map of state to output uncertainty ellipsoids is assumed to be known. Otherwise, its action has to be validated in a simulation environment. This assumption is analogous to the case where the error covariance of a sensor is assumed to be given in the filter design. For the optimization approach to estimation-aware planning, a metric is required to quantify the size of the uncertainty set. Here, a conservative bound for comparing ellipsoidal uncertainty sets is adopted, based on their radii. Specifically, the maximum eigenvalue of the positive definite matrix defining the ellipsoid will be used as this metric.

Note that for an ellipsoid centered at x, the distance from x to any other element of the ellipsoid is bounded by the largest eigenvalue of the positive definite matrix defining the ellipsoid. Hence,

$$\sup_{y \in \mathcal{Y}_x} d(x, y) = \lambda_{\max}(Q_x); \tag{3.6}$$

the notation $\Lambda(\mathcal{Y}_x)$ denotes the largest eigenvalue of the ellipsoidal uncertainty corresponding to state x; see Figure 3.1a. In this figure, we show the map $\mathcal{Y}$ transports x to its uncertainty set $\mathcal{Y}_x$, shown in blue, and the function $\Lambda(\mathcal{Y}_x)$ covers the uncertainty with its maximum size

shown in gray.

Given this measure of ellipsoidal uncertainty for a state, its variation as a function of the state can be quantified. A key assumption in the subsequent analysis is as follows.

Assumption 3.1.1. *Let the largest eigenvalue of the output uncertainty ellipsoid corresponding to state x be denoted by $\Lambda(\mathcal{Y}_x)$. The map Λ is convex and uniformly bounded with respect to $x \in \mathcal{X}$.*

Convexity of the above measure of uncertainty implies Lipschitzness of $\Lambda(Q_x)$ [174]; as such, for any feasible $\bar{x}, x \in \mathcal{X}$ one has,

$$|\Lambda(\mathcal{Y}_x) - \Lambda(\mathcal{Y}_{\bar{x}})| \leq L(\bar{x}, r)\, \|x - \bar{x}\|. \tag{3.7}$$

For the purpose of subsequent discussion, the notation $\bar{x}$ in Eq. (3.7) signifies a state on the nominal trajectory; the state x on the other hand, belongs to an open neighborhood U of $\bar{x}$. In Eq. (3.7), $L(\bar{x}, r)$ is the "local" Lipschitz parameter. This parameter can be chosen as,

$$L(\bar{x}, r) = \frac{2M(\bar{x}, r)}{r}, \quad \text{where} \quad M(\bar{x}, r) = \max_{x \in B_r(\bar{x})} \{\Lambda(x)\}\,; \tag{3.8}$$

in Eq. (3.8), M represents the local bound on $\Lambda(\bar{x})$ in the neighborhood $B_r(\bar{x})$ of x. One can observe that $L(\bar{x}, r)$ is convex with respect to $\bar{x}$. Note that if the analytical expression for $\Lambda(\bar{x})$ is given, then $L(\bar{x}, r)$ can be obtained via (3.8) for a given radius r. For certain applications where a perfect sensor model is not known but Assumption 3.1.1 holds, both $L(\bar{x}, r)$ and $\Lambda(\bar{x})$ can be characterized via a simulation oracle.

Equation (3.7), gives the inequality,

$$|\Lambda(\mathcal{Y}_x)| \leq |\Lambda(\mathcal{Y}_{\bar{x}})| + L(\bar{x}, r)\|x - \bar{x}\|; \tag{3.9}$$

picking $r = \|\bar{x} - x\|$, one can obtain a viable yet conservative bound on $|\Lambda(\mathcal{Y}_x)|$. For constructing the set-valued observability metric, set $r = \delta x$, for the maximum deviation δx from the trajectory for which the metric is being computed. Assumption 3.1.1 and inequality Eq. (3.9) facilitate "control" over the size of the variation in output ellipsoidal uncertainty.

A note on convexity of the output map

The generation of estimation-aware trajectories can be mathematically formalized when one can quantify how certain states improve the state estimation process relative to others. Conversely, if the uncertainty set is uniform over all states, estimation -aware planning becomes unnecessary, as it does not improve observability. For cases with a known state dependency on the state, the aim is to design an optimization problem that produces a unique exploration trajectory. In Section 3.2, it is shown that Assumption 3.1.1 plays a crucial role in ensuring a unique solution to this planning problem.

In some applications, state dependence may not exhibit a convexity property but instead display a local monotonicity that aids exploration. For example, quasi-convexity of state-dependent uncertainties become relevant in such cases; this is common in vision-based sensing, for example, where identifiers or features are most visible at specific states, with sensing accuracy diminishing monotonically as one moves away from these optimal visibility states. To address such scenarios, the size of the uncertainty set is approximated using an enveloping convex function. This function is defined as

$$\hat{\Lambda}(x) = \inf\{g(x) \,|\, g \text{ is convex and } g(x) \geq \Lambda(x)\}, \tag{3.10}$$

for x in a domain of interest. The selection of candidate convex function g depends on specific applications. [3] Note that this approach results in an over-approximation of the upper bound. In subsequent analysis, notation Λ refers to the envelope $\hat{\Lambda}$.

Using the enveloping function, the observability metric is computed using a separation function. This separation function is a distance metric between sets, defined to be positive when "distinguished" trajectories are generated from neighboring initial conditions. In practice, designing the enveloping function from the data is referred to as the validation procedure. For the case study on the Ego-Target problem, this validation procedure is detailed in Sections 3.5.3 and 3.5.4.

[3] In the context of this work, the enveloping function g is required be convex. The convex metric derived from this function will be used as the objective in an optimization problem. For faster convergence rates with say gradient descent, one can select g to be strongly convex.

Set Distance metric

For the upcoming observability analysis, one has to define a notion of separation distance between two sets.

Definition 3.1.1. *(Set Distance) The distance between sets $\mathcal{A}$ and $\mathcal{B}$ in $\mathcal{X}$ is defined as,*

$$d_s(\mathcal{A}, \mathcal{B}) = \inf \{d(v, w) \mid v \in \mathcal{A}, w \in \mathcal{B}\}, \tag{3.11}$$

where $d(v, w)$ for a pair of elements in $v, w \in \mathcal{X}$ is their Euclidean distance.

Note that Eq. (3.11) defines a *pseudo-metric* on sets in $\mathcal{X}$, as $d_s(\mathcal{A}, \mathcal{B}) = 0$ does not imply that $\mathcal{A}$ and $\mathcal{B}$ are the same set; instead, this distance is zero when $\mathcal{A} \cap \mathcal{B} \neq \emptyset$. In this chapter, for simplicity, the pseudo-metric referred as a metric, only on separated sets. It is implied that metric properties of the distance function Eq. (3.11) holds when sets are separated and that $d_s(\mathcal{A}, \mathcal{B}) = 0$, also written as $\mathcal{A} = \mathcal{B}$, implies their *non-separability* instead of them being identical. The same interpretation holds for the tube distance "metric" defined subsequently. As such, in this analysis, maximizing set separation is only applied for separated sets.

The set distance Eq. (3.11) is used to quantify separation between the uncertainty ellipsoids; in particular, a nonnegative lower bound for this distance uses Eq. (3.11). In this direction, consider the sets $\mathcal{E}(x_1)$ and $\mathcal{E}(x_2)$, with $v \in \mathcal{E}(x_1)$ and $w \in \mathcal{E}(x_2)$. Since,

$$d(x_2, v) \leq d(v, w) + d(w, x_2), \tag{3.12}$$

$$d(x_1, x_2) \leq d(x_1, v) + d(x_2, v), \tag{3.13}$$

by substituting Eq. (3.12) in Eq. (3.13) we obtain,

$$d(v, w) \geq d(x_1, x_2) - d(x_1, v) - d(w, x_2) > 0, \tag{3.14}$$

where the last inequality for positivity implies that these sets are separated. Taking the infimum in Eq. (3.14), it now follows that,

$$\begin{aligned} \inf_{v,w} \; d(v, w) &\geq \inf_{v,w} \{d(x_1, x_2) - d(x_1, v) - d(w, x_2)\} \\ &\geq d(x_1, x_2) - \sup_{v} \{d(x_1, v)\} - \sup_{w} \{d(w, x_2)\}. \end{aligned} \tag{3.15}$$

A simple lower bound on this set distance is given by,

$$d_s(\mathcal{E}(x_1), \mathcal{E}(x_2)) \geq \|x_1 - x_2\|_2 - |\Lambda(Q(x_1)) - \Lambda(Q(x_2))|. \tag{3.16}$$

In order to ensure that two ellipsoidal sets with centers x_1, x_2, are *separable*, we will enforce the inequality,

$$d(x_1, x_2) \geq \Lambda(Q(x_1)) + \Lambda(Q(x_2)); \tag{3.17}$$

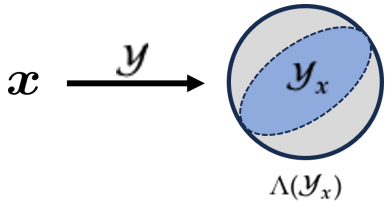
this will ensure that the corresponding set distance Eq. (3.16) is in fact a pseudo-metric.

3.2 Observability with set-valued output uncertainty

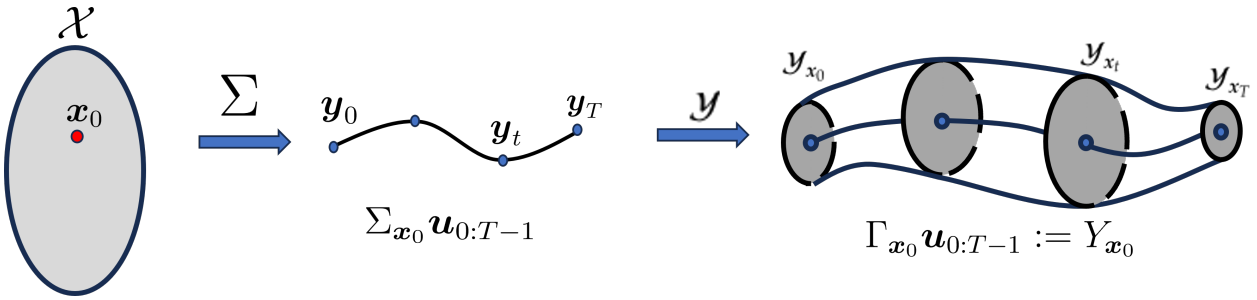
To "control" the quality of the estimation process along a planned state sequence $x_{0:T}$, a set-valued observability measure is defined in this section. For a LTI system with a Gaussian noise model, it is known that estimation can be improved by directly optimizing properties of the OG [7]. For set-valued measurements, an analogous setup is proposed in this work. In this direction, the notion of *distinguishability* is first examined [98].

Consider the system Eq. (3.1)-(3.2) with a given input sequence $u_{0:T-1}$ and an initial condition x_0. Here, the output sequence $y_{0:T}$ is defined as the image of the map Σ parameterized with a fixed input sequence for any initial condition. Thereby, define the initial condition to output sequence mapping as $\Sigma_{x_0}(u_{0:T}) = y_{0:T}$. We say that state $\bar{x}_0$ is *indistinguishable* from another state $x_0 \in \mathcal{X}$ if for every admissible input sequence, we have $\Sigma_{\bar{x}_0}(u_{0:T}) = \Sigma_{x_0}(u_{0:T})$. The system is said to be observable at $\bar{x}_0$, if $\bar{x}_0$ is distinguishable. *Distinguishability* ensures that for some input sequence and distinct initial conditions $x_0, \bar{x}_0 \in \mathcal{X}$, $\Sigma_{\bar{x}_0}(u_{0:T}) \neq \Sigma_{x_0}(u_{0:T})$, i.e., $d_s(\Sigma_{\bar{x}_0}(u_{0:T}), \Sigma_{x_0}(u_{0:T}) > 0$.

The notion of set separation is now extended to set-valued *output* sequences that will be referred to as "tubes." Such output sequences are characterized via the composition of the point-to-set map $\mathcal{Y}$ with Σ denoted by $\mathcal{Y} \circ \Sigma := \Gamma$. As such, Γ maps the initial condition x_0 to the output tube $\{\mathcal{Y}_{x_0}, \mathcal{Y}_{x_1}, \ldots, \mathcal{Y}_{x_T}\}$ for an input sequence $u_{0:T-1}$; in this chapter, the tube is a time-series of uncertainty sets denoted by the variable $Y_{x_{0:T}} = \{\mathcal{Y}_{x_0}, \mathcal{Y}_{x_1}, \ldots, \mathcal{Y}_{x_T}\}$.



(a) State to output uncertainty mapping



(b) Initial condition to output tube mapping

Figure 3.1: Representations of static and dynamics state to output maps for a given input sequence.

For a fixed initial condition, the mapping from initial condition to output tube is illustrated as follows:

$$x_0 \xrightarrow{\Sigma(u_{0:T-1})} \Sigma_{x_0} u_{0:T-1} \xrightarrow{\mathcal{Y}} \Gamma_{x_0}(u_{0:T-1}) := Y_{x_{0:T}} \qquad (3.18)$$

$\Gamma = \mathcal{Y} \circ \Sigma$

This mapping scheme is illustrated in Figure 3.1b. The first map (depicted on the left panel) generates an output sequence $\Sigma_{x_0} u_{0:T-1}$ for some $x_0 \in \mathcal{X}$. Then this sequence is converted to the tube sequence shown in Figure 3.1b (right panel), with ellipsoidal sets generated for each point y_t with the size dependent on x_t.

Now the notion of observability is defined such that, Γ is a one-to-one mapping from $\bar{x}_0$ to the output tube of length T; for brevity, $u_{0:T-1}$ will be dropped from the Γ map.

Definition 3.2.1. *The distance metric $d_\Gamma(.,.)$ on output tubes $Y_{\bar{x}_{0:T}}$ and $Y_{x_{0:T}}$ over the time interval $[0,T]$ $(T>0)$, generated by initial conditions $\bar{x}_0$ and x_0 for an input sequence, is defined as,*

$$d_\Gamma\left(Y_{\bar{x}_{0:T}}, Y_{x_{0:T}}\right) = \sum_{t=0}^{T} d_s\left(\mathcal{Y}_{\bar{x}_t}, \mathcal{Y}_{x_t}\right), \tag{3.19}$$

where the set-distance is given by Definition 3.1.1. [4]

The above notion of tube distance is analogous to that of sets; as such, it is also a pseudo-metric since $d_\Gamma\left(Y_{\bar{x}_{0:T}}, Y_{x_{0:T}}\right) = 0$ does not imply that tubes $Y_{\bar{x}_{0:T}}$ and $Y_{x_{0:T}}$ are identical. The function d_Γ is now used to define the following notion of observability.

Definition 3.2.2. *The system with a set-valued output is said to be finite horizon weakly observable at $\bar{x}_0$ over horizon $T>0$, if for all x in an open neighborhood U of $\bar{x}_0$, and an arbitrary input sequence, one has,*

$$d_\Gamma\left(Y_{\bar{x}_{0:T}}, Y_{x_{0:T}}\right) = \sum_{t=0}^{T} d_s\left(\mathcal{Y}_{\bar{x}_t}, \mathcal{Y}_{x_t}\right) > 0, \quad \textit{implies that} \quad d(\bar{x}_0, x_0) > 0; \tag{3.20}$$

as such, $d_\Gamma\left(Y_{\bar{x}_{0:T}}, Y_{x_{0:T}}\right) > 0$ implies that $\bar{x}_0$ and x_0 are distinguishable.

For tubes to be *separable*, at least one output-uncertainty pair $(\mathcal{Y}_{\bar{x}_t}, \mathcal{Y}_{x_t})$, generated using the same input sequence, should be *separable* for some t. In this case, Eq. (3.20) holds and trajectories $(Y_{\bar{x}_{0:T}}, Y_{x_{0:T}})$ are *distinct*, necessary for the validity of set-valued observability.

The term "weakly" Definition 3.2.2 suggests that the corresponding conditions are valid for $\bar{x}_0$ in a local neighborhood and not necessary for all $x \in \mathbb{X}$. This definition ensures that, for finite horizon observability $(T>0)$, the output tube generated by x_0 is distinct from those generated by neighboring initial states. Figure 3.2 illustrates this for x_0, depicting unobservable (left) and observable maps (right), respectively. In Figure 3.2, the output tubes are generated for $\bar{x}_0$ and a state x in the neighborhood of $\bar{x}_0$, with a fixed input sequence.

[4] Note that d_Γ is a pseudo-metric as $d_\Gamma\left(Y_{\bar{x}_{0:T}}, Y_{x_{0:T}}\right) = 0$ does not imply that the tubes $Y_{\bar{x}_{0:T}}$ and $Y_{x_{0:T}}$ are identical. In fact, the metric is only applicable for tubes where $Y_{\bar{x}_{0:T}} \bigcap Y_{x_{0:T}} \neq \emptyset \implies d_\gamma(.,.) > 0$.

For the case shown on the left panel of the figure, the output tubes are indistinguishable as the separation metric is zero; as such, $\bar{x}_0$ is not observable. On the contrary, the case shown on the right panel has a positive separation between the output tubes; as such, $\bar{x}_0$ is observable. An important aspect of set-valued observability of importance for path planning

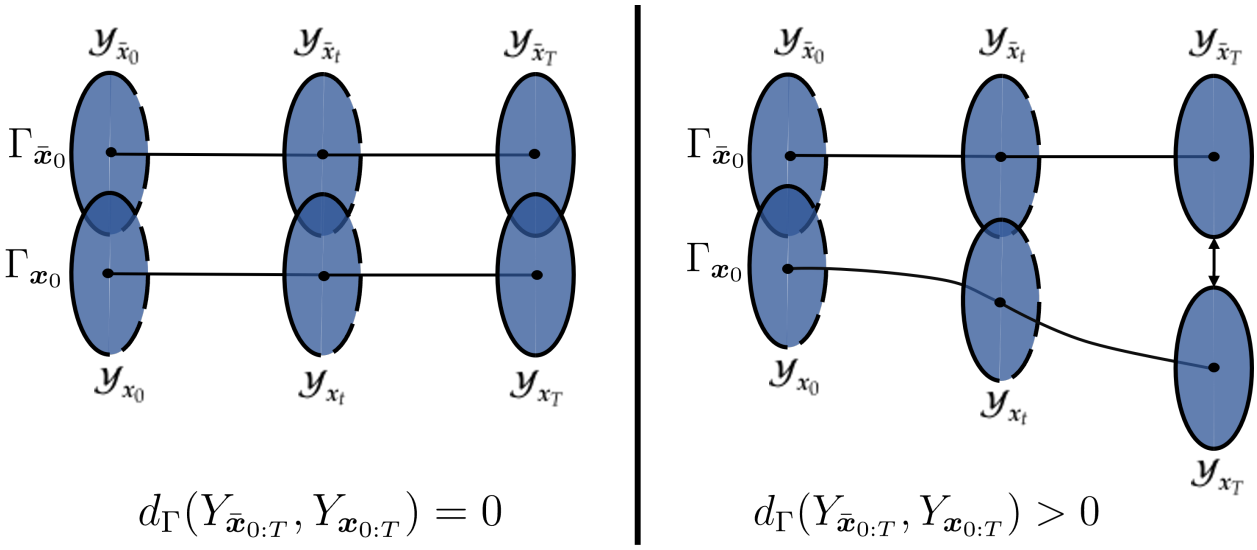


Figure 3.2: Representation of distinguishable and undistinguishable initial conditions using tube separation.

is the finiteness of the planned trajectory. Observability in general does not require a "short-horizon" distinguishability condition and T in Definition:3.2.2 can be arbitrarily large. A notion of short-time observability has been used for trajectory generation in deterministic output case by Alaeddini, Morgansen and Mesbahi [8]. Here, by solving to optimize for observability in the trajectory planning problem, the intention is to search for a map where states are observable within the planning horizon.

Therefore, an observability condition must be imposed for a fixed T, where any initial condition $x_0 \notin B_\delta(\bar{x}_0)$ is distinguishable from $\bar{x_0}$. Here, $B_\delta(\bar{x}_0)$ denotes a ball around $\bar{x}_0$. In order to optimize for observability, the conditioning notion for observer design is defined

for the set-valued analysis. A metric can then be defined to characterize observability, a term that will subsequently be embedded in an optimization objective for estimation-aware planning.

Degree of Observability

Now, consider an inverse map from the space of output tubes to the space of feasible state vectors, that is, $\Gamma^{-1} : Y \to \mathcal{X}$, where Y represents the space of tubes and $\mathcal{X}$ the space of feasible states. If such an inverse map exists, that is, when for every tube Y_x there exists a unique initial condition x, then conditions for finite horizon observability have been met. It is important to note here that Γ^{-1} represents the estimation map as it takes the measurement sequence to a state vector. The degree of observability using a regularity metric can now be defined using this inverse map [61, 77, 62].

Definition 3.2.3. *(Inverse mapping and Metric Regularity) Consider metric spaces $(\mathcal{X}, d)$ and (Y, d_Γ) and the set-valued mapping $\Gamma : \mathcal{X} \to Y$. Then Γ is metrically regular at $(\bar{x}, Y_{\bar{x}})\mathcal{X} \in \mathcal{X} \times Y$, if there exists constant $c > 0$ and neighborhoods U of $\bar{x}$ and V of $Y_{\bar{x}}$ such that:*

$$d(x, \Gamma^{-1}(Y_x)) \leq c\, d_\Gamma(Y_x, \Gamma(x)) \quad \forall\, (x, Y) \in (U \times V), \tag{3.21}$$

where $\Gamma^{-1}(Y_x) = \{x \in \mathcal{X} | Y_x \in Y\}$; the metrics d and d_Γ are defined on points and tubes, respectively, for sufficiently small neighborhoods U and V.

The constant c in Eq. (3.21) is called the modulus of regularity for the set map [77]. When Γ^{-1} exists, $c > 0$ is a system dependent constant.

Equation (3.21) ensures that a perturbation in the output tube is lower bounded by the tolerance on the corresponding initial state. For the state estimation process to be robust, one requires that the set of feasible states x be consistent with the output tube, and the respective distance in $\mathcal{X}$ scales favorably compared with perturbations in the output tube.

Lastly, given the tolerance region parameterized as $B_\epsilon(\bar{x}_0)$, it is desired that $\bar{x}_0$ be ϵ-distinguishable and robust to perturbations in the output tubes. Therefore, it is desired that

for a perturbed state $x_0 \in \mathcal{A} = \{x_0 | x_0 \notin B_\epsilon(\bar{x}_0)\}$, the perturbation in the output tube must be maximized. As such, a quantity for characterizing the worst case separation in output due to perturbation in the state is defined as the *degree of observability* [5], denoted by,

$$D_{\mathcal{O}}\left(Y_{\bar{x}_{0:T}}\right) = \inf_{x \in \mathcal{A}} d_\Gamma\left(Y_{\bar{x}_{0:T}}, Y_{x_{0:T}}\right); \tag{3.22}$$

as such, for initial states in $\mathcal{A}$ in Eq. (3.22) all trajectories are distinguishable from those initialized from $\bar{x}_0$ if this quantity is positive.[6] Here, estimation aware planning would involve maximizing $D_{\mathcal{O}}\left(Y_{x_{0:T}}\right)$ with respect to Γ, as this would result in decreasing the sensitivity of the estimator to large perturbations in measurements. In the next section, the problem of selecting input sequences to maximize observability will thus be examined.

3.3 Observability metric for discrete time output-tubes

As shown in Eq. (3.18), the nominal output map Γ_{x_0} is partially parametrized by the input sequence $u_{0:T}$; the corresponding value of $D_{\mathcal{O}}$ is computed by evaluating trajectories in the region around the nominal trajectory with the same input sequence. This nominal trajectory is denoted by the state/input pair sequences $(\bar{x}_{0:T}, \bar{u}_{0:T-1})$, for the system Eq. (3.1)-(3.2), in the absence of output uncertainties. The perturbed trajectory is generated by sampling initial condition from the aforementioned set $\mathcal{A}$ in the previous section.

The analysis for degree of observability requires computation of distances between output tubes, which are state-dependent. In order to compute these tubes, a primary deterministic output trajectory $\Sigma_{\bar{x}_0} u_{0:T-1}$ is generated, and then a second trajectory $\Sigma_{x_0} u_{0:T-1}$, generated by perturbing the initial state to x_0 such that $x_0 \in \mathcal{A}$. The state-dependent primary tube and the perturbed tube are then computed by applying the tube map Γ to $\bar{x}_0$ and x_0, respectively. These tubes are denoted by $Y_{\bar{x}_{0:T}}$ and $Y_{x_{0:T}}$.

---

[5] The notion of *degree of observability* comes from the definition of *modulus of regularity* [62] defined for the inverse map in Definition 3.2.3. This notion of regularity is often used to analyze perturbations on set-valued maps.

[6] For example, the set can be defined as $\mathcal{A} = \{x | \|x - x_0\| = \epsilon\}$.

In order to compute the deterministic trajectories, we define the perturbed dynamics around $(\bar{x}_t, \bar{u}_t)$, in the absence of uncertainties, using the Taylor series expansion,

$$\delta x_{t+1} = \left.\frac{\partial f}{\partial x}\right|_{x=\bar{x}_t} \delta x_t \quad + \left.\frac{\partial f}{\partial u}\right|_{u=\bar{u}_t} \delta u_t \quad + \mathcal{O}(\|\delta x_t\|^2, \|\delta u_t\|^2),$$
$$\delta y_t = \left.\frac{\partial h}{\partial x}\right|_{x=\bar{x}_t} \delta x_t,$$

where $\delta x_t = x_t - \bar{x}_t,\ \delta u_t = u_t - \bar{u}_t$ and $\delta y_t = y_t - \bar{y}_t$ are deviations from the nominal states, inputs and outputs during the time interval $t \in [0, T-1]$. It is assumed that when the state deviation $\|\delta x_t\|$ is small and $\|\delta u_t\|$ is bounded, then the higher-order terms $\mathcal{O}(\|.\|^2)$ are negligible. The linearized time-varying dynamics is then,

$$\delta x_{t+1} = A_t \delta x_t + B_t \delta u_t, \tag{3.23}$$
$$\delta y_t = C_t \delta x_t, \tag{3.24}$$

where $A_t = [\partial f/\partial x](x_t, u_t) \in \mathbb{R}^{n_x \times n_x},\ B_t = [\partial f/\partial u](x_t, u_t) \in \mathbb{R}^{n_x \times n_u}$ and $C_t = [\partial h/\partial x](x_t, u_t) \in \mathbb{R}^{n_x \times n_y}$ are the corresponding Jacobians in the Taylor series expansion. Here, a *trust region* must be defined such that higher-order terms are bounded and the local linearization is valid. As such, the maximum state deviation is restricted to a trust region where for a fixed control maximum deviation $\|\delta u\| \leq \Delta_u$,

$$\|\delta x\| \leq \Delta_x \implies \mathcal{O}(\|\delta x_t\|^2, \|\delta u_t\|^2) \leq \|P^{1/2} \delta x_t\|; \tag{3.25}$$

where $P \succ 0$ bounds maximum local state deviation and depends on the system. Note that it also naturally follows that ϵ must be selected such that $\|\delta x_0\| = \epsilon \leq \Delta_x$. A lower bound on the discrete-time degree of observability can be computed as defined in Eq. 3.22. Here x_t is the trajectory generated by sampling x_0 from $\mathcal{A}$ and propagating the dynamics with input sequence $u_t = \bar{u}_t \implies \delta u_t = 0$. As discussed previously, the initial perturbation is set up ϵ; thus, $\|\delta x_0\| = \epsilon$. The perturbed output trajectory can be computed for the dynamics Eq. (3.23-3.24) as $y_t = \bar{y}_t + C_t(A_{t-1} \dots A_1 A_0)\, \delta x_0$. With perturbations defined on deterministic dynamics Eq. (3.1-3.2), the tubes can be generated with the map Γ. If for such $x_0 \in \mathcal{A}$,

$d_\Gamma\left(Y_{\bar{x}_{0:T}}, Y_{x_{0:T}}\right) > 0$, then $\bar{x}_0$ is *finite horizon weakly observable*. A lower bound on the degree of observability, using Eq. (3.16) and Eq. (3.22), can then be computed, i.e.,

$$D_{\mathcal{O}}\left(Y_{\bar{x}_{0:T}}\right) = \inf_{x\in\mathcal{A}} \sum_{t=0}^{T} d_s\left(\mathcal{Y}_{\bar{x}_t}, \mathcal{Y}_{x_t}\right) \geq \sum_{t=0}^{T} \inf_{x\in\mathcal{A}} d_s\left(\mathcal{Y}_{\bar{x}_t}, \mathcal{Y}_{x_t}\right) \tag{3.26}$$

$$= \sum_{t=0}^{T} \inf_{x_0\in\mathcal{A}} \max\ \{\|\bar{y}_t - y_t\| - \Lambda(\mathcal{Y}_{\bar{x}_t}) - \Lambda(\mathcal{Y}_{x_t}),\ 0\} \tag{3.27}$$

$$\geq \sum_{t=0}^{T} \max\ \left\{\left(\inf_{x_0\in\mathcal{A}} \|C\delta\bar{x}_t\|\right) - \Lambda(\mathcal{Y}_{\bar{x}_t}) - \Lambda(\mathcal{Y}_{\bar{x}_t}) - L(\bar{x}_t, \delta x_t)\|\delta x_t\|,\ 0\right\} \tag{3.28}$$

$$\geq \sum_{t=0}^{T} \max\left\{\underbrace{s_1\epsilon}_{T1},\ \underbrace{-\ s_2\epsilon L(\bar{x}_t, s_2\epsilon) - 2\Lambda(\mathcal{Y}_{\bar{x}_t})}_{T2},\ 0\right\} := D^{\ell}_{\mathcal{O}}(Y_{\bar{x}_{0:T}}). \tag{3.29}$$

Here, $s_1(t) = \sigma_n(C_t A_{t-1}\dots A_0)$ and $s_2(t) = \sigma_1(A_{t-1}\dots A_0)$ and σ_1 and σ_n are the largest and smallest singular values of their respective matrix arguments.[7]

The parameter ϵ in the above analysis is selected such that, for some positive $t \leq T$, the right-hand side of Eq. (3.29) is positive. Note that the separation term $T1$ and the uncertainty term $T2$ are independent of each other. Moreover, the separation term $T1$ depends only on the initial deviation ϵ and the dynamics of the system. It is desired to have $T1$ be bounded, as the optimization problem involves maximizing this lower bound for observability. When A_t is not necessarily Schur stable, then $T1$ can grow exponentially, even for finite horizon problems, and scaling the augmented observability costs becomes an issue. Maximizing T_1 does improve observability as defined in Definition 3.3.1, but it implies that the linearized dynamics are unstable, which is undesirable for tracking. Therefore, exploration terms must be bounded. The term $T2$ which maps how uncertainty grows over the trajectory also needs to be bounded as with unbounded uncertainty, separability of trajectories, and consequently observability, becomes infeasible; hence, Assumption (3.1.1) is required for this analysis; the term $D^{\ell}_{\mathcal{O}}$ is now becomes well-defined. Note that for distinguishability, we need $D^{\ell}_{\mathcal{O}} > 0$. One can select a large enough ϵ such that the derived lower bound is positive.

[7]Here, the lower and upper bounds on the induced norm of matrices have been used.

Hence, for the optimization problem that will be posed shortly, the parameter ϵ acts as a tuning parameter that must be selected large enough for $D^{l}_{\mathcal{O}}$ to be positive for the initial trajectory. This is implied by stating that the system is observable for a particular map Γ for an initial trajectory $\bar{x}_{0:T}, \bar{u}_{0:T-1}$.

The lower bound $D^{\ell}_{\mathcal{O}}(Y_{\bar{x}_{0:T}})$ on the degree of observability has been defined as a function of the state trajectory $x_{0:T}$. Under the assumption on the output uncertainty, one obtains a system with set-valued outputs, defined here as *observability-regular*. The system is *finite horizon weakly observable* when $D^{\ell}_{\mathcal{O}}(Y_{\bar{x}_{0:T}}) > 0$.

Definition 3.3.1. *The system Eq. (3.1)-(3.2) with set-valued outputs defined in Eq. (3.5) is observability-regular if Assumption 3.1.1 holds and the system is finite horizon weakly observable.*

The main theorem for the observability measure is stated as follows:

Theorem 3.3.1. *Given a system with a finitely observable state $\bar{x}_0$, the lower bound on the degree of observability is a concave function with respect to the state sequence $x_{0:T}$ if the output map is observability-regular.*

Proof: Given x_0 and an observability-regular uncertainty map, one can compute the lower bound on the degree of observability as given by Eq. (3.29). Here Λ, and consequently $L(\bar{x})$, are convex. Thus, Eq. (3.29) is a summation of concave functions and, as such, concave.

Note that this proof is structured on the condition that the uncertainty map is convex with respect to the state or has an upper-bounding envelope that is convex. Now, one can use this metric as a cost function in an optimal trajectory design problem. One notes that it is in fact the lower bound Eq. (3.29) that will be used in the guidance objective that, when maximized, improves the observability of the trajectory. Since the T1 term is constant within the trust region, maximizing the degree of observability with respect to the trajectory, only increases the tube separation by optimizing over the convex function Λ. For the optimization problem, max function in Eq. (3.29), can be substituted out and objective for maximizing

observability can be written as,

$$\widehat{D}^{\ell}_{\mathcal{O}}(Y_{\bar{x}_{0:T}}) = \sum_{t=0}^{T} s_1\epsilon - s_2\epsilon L(\bar{x}_t, s_2\epsilon) - 2\Lambda(\mathcal{Y}_{\bar{x}_t}), \tag{3.30}$$

where $s_1 = (\sigma_n(C_tA_{t-1}, \dots A_0))$ and $s_2 = \sigma_1(A_{t-1}, \dots A_0)$ are constants computed over the parameters of the nominal trajectory. Here, the convexity of Λ and L with respect to $\bar{x}_t$ ensures that when initialized with an observable trajectory (with $D^{\ell}_{\mathcal{O}}(Y_{\bar{x}_{0:T}}) > 0$), the updated trajectory is "more" observable. Due to the convexity assumption, the optimization step at iteration k from $\bar{x}^k_{0:T}$ to $\bar{x}^{k+1}_{0:T}$ guarantees that $\widehat{D}^{\ell}_{\mathcal{O}}(Y_{x^{k+1}_{0:T}}) \geq \widehat{D}^{\ell}_{\mathcal{O}}(Y_{\bar{x}^k_{0:T}}) > 0$.

In the above analysis, the function $\Lambda(\mathcal{Y}_x)$ is derived from the sensor model and L can be computed using Eq. (3.8). If the analytical form of $\Lambda(\mathcal{Y}_x)$ is known, then as stated in Assumption 3.1.1, the convexity of these functions is a requirement for the existence of a unique trajectory; as such, the user must approximate the necessary bounds via a convex envelope function. For a special case, when Λ is defined by a quadratic form x^TQx for some positive definite Q, the objective can be computed by defining a lower bound on Eq. (3.27). In this case, one can write,

$$\begin{aligned}
D_{\mathcal{O}}\left(Y_{\bar{x}_{0:T}}\right) &= \sum_{t=0}^{T} \inf_{x_0 \in \mathcal{A}} \left\{\|\bar{y}_t - y_t\| - \Lambda(\mathcal{Y}_{\bar{x}_t}) - \Lambda(\mathcal{Y}_{x_t})\right\} \geq 0 \\
&\geq \sum_{t=0}^{T} \left\{s_1\epsilon - \Lambda(\mathcal{Y}_{\bar{x}_t}) - \Lambda(\mathcal{Y}_{x_t})\right\} \\
&= \sum_{t=0}^{T} \left\{s_1\epsilon - \bar{x}_t^{\top}Q\bar{x}_t - (\bar{x}_t + \delta x_t)^{\top}Q(\bar{x}_t + \delta x_t)\right\} \\
&= \sum_{t=0}^{T} \left\{s_1\epsilon - 2\bar{x}_t^{\top}Q\bar{x}_t - \sigma_1(Q)s_2\epsilon - 2\|x_t\|\sqrt{s_2\epsilon})\right\} \\
&= \sum_{t=0}^{T} \left\{(s_1\epsilon - \sigma_1(Q)s_2\epsilon) - 2\bar{x}_t^{\top}Q\bar{x}_t - 2\|x_t\|\sqrt{s_2\epsilon})\right\}.
\end{aligned} \tag{3.31}$$

When the enveloping function that approximates the upper bound on the ellipsoid size is a quadratic function on x_t, then the lower bound on the degree of observability can be approximated with Eq. (3.31) and used as the objective function.

It is also important to note, as stated earlier, the observability-maximizing trajectory could lead to a solution that is "unstable" for an infinite horizon problem. For the finite horizon setting, maximizing this metric will lead to a trajectory that seeks optimal state estimation, but in applications, this trajectory might conflict with task-completion constraints, such as reaching a desired final state. The observability maximization seeks to minimize uncertainty and naturally will prescribe states corresponding to low uncertainty. Therefore, an estimation-aware optimization problem is presented as a tool to generate a deviated trajectory with respect to some nominal trajectory that achieves the primary task. The observability metric can then be maximized under a constraint that limits the deviation from this nominal trajectory. Alternatively, one can augment the observability metric, with an existing trajectory design objective, in order to plan an estimation-aware trajectory.

3.4 Remarks on Set-valued Observability and deterministic planning approach

In the proposed approach Sections 3.2-3.3, we discussed set-valued observability and a metric for designing a subsequent planning approach for set-valued uncertainty. We used bounded sets and the separation measure to design the trajectory distinguishability conditions under the worst-case uncertainties. This approach is geared towards a setup where a ML-based sensor is being used and its uncertainty distribution is unknown or is non-Gaussian. Moreover, the uncertainty has state-dependent characteristics. Particularly in this work, we explore how observability can be evaluated under such conditions if priors of set sizes and state-dependent characteristics are known, as defined in Assumption 3.1.1. The metric defined as degree of observability is a function of output maps for a given state, such that its maxima corresponds to the trajectory a with optimum estimation. Such a metric can be computed for any type of uncertainty where we can define a distance function that can separate output uncertainties corresponding to distinct states.

Note: The Theorem 3.3.1 can be restated for any separation metric and non-set-valued uncertainty type such that Definition 4.6.1 and Assumption 3.1.1 holds.

To further understand the estimation-aware planning approach, let us look at a discrete-

time linear system with Gaussian uncertainty and draw connections to the general set-valued approach.

3.4.1 Connection to Sensor Design problem for Linear systems

Conventionally, noise models are assumed to have Gaussian distributions and to be uncorrelated with the state and output vectors over time. We want to quantify the difference Now, consider the concept of observability, which, in principle, is the invertibility condition on the mapping from the state to the output sequence [16]. Given any output sequence, observability ensures that it was generated by propagating a unique state vector. If the output sequence is perturbed, then—because the system is observable—this perturbation can also be detected in the state vector.

Let us formally define observability for the case of discrete-time linear systems. Consider the mapping $\mathcal{M}_T : \mathcal{X} \to \mathcal{Y}$, where this linear operator maps an initial condition to an output trajectory over a time horizon T:

$$\mathcal{M}_T(x_0) = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(T) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{T-1} \end{bmatrix} x_0.$$

If this map is injective, then any output in the range of $\mathcal{M}_T$ corresponds to a unique x_0. To characterize its injectivity, we compute the adjoint operator $\mathcal{M}_T^* : \mathcal{Y}^* \to \mathcal{X}^*$ defined by

$$\mathcal{M}_T^*(\sigma()) = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{T-1} \end{bmatrix}^* \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(T) \end{bmatrix}$$

for $\sigma() \in \mathcal{Y}^*$. The map $\mathcal{M}_T$ is injective if its rank equals the dimension of $\mathcal{X}$ (i.e., the state dimension). The rank of this operator is equal to the rank of $\mathcal{M}_T^* \mathcal{M}_T$. In the discrete-time

case, this operator is given by

$$W_o = \sum_{t=0}^{T} (A^t)^\top C^\top C A^t. \tag{3.32}$$

.

Given an output sequence $y() \in \mathcal{Y}$ over the interval $[0, T]$, the initial state can be computed from W_o. Assuming no input, the system is $y(t) = CA^t x_0$ and the corresponding x_0 for the given output sequence is

$$x_0 = \mathcal{M}^\dagger y() = (\mathcal{M}_T^* \mathcal{M}_T)^{-1} \mathcal{M}_T^* y() = W_o^{-1} \begin{bmatrix} C^\top & A^\top C^\top & \dots & A^{T-1\top} C^\top \end{bmatrix} \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(T) \end{bmatrix} \tag{3.33}$$

Now with noisy measurements $\hat{y}(t) = CA^t x_0 + v_t$ with $v_t \in \mathcal{N}(0, \Sigma_y),\ \Sigma_y = \sigma I$ we design an optimal estimator for x_0 by solving:

$$\begin{aligned} \min_{x_0} \quad & \|v_t\|_{L_2}^2 \\ \text{s.t} \quad & \hat{y}(t) = CA^t x_0 + v_t, \quad t = 0, \dots, T \end{aligned}$$

The least-squares approach yields the optimal solution given in Eq. (3.33), where the least-squares solution corresponds to the pseudo-inverse of the operator $\mathcal{M}_T$ when the noise is zero-mean. The state covariance can then be computed as

$$E[(\hat{x}_0 - Ex_0)(\hat{x}_0 - Ex_0)^T] = \Sigma_x = \sigma W_o^{-1} \tag{3.34}$$

The inverse observability Gramian W_o^{-1} maps the output covariance to the state covariance. The map is called *metrically regular* if W_o^{-1} is non-singular [77]. The *modulus of regularity* is defined as the operator norm of W_o^{-1}, representing the scaling from measurement perturbations to state perturbations. As the trajectory length $T \to \infty$, W_o^{-1} converges to the true observability Gramian [111].

The matrix W_o^{-1} thus serves as a design matrix that scales state covariance. For a linear system with fixed noise characteristics Σ_y, the state covariance Σ_x quantifies the uncertainty volume. Since W_o^{-1} depends only on the system (A, B, C), for a system with static noise, the estimation quality[8] depends solely on the system's observability metrics.

For the static noise case, Eq. 3.34 provides the best estimate in the least-squares sense. If the noise characteristics are non-static, weighted least squares can be applied [2]. For example, when a confidence metric is available for each output, a proportional weighting matrix can scale the measurement covariances accordingly. Alternatively, if sensor-dependent noise exists and sensor selection is possible, one can pose a sensor-selection problem to choose a design matrix that optimizes a property such as the modulus of regularity of W_o^{-1} [231].

To further understand the significance of the modulus of regularity, let us consider a discrete-time linear system with measurement noise given by:

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k + v_k$$

The degree of observability as defined in Eq. (3.22) is a conservative measure of tube separation which guarantees the distinguishability of any realization of a trajectory. So for this setup, let us fix a finite horizon trajectory for time $[0, T]$, by selecting an input sequence $u_{0:T-1}$. Also consider output uncertainty v_k sampled from a ball around Cx_k of a constant radius r with an unknown distribution. This defines a constant uncertainty model without state dependence. The output set is represented as $\mathcal{Y}_k = Cx_k \oplus \mathcal{B}_r(Cx_k)$ which is a Minkowski sum. We point the reader to the Section 3.4.3, on how to convert a Gaussian model to a bounded sphere. Now, consider initial conditions x_0 and any state $x_0' = x_0 + \epsilon$. For simplicity, set $u_{0:T-1} = 0$ and propagate both the state trajectories for x_0 and x_0' and compute the conservative lower bound on degree of observability for trajectory $x_{0:T}$ using the setup defined in Eq.(3.31). As shown in the previous section, we can guarantee

[8]Represented by any function of Σ_x.

observability if $D_O(x_0) > 0$ ie any realization of an output trajectory generated from x_0' can be separated from any realization of the nominal output trajectory $y_{0:T}$. This implies that x_0' can be distinguished from x_0. Using Definition 3.3.1 we can show observability by computing $d_\Gamma(Y_{x_0}, Y_{x_0'})$:

$$d_\Gamma(Y_{x_0}, Y_{x_0'}) = \sum_{t=0}^{T} d_s(\mathcal{Y}_{x_t}, \mathcal{Y}_{x_t'}) \tag{3.35}$$

$$\geq \sum_{t=0}^{T} \max\{\|C(x_t - x_t')\| - 2r, 0\} \tag{3.36}$$

Here, positivity of Eq. 3.36 guarantees the observability under a set-valued model. In contrast, the observability computation by analyzing the rank of W_O defined in Eq.(3.32) or validating its invertability, does not guarantee observability as output uncertainty size is not captured in W_O. In our approach, we compute a lower bound on Eq.(3.36) as:

$$d_\Gamma(Y_{x_0}, Y_{x_0'}) \geq \sum_{t=0}^{T} \max\{\|C(x_t - x_t')\| - 2r, 0\} \tag{3.37}$$

$$\geq \sum_{t=0}^{T} \max\{\|CA^t\epsilon\| - 2r, 0\} \tag{3.38}$$

$$\geq \sum_{t=0}^{T} \max\{\sigma_n(CA^t)\epsilon - 2r, 0\} \tag{3.39}$$

where $\sigma_n()$ is the smallest eigenvalue of its argument. This lower bound is then used as the regularity metric as compared to the trace or eigenvalue of W_O as used in conventional approaches [7]. The Eq(3.40) is equivalent to scaling property of W_O with the compensation for set size r. Further, if state dependence of output uncertainty sets is considered, we need to compensate the set size by considering state dependent size,

$$d_\Gamma(Y_{x_0}, Y_{x_0'}) \geq \sum_{t=0}^{T} \max\{\sigma_n(CA^t)\epsilon - 2r(x), 0\}. \tag{3.40}$$

On use of observability for planning

In planning problems, paths are typically designed for agents with online estimators, such as a Kalman filter, which use prior estimates without adaptive algorithms for changing covariances. In these widely applicable robotics and navigation scenarios, we aim for agents to traverse paths with higher estimation accuracy. To achieve this, the estimation-aware framework solves the following optimization problem with a general system transition $\phi()$:

$$\begin{aligned} \min_{x_{0:T}} \quad & f(W_o^{-1}[y_{0:T}]) \\ \text{s.t.} \quad & x_{t+1} = \phi(x_t) \quad t = 0, \ldots, T-1 \\ & y_t = h(x_t) \quad t = 0, \ldots, T \end{aligned} \tag{3.41}$$

Here, $f(W_o^{-1}[y_{0:T}])$ evaluates the modulus of regularity for the output trajectory $y_{0:T}$. Some approaches compute an empirical observability Gramian over the trajectory and use a property such as the operator norm for $f()$ [7]. If $f()$ is convex with respect to the state sequence $x_{0:T}$, an estimation-aware path can be directly designed; otherwise, if $f()$ is differentiable and locally smooth, iterative algorithms can be applied to solve Problem 3.41.

3.4.2 Noise characteristics and effect of Observability

The observability metric $f()$, which corresponds to the operator norm of the Gramian in the linear case, must now be adapted to account for the true outputs affected by v_t. In most approaches, $f()$ can be computed over the output trajectory $y_{0:T}$ without explicitly considering noise, provided that the state vector captures the properties of uncertainty variation and magnitude. Since the distinguishability of the initial condition depends on the distinguishability of outputs, we require a method to compute the modulus of regularity while accounting for state-dependent noise.

Our approach in Section 3.5 uses an oracle that defines the state-dependent uncertainty. We evaluate the regularity modulus by assessing the minimum separation of outputs for

a given perturbation of the initial state. Specifically, for a perturbation x_0 around the principal state $\bar{x}_0$, with $d(\bar{x}_0, x_0) = \epsilon$, we define the modulus of regularity as $d_\Gamma(\bar{\mathcal{Y}}_{0:T}, \mathcal{Y}_{0:T})$. which represents the minimum perturbation in the propagated output trajectory sets for all x_0 around $\bar{x}_0$ under a given input sequence. By using d_Γ, which forms a valid pseudo-metric on separated sets[9], we convert a set-based representation into a point-wise measure.

This separation can be computed using any valid distance metric on output trajectories. Importantly, $d_\Gamma()$ does not require sets to be fully separated in the Euclidean sense. Alternative measures, such as KL divergence, can also be used to differentiate distinct uncertainties and evaluate the observability of a trajectory as defined in Definition 3.2.2.

Furthermore, any uncertainty model and distance metric can be used within the estimation-aware planning framework, provided that the variation rules in Assumptions 3.1.1[10] are satisfied and Definition 3.3.1 holds for the system. Under these conditions, Theorem 3.3.1 guarantees that the estimation-aware planning problem can be solved.

3.4.3 Using Gaussian uncertainty models

In this work, we have considered systems where noise is a bounded set with unknown distribution. Equivalently, a gaussian model can be used to fit the uncertainty. Alternatively, purely gaussian noise models with state-dependence can also be used in this setup. The gaussian uncertainty can be written as sets with the bound $d > 0$ that can be any user defined value that defines the set membership. Here, for gaussian uncertainty $y \sim \mathcal{N}(\mu, \Sigma)$ is a set of points at d distance from mean given by: $\mathcal{B} = \{y \in \mathbb{R}^n | (y-\mu)^\top \Sigma^{-1} (y-\mu) \leq d\}$. We can use concentration inequalities and define a probability that a given realization lies within this set using concentration inequality on norm of random variable y[11]. Using set-membership and concentration inequalities on random variables we convert the stochastic uncertainties

[9] d_Γ is positive if and only if the argument sets do not intersect.

[10] We use the largest eigenvalue in this assumption to compute the set radius. Similarly, set size can be computed depending on the desired representation, or generically as $\sup_{x,y \in \mathcal{A}} d(x, y)$.

[11] Refer Theorem 3.1.1(Concentration of Norms) [218]

to set-valued counterparts with a given probability. The observability can be evaluated with that given confidence measure using distinguishability metric defined in this chapter. We can then define the proposed planning problem as long as the sets follow the variation rules.

3.5 Finite horizon Estimation-Aware Trajectory Planning

Offline trajectories generated for vehicular systems are often in the form of waypoints that are tracked via a lower-level feedback controller. Generally, this low-level controller has a time-constant that is smaller than timescales inherent in the planning trajectory generation. [12] The internal controllers used for tracking depend on state observers. It is assumed that these observers are designed without knowledge of the statistical properties of the measurement uncertainties. For the deterministic case, the state observers are guaranteed to converge to the "true" state when the system is observable. For the state-dependent and set-based uncertainties that considered in this chapter, these state observers will generally have sub-optimal estimation performance and the system may not even be observable. Therefore, in order to optimize the tracking performance for these systems, the estimation-aware trajectory design is formulated such that the lower bound on the degree of observability is optimized.

In this section, two scenarios for generating estimation-aware trajectories are discussed. In the first case, it is assumed that a nominal trajectory $\bar{x}_{0:T}$ has been defined and user requires a improvement in estimation. In this setting, the estimation-aware trajectory are realized via deviations to $\bar{x}_{0:T}$, while satisfying the hard constraints of the original task. In the second case, the task and estimation-aware design problem are solved simultaneously as a solution to an optimization problem. In both scenarios, the main ingredient of the formulation involves maximizing Eq. (3.29).

[12] The open loop convergence time for low level controllers is referred to as the time constant of a control system. The time constant for the feedback control ensures that the tracking process does not incur a large lag due to control implementation.

3.5.1 Maximizing observability for a pre-defined nominal trajectory

Consider the system described in Eq. (3.1)-(3.2), where the linearization has been given by Eq. (3.23)-(3.24). The approximation error for this setting is negligible for the trust region defined by Eq. (3.25). Consider a nominal trajectory and control for achieving a certain task has been given in the form of $\bar{x}_{0:T}, \bar{u}_{0:T-1}$. One then explore deviations from the nominal trajectory that maximizes observability while following strict constraints specified by the primary task. It is assumed that the enveloping function for the size of the uncertainty, $\Lambda(\mathcal{Y}_x)$, has been defined for this scenario. In practice, the state-dependent envelope is either known due to knowledge of the sensor's characteristics or estimated and stored as an oracle. Subsequently, depending on the mission constraints, an exploration parameter $\gamma > 0$ must be defined that bounds the total deviation from the nominal trajectory. Hence, one can add an exploration constraint $\|x_t - \bar{x}_t\|_2 \leq \gamma,\ \forall t \in [0, T]$, thereby bounding the total deviation due to exploration. [13]

The estimation-aware planning problem for the given nominal trajectory can now be defined as **Problem 1**:

$$\begin{aligned} \min_{u_{0:T-1}} \quad & -D^{\ell}_{\mathcal{O}}(Y_{x_{0:T}}) \\ \text{s.t.} \quad & x_{t+1} = f(x_t, u_t) \quad \text{for all } t \in [0, T-1], \\ & \|x_t - \bar{x}_t\| \leq \gamma \quad \text{for all } t \in [0, T-1], \\ & x_0 = \bar{x}_0, \\ & x_T = \bar{x}_T. \end{aligned} \tag{3.42}$$

Here, the objective has been constructed as a convex function of the state sequence as defined in Eq. (3.30), but the dynamics are generally nonlinear, making the overall problem non-convex. A trust region-based sequential programming approach is now used to iteratively

[13] The norms $\{1, 2, \infty\}$ can also be chosen depending on the application. In this scenario, the aim is to bound the overall error with respect to the infinity norm.

construct and solve a sequence of convex sub-problems. The trust region algorithm used here is a variation on the Basic Trust Region (BTR) algorithm defined in [41]. [14] As such, the main problem is converted to a sequence of convex sub-problems by linearizing the dynamics and solving for optimal deviation from the nominal trajectory. A trust region is constructed to restrict the optimization solution to be within a region around the nominal trajectory where linearization error is minimal. The trust region algorithm has been described in Algorithm 2, where the exploration constraint is used as a termination condition. When the exploration condition is reached for an iteration sub-problem, the trust region is reduced and the subproblem is re-instantiated. This step is repeated until trajectory updates between the iterations become negligible.

The trust regions are derived from bounds on linear approximations of the dynamics as defined in Eq. (3.25).For linear approximation of dynamics Eq. (3.23-3.24), the user defines the initial trust region bound Δ_x^0 such that bound defined in Eq. (3.25) is satisfied. Here, for the k^{th} iteration, one has the linear approximation defined as $\delta x_{t+1}^k = A_t^k \delta x_t^k + B_t^k \delta u_t^k,\ \forall\, t \in [0, T-1]$, for $\delta x^k = x_t^k - x_t^{k-1}$ and $\delta u_t^k = u_t^k - u_t^{k-1}$. At the k^{th} iteration, the linearization is performed around the sequence $x_{0:T}^{k-1}, u_{0:T-1}^{k-1}$, and the subproblem is defined as **Problem 2** below,

$$
\begin{aligned}
\min_{\delta u_{0:T-1}^k} \quad & -D_{\mathcal{O}}^{\ell}(Y_{x_{0:T}}^k) \\
\text{s.t.} \quad & \delta x_{t+1}^k = A_t^k \delta x_t^k + B_t^k \delta u_t^k, \quad \text{for all } t \in [0, T-1], \\
& \|\delta x_t^k\|_\infty \leq \Delta_x^k \quad \text{for all } i \in [0, T-1], \\
& \delta x_0^k = 0, \\
& \delta x_T^k = 0.
\end{aligned} \tag{3.43}
$$

The state constraints above follow directly from **Problem 1**, ensuring that the initial conditions and final goal for the estimation-aware trajectory remain the same as that of the

[14] For problems with non-convex inequality constraints, one can use for example, successive convexification algorithm described in [136].

nominal trajectory. The exploration constraint shown in **Problem 1** Eq. (3.42) sets the termination criterion for the iterative algorithm as defined in Section 3.5.1.

In order to make the problem formulation more concrete, consider the so-called nonlinear unicycle model; in fact, the model is a Dubins car with a double integrator system rather than a conventional single integrator model [19]; in order to streamline the presentation, no turning-radius constraints are enforced. Specifically, the nonlinear dynamic model is defined as,

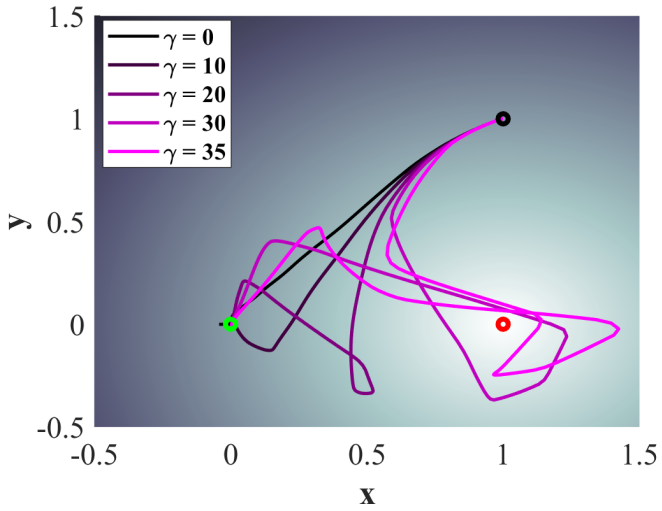
$$\begin{aligned}\dot{v}_x &= u_1 \cos\theta,\\ \dot{v}_y &= u_1 \sin\theta,\\ \dot{\omega} &= u_2,\end{aligned}$$

where v_x, v_y denote the linear velocity in the x, y directions, and θ is the vehicle's orientation with respect to the world frame. The derivatives $\dot{v}_x, \dot{v}_y$ are the acceleration dynamics of the agent driven by the forward thrust u_1 and the turning torque u_2. The output is measured as the position of the vehicle with a state-dependent uncertainty. The description of the vehicle dynamics for the estimation problem and linearized discrete time dynamics are discussed in Section 3.5.1 where the system is converted into the form Eq. (3.23)-(3.24) for the vehicle state, control input, and output given as (x, u, y). The output for the discretized system is uniformly sampled from the uncertainty set such that $y \sim U(\mathcal{Y}_x)$, where $\mathcal{Y}_x = \{y|\ Cx + K\|Cx - y_s\|_2^2 + r_0\}$. This candidate function has been chosen to emulate sensors that are state dependent but their uncertainty does not change arbitrarily with respect to the state. In more general cases, only a uniform maximum bound might properly characterize the uncertainty set. Instead, here the focus is on cases where the sensor state dependence property has been predetermined by the user and its upper bound can be approximated as a convex function. One notes that this is a valid assumption for physical systems where this model abstracts the local behavior of state-dependent sensors. For example, sensors that operate based on range-type measurements such as LiDAR often exhibit this behavior [228]. The purpose of using the candidate model is to show how to generate estimation-aware trajectories. As

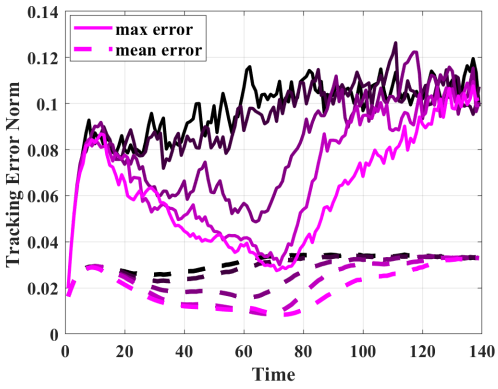
such, the presented methodology can be realized in applications where the state dependency of the sensor's error is set-theoretically well defined.

For the uncertainty set in subsequent discussion, y_s denotes the location of the light source and r_0 is the residual error that is independent of the state. This error profile is represented by the color gradient in the background of Figure 3.3a, where the light source is shown by the red circle and the darker regions represent higher output uncertainty at the location. This measurement model has been chosen for the problem considered not only for its practical relevance, but also as it captures a state-dependent uncertainty whose size $\Lambda(\mathcal{Y}_x)$ has a convex variation with respect to the state. The maximum size of the measurement uncertainty is characterized by a linear function of the agent's distance from the light source, making $\Lambda(x)$ convex. Since for this case Assumption 3.1.1 holds, the output map is *observability-regular* due to Theorem 3.3.1. The nominal trajectory for **Problem 1** is obtained by solving the primary objective where the agent/vehicle starts at position $(1, 1)$ and reaches the goal at $(0, 0)$. The nominal trajectory $x_{0:T}, u_{0:T-1}$ has been characterized in order to optimize the objective $J(x, u) = (x_T - x_r)^\top Q(x_T - x_r) + \sum_{t=0}^{T-1}[(x_t - x_r)^\top Q(x_t - x_r) + u_t^\top R u_t]$, where Q, R are positive semidefinite and positive definite matrices, respectively. Note that the nominal trajectory can be generated through other procedures. **Problem 2** defined in Eq. (3.43) is now solved using the BTR algorithm. In addition to ensuring the boundary conditions for the nominal trajectory generation, in the estimation-aware trajectory generation, input saturation and state constraints are included in **Problem 2**. The exploration parameter is a design parameter for the planning problem.

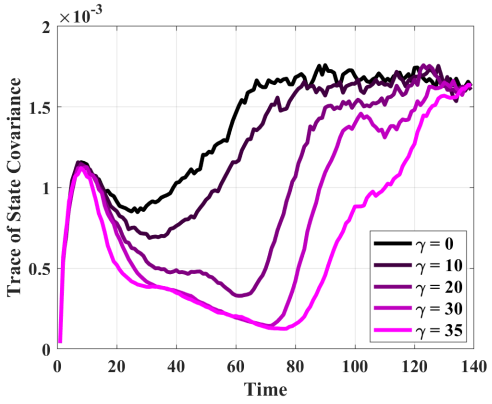
The resulting estimation-aware trajectories and the estimation performance statistics are shown in Figure 3.3. Figure 3.3a depicts the results of the estimation-aware optimization as a function of the exploration parameter γ compared with the nominal trajectory. The nominal trajectory starts at location $(1, 1)$ with the initial vehicle orientation of -π(facing -x axis). The trajectory is designed such that vehicle reaches the goal $(0, 0)$ at $-\pi$ orientation. This trajectory is shown in black and also corresponds to the solution of the estimation-aware planning when no exploration is allowed, i.e., when $\gamma = 0$. Here, the trajectories



(a) Nominal and estimation-aware trajectory comparison starting at **(1,1)** ending at **(0,0)** with light source at **(1,0)**



(b) Comparing the state error while tracking nominal and estimation-aware trajectories over time over 10000 roll-outs.



(c) Comparing the state variance across estimation-aware trajectories over time for 10000 roll-outs.

Figure 3.3: Comparative Analysis for Dubins car example with nominal trajectory and estimation-aware trajectories for increasing exploration bounds are presented.

with higher exploration constants are represented with a lighter shade of pink. Figure 3.3a shows that in order to optimize for observability, the solutions for trajectories with $\gamma > 0$ "explore" regions with lower output uncertainty. The intended optimized trajectory does not prioritize optimizing for reducing output uncertainty but instead, optimizes for state uncertainty. The degree of observability metric as defined in Eq. (3.22), is designed such that the size of the uncertainty for the state estimation process generated by an observer over this trajectory is reduced. In order to compute the state estimation performance over the trajectories, assume that a simple lower-level state-feedback controller for the Dubins car agent exists. Since the focus of this work is on the offline trajectory design, the design of a low-level controller for trajectory tracking is not discussed, as it is also system dependent. Nevertheless, the controller is designed to closely follow the waypoints with a fast convergence rate compared with the discretization time used in the system Eq. (3.23)-(3.24). The agent is equipped with a Lurenberger observer K_t, such that a state estimate can be derived from the output measurements. [15] The existence of this observer justifies the assumption that an inverse map Γ^{-1} exists. Now, the state of the online agent can be represented as $\hat{x}_{t+1} = (A_t - K_t C_t)\hat{x}_t + B_t u_t + K_t y_t$, with the output uniformly drawn from the uncertainty set as $y_t \sim U(\mathcal{Y}_{x_t})$. [16] The agent now follows a planned trajectory with the stated waypoints and experiences errors due to the corresponding estimation uncertainties. The tracking error norm $\|e_t\| = \|x_t - \bar{x}_t\|$, over the trajectory, is computed as a metric for tracking efficiency. For analysis of performance around each trajectory, a Monte Carlo simulation for $N = 10,000$ trajectories (roll outs) with the same initialization and the sensor model for each case of γ shown in Figure 3.3a. The measurements are drawn uniformly from the set defined by $\mathcal{Y}_x$ for each roll-out. Figure 3.3b shows the error norm over time and the maximum error norm over time over N roll-outs for each γ. The maximum error is shown by solid lines and the average error is shown by dashed lines corresponding to each estimation-aware

[15] Here a Lurenberger observer is designed for the linear approximations around the planned trajectory [9]; the corresponding poles are placed such that the observer has a fast error convergence.

[16] Other estimator design approaches can be used with the assumption that the system, without state-dependent uncertainties, remains observable.

trajectory shown with shades of pink. Observe that the expected total error (integral of tracking error over time) is reduced with the estimation-aware trajectories and the worst-case state estimation errors are also reduced over time. Figure 3.3c depicts the trace of the state covariance ($E[(\hat{x} - E\hat{x})(\hat{x} - E\hat{x})^\top]$) over N roll-outs as a function of time. The trace of the variance represents sum of the deviations across all states, in this case, position, orientation and velocities. The total variance shows a significant improvement for scenarios where a higher level of exploration has been permitted. The trade-off for improving the estimation performance is an increased deviation from the nominal task-based trajectory and a possibility of reaching the boundary of the state and control constrained sets.

System dynamics for Dubins Car

The nonlinear full state Dubins car dynamics is defined as,

$$\begin{aligned}
\dot{p_x} &= v_x, \\
\dot{p_y} &= v_y, \\
\dot{\theta} &= \omega, \\
\dot{v_x} &= u_1 \cos\theta, \\
\dot{v_y} &= u_1 \sin\theta, \\
\dot{\omega} &= u_2,
\end{aligned}$$

where the agent's state is defined as $x = [p_x, p_y, \theta, \dot{p_x}, \dot{p_y}, \dot{\theta}]^\top$ and inputs for the forward thrust and turning torque are designated as $u = [u_1, u_2]^\top$, where p_x, p_y are the agent's position coordinates in 2D and θ is the heading angle in the world frame. The corresponding velocities are given by v_x and v_y and the angular rate by ω. The output is measured by a position $y = [p_x, p_y]^\top$ sensor that has an uncertainty dependent on the light intensity. The discretized dynamics at step dt are represented as Eq. (3.23)-(3.24), where the system

matrices are,

$$A(x,u) = \begin{bmatrix} 1 & 0 & -\frac{dt^2 u_1 \sin\theta}{2} & dt & 0 & -\frac{dt^3 u_1 \sin\theta}{6} \\ 0 & 1 & \frac{dt^2 u_1 \cos\theta}{2} & 0 & dt & \frac{dt^3 u_1 \cos\theta}{6} \\ 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & -dt u_1 \sin\theta & 1 & 0 & -\frac{dt^2 u_1 \sin\theta}{2} \\ 0 & 0 & dt u_1 \cos\theta & 0 & 1 & \frac{dt^2 u_1 \cos\theta}{2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \; B(x,u) = \begin{bmatrix} \frac{dt^2 \cos\theta}{2} & -\frac{dt^4 u_1 \sin\theta}{24} \\ \frac{dt^2 \sin\theta}{2} & \frac{dt^4 u_1 \cos\theta}{24} \\ 0 & \frac{dt^2}{6} \\ dt\cos\theta & -\frac{dt^3 u_1 \sin\theta}{6} \\ dt\sin\theta & \frac{dt^3 u_1 \cos\theta}{6} \\ 0 & dt \end{bmatrix},$$

$$C(x,u) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Trust Region Algorithm Implementation

The trust region algorithm used to solve **Problem 1** is described as **Algorithm 2**.

Algorithm 2 Trust Region algorithm

Initialize: Nominal trajectory $(\bar{x}^0_{0:T}, \bar{u}^0_{0:T-1})$, Exploration Parameter:$\gamma$, Trust region : Δ^0_x.

Set $k = 0$, $0 \leq l \leq 1$, $r = 0, \alpha$

while $\|x^{k+1} - x^k\|_2 \geq \varepsilon$ **do** ▷ Termination Condition

  while $\|\bar{x}^0 - x^k\|_2 \leq \gamma$ **do**

    At iteration k Solve : **Problem 2**Eq. (3.43) for $(\delta x^k, \delta u^k)$ for $\Delta_x = \alpha * l^r$

    $x^k_{0:T} \leftarrow x^{k-1}_{0:T} + \delta x^k$

    $k \leftarrow k + 1$

  end while

  if $\|\bar{x}^0 - x^k\|_2 \geq \gamma$ **then** ▷ Exploration violated, update trust region

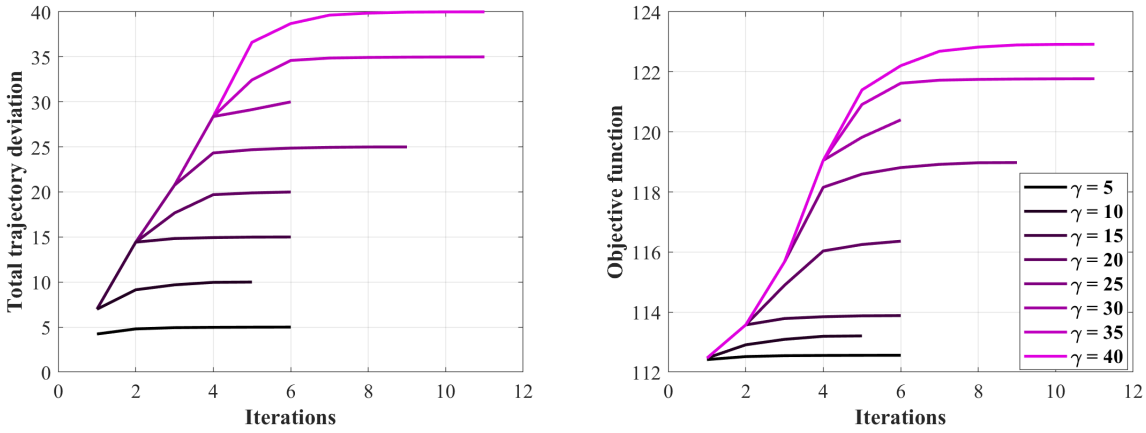
    discard update δx^k

    $k \leftarrow k$

    $r \leftarrow r + 1$

  end if

end while



(a) Total deviation from nominal condition for each exploration conditions.

(b) Objective function growth per iteration for each exploration condition.

Figure 3.4: Trust region algorithm analysis for different observability exploration conditions used as termination conditions.

The algorithm we have implemented in this chapter for the trajectory generation problem is based on a sequential method that involves a trust region, for which the convex subproblem is feasible. The setup is described in **Algorithm** 2. For the dynamics defined in the previous section, we set a trust region by setting $\|\delta x\|_\infty \leq \alpha * l^r$; l denotes the learning rate set to the value of 0.5. The parameter α must be set such that for $r = 0$, the initial trust region $\alpha * l^r \leq \Delta_x^0$ satisfies Eq. (3.25). The trust regions shrink as iterations increase, therefore dynamic constraints are always satisfied. The trajectory at the first iteration is initialized with the nominal trajectory; at each subsequent iteration **Problem 2** Eq. (3.43) is solved in MATLAB using CVX [47]. We present the analysis for the performance of Algorithm 2 in Figure 3.4. The exploration condition $\|\bar{x} - x\|_2 \leq \gamma$ represents the total deviation from the nominal trajectory. We observe that the trust regions updated by the algorithm ensure that the observability maximization does not violate the exploration constraints as seen in Figure 3.4a. For case of estimation aware trajectory shown in Figure 3.3, we present the

corresponding γ evolution and degree of observability progression as a function of algorithm iterations. The trust region algorithm reduces maximum deviation as exploration constraints are reached. The shades of pink s how increasing exploration γ from nominal trajectory on x-axis in Figure 3.4a. The γ values for each case of estimation-aware trajectory are shown on the right in Figure 3.4b. The algorithm is terminated when the maximum exploration condition is reached and the trust region has been reduced below a threshold where no more deviations are feasible. Figure 3.4b shows that the degree of observability is increased by allowing more exploration. As seen in Figure 3.3, a higher exploration limit leads to an improved estimation process. This comes at a cost as the system potentially reaches the boundary of the set for the control and state constraints; depending on the mission, this may not be desirable. Therefore, it is judicious to increase the observability metrics while monitoring their potential undesirable implications on the synthesized trajectory.

3.5.2 Observability based augmented trajectory design for satellite rendezvous

The estimation-aware trajectory planning problem can be solved alongside the original task optimization, by augmenting the observability metric Eq. (3.29), with a penalty term denoted as $\lambda_{\text{obs}} > 0$. This penalty term determines how priority must be given to observability against task completion. In practice, the validation step must first be defined to explore the output space and determine the enveloping function given in Assumption 3.1.1, ensuring that the observability problem is feasible. Given a convex objective function for the task optimization problem, expressed as $J'(x,u) = \sum_{t=0}^{T} c(x_t, u_t)$, consider its estimation-aware version via the augmented objective function as,

$$\begin{aligned} \min_{u_{0:T-1}} \quad & \left(\sum_{t=0}^{T} c(x_t, u_t)\right) - \lambda_{\text{obs}} D_O^{\ell}(Y_{x_{0:T}}) \\ & x_{t+1} = f(x_t, u_t) \quad \forall t = 0, \cdots, T-1, \end{aligned} \tag{3.44}$$

where $Y_{x_{0:T}} = \Gamma_{x_0} u_{0:T-1}$, emphasizing that the observability metric implicitly depends on the input sequence. The objective function used here is defined in Eq. (3.31). The problem

formulation consists of a convex objective function, nonlinear dynamic constraints, and state constraints. Is is assumed that the dynamics and constraint functions are differentiable. In order to iteratively solve this planning problem, the Successive Convexification ((SCvx)) Algorithm [136] can be used. Any other programming approach that handles the nonlinear dynamics and non-convex constraints can be applied here depending on the problem. Specifically, at each iteration k, the following optimal control problem is solved around the trajectory $x^k_{0:T}, u^k_{0:T-1}$,

$$\min_{\delta u_{0:T-1}} \quad L^k(\delta x^k, \delta u^k) \tag{3.45}$$

$$\delta x^k_{t+1} = A_t \delta x^k_t + B_t \delta u^k_t + E v^k_t \quad \text{for all} \quad t = 0, \ldots, T-1,$$

$$\|\delta u^k_t\|_\infty \leq (\Delta_u)^k, \tag{3.46}$$

where $L^k(\delta x^k, \delta u^k) = \left(\sum_{t=0}^{T} c(x^k_t + \delta x^k_t, u^k_t + \delta u^k_t) \right) + \lambda_{\text{obs}} D^\ell_O + \sum_{t=0}^{T} \lambda P(E^k_t v_t)$ and the solution to the current iteration is given by $x = x^k + \delta x^k,\ u = u^k + \delta u^k$. Here, P is the exact penalty function and λ is the penalty weight defined in [136]. The term v^k_t is an optimization variable called the *virtual control buffer*. This term is used to make the problem feasible when the convexification step leads to dynamic infeasibility. Therefore, the matrix E is chosen to have full row rank in order to provide full authority for selection of v^k_t. The penalty imposed on this term ensures that eventually, the virtual buffer converges to zero. The iterative algorithm and the trust region update policy are described in [137].

Now the trajectory generation problem for an Ego-Target rendezvous operation using the above setup can be discussed. The underlying model describes the relative dynamics between an Ego spacecraft and an uncooperative and uncontrolled Target satellite in the Hill's frame. The Target's orbit has been identified, and the Ego spacecraft is initially placed in a nearby parking orbit. The objective is for the Ego spacecraft to rendezvous with the Target while avoiding a predefined keep-out zone (shown as the blue sphere), as illustrated in Figure 3.6.

For this scenario, its has been assumed that a camera on the Ego spacecraft captures images of the Target, and a machine learning (ML) algorithm estimates the Target's pose (position and orientation) relative to the Ego spacecraft. The output of the ML algorithm,

serves as the observed output of the system in the Target-Ego Hill's frame. A tracking controller maintains the attitude of the Ego spacecraft to ensure that the Target remains centered in the camera's field of view. The implementation of such an ML-based attitude tracking setup, along with its integration with ML-based estimation, is discussed in [57]. The ML algorithm's architecture, implementation, and validation are detailed in [20].

The ML based approach has been used here as the demonstration case since it has been observed that the corresponding uncertainties are generally state-dependent. Specifically, the performance of the ML-estimator depends on key-point identification on the Target's image. As such, the accuracy of the estimation process is directly proportional to the number of keypoints detected. The visibility of these keypoints is essential for the estimation accuracy as keypoints not illuminated by the Sun cannot be identified. As a result, the accuracy of the estimation is proportional to the illumination of Target as studied in [20] and shown in Figure 3.2. Moreover, for this rendezvous problem, illumination is dependent on the Ego position, as Sun's relative position is fixed during the operation. Thus, illumination is dependent on the state of the system, and therefore the output uncertainty in this case is state-dependent. This type of sensor model is also applicable to other generic image processing tools that rely on feature extraction from the image. Note that several factors may affect the sensor accuracy, and characterizing uncertainty pertaining to every feature and parameter is non-trivial. Therefore, one needs to analyze the uncertainty as a function of the agent state while exploring the parameter space at each state. For this setup, assume that the maximum uncertainty size can be closely approximated as a convex set with respect to the state, as described in Assumption 3.1.1; for this example, a set defined by a quadratic function will be assumed. Hence, the sensor uncertainty for this case is upper bounded by the quadratic enveloping function given by $\widehat{\Lambda}(\mathcal{Y}_x) = x^T Q x + p^T x + r$, where $Q \in \mathbb{S}_+^{n_x}, p \in \mathbb{R}^{n_x}, r \in \mathbb{R}$. The uncertainty set $\mathcal{Y}_x$, if not predetermined, can be approximated from the data as discussed in Section 3.5.3.

It is important to emphasize that for different scenarios, features that define state-dependency of the measurement uncertainty are distinct. Moreover, the architecture, pa-

rameters and training datasets of ML algorithms will have a significant impact on the state dependency of the output uncertainty. The implementation details for this scenario are discussed in Section 3.5.3

The estimation-aware trajectory design problem just described satisfies conditions of Theorem 3.3.1; as such, it can be formalized as Eq. (3.44). This problem involves the synthesis of a trajectory such that the Ego spacecraft approaches the Target with some initial separation and velocity relative to the Target. The Ego spacecraft is required to approach the Target while keeping out of a zone of five meters from the Target. The design objective is formulated such that the quadratic terms add penalty to the relative Ego approach error and fuel consumption. The observability metric augments this cost with a regularization term used as a tuning parameter. The overall optimization problem is thereby defined as,

$$\min_{u_{0:N-1}} \quad \left(\sum_{t=0}^{T} c(x_t, u_t)\right) - \lambda_{\text{obs}} D_O^{\ell}(Y_{x_{0:T}}) \tag{3.47}$$

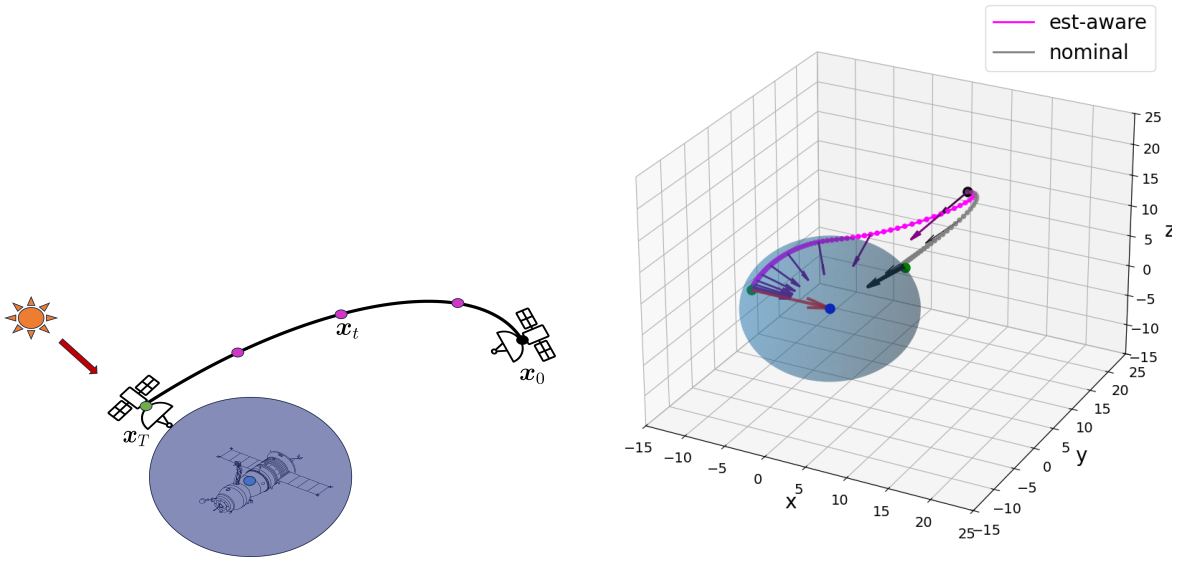
$$x_{t+1} = f(x_t, u_t) \quad \forall t = 0, \ldots, N, \qquad \text{Discretized relative dynamics} \tag{3.48}$$

$$||Cx_t|| \geq d \quad \forall t = 0, \ldots, N, \qquad \text{Keep-out zone} \tag{3.49}$$

$$u_t \sim \mathcal{U}_{\mathcal{O}}, \quad Cx_t \sim \mathcal{X} \quad \forall t = 0, \ldots, N. \qquad \text{State and action constraints} \tag{3.50}$$

Here, the regularization term λ_{obs} parameterizes the permissible deviation from the nominal trajectory. The objective function J is designed to be convex as $l(x_k, u_k) = x_k^\top Q x_k + u_k^\top R u_k$ and $-D_{\mathcal{O}}^{\ell}$ are both convex. As the enveloping function is defined with a quadratic form, the cost function defined in Eq. (3.31) is used here. The discretized dynamics in this setting is defined in Eq. (3.51)-(3.53). The keep-out zone constraints shown here are non-convex, and as such, are linearized at each iteration of the SCvx approach. Using the iterative SCVx approach, the convex subproblems as described as Eq. (3.45). The implementation details are described in Section 3.5.3.

Representative trajectories for this planning problem formulated as Eq. (3.47)-(3.50), are shown in Figure 3.5b. The proposed solution strategy thereby obtain an estimation-aware



(a) Schematic representation of the Ego-Target Rendezvous problem.

(b) Trajectory comparisons for Ego spacecraft.

Figure 3.5: Ego Agent trajectory for Target rendezvous with a keep-out zone.

trajectory such that the Ego camera points to the Target with the optimum illumination angle. Figure 3.5a depicts the schematic for the problem where the Target is shown in the center within the keep-out zone in blue and the Ego agent following the proposed trajectory. The Sun direction is shown by the red arrow. Note that aligning with the Sun-angle reduces the instantaneous output uncertainty for the keypoint-based ML estimator. Figure 3.5b depicts the derived solutions for the nominal trajectory which is designed without the augmented observability cost compared with the estimation-aware trajectory that is the solution to Eq. (3.47)-(3.50). The Ego trajectories for the nominal case is shown in gray, where the Target relative position is the origin and its keep-out zone is shown by the blue sphere. This is the best trajectory for a rendezvous scenario. The Ego trajectory is initialized at the location $[7, 20, 10]^\top$m shown by the black dot in Fig 3.5b, and an initial velocity of $[10, -5, -5]^\top$m/s relative to the Target shown in green at the origin. The Ego reaches

the goal location shown in green with relative position vectors shown by the arrows on the trajectory. The estimation-aware trajectory is shown in pink, where the final Ego position is such that the relative Ego position achieves the best viewing angle during the course of its trajectory.

The choice of the regularization term λ_{box} signifies the weighting of the planned trajectory between estimation and rendezvous tasks. The estimation performance over the nominal and estimation-aware trajectories and the progression of uncertainty for different cases of λ_{box} are shown in Figure 3.6. For these simulations, a simple low-level controller that tracks the waypoints on the designed trajectories have been chosen. The trajectories are followed by tracking the waypoints $x_{0:T}$ generated for each λ_{obs}. As also mentioned previously, a Lurenberger state observer has been designed for the discrete time system and tracking error norm is computed at each waypoint. Similar to the previous scenario, the online system is $\hat{x}_{t+1} = (A - LC)\hat{x}_{t+1} + Bu_t + Ly_t$, where the output is drawn uniformly from the uncertainty set given by $y_t \sim U(\mathcal{Y}_{x_t})$. The tracking error norm can be computed by $\|e_t\| = \|\hat{x}_t - x_t\|$. Now, for analyzing the tracking performance, a Monte Carlo simulation with $N = 1000$ roll-outs for the system for trajectory designed by each case $\lambda_{\text{obs}} = [0, 1, 2, 3]$ has been developed. Figure 3.6b depicts the error norm over time $\|e_t\|$ for each trajectory. The dashed lines show the mean error for all roll-outs for each trajectory and the solid lines show the maximum error norm over each roll-out. Here we can observe that the full state error is significantly reduced for the estimation-aware trajectories and it improves as more importance is given to the estimation performance. Note that this is not just the measurement error but the state estimation error, which highlights that tracking for the velocity state improves as well (not simply the static position outputs). Next, the trace of the state covariance Trace $\big(E[(x_t - E[x_t])(x_t - E[x_t])^\top]\big)$, at each time for $N = 1000$ roll-outs, has been computed for each λ_{obs}. In Figure 3.6c, the progression of the trace of the state covariance over time has been shown. Here, one observes that the total state variance is reduced for the estimation-aware trajectories as more importance is given to the measure of observability. Moreover, since the Rendezvous problem is a free-final state problem, we

observe that the Ego spacecraft reaches different final states based on λ_{obs}. The following section talks about implementation of trajectory design scenario and experimental validation.

System Dynamics for Satellite Rendezvous

The relative (translational) dynamics for the Ego spacecraft are specified in the Hill's frame of the Target. Specifically, the Clohessy–Wiltshire equations [192] are used to define the dynamics for the position (x, y, z) (coordinates in the Hill's frame) and velocity $(\dot{x}, \dot{y}, \dot{z})$ as,

$$\ddot{x} = 3n^2 x + 2n\dot{y} + u_x, \tag{3.51}$$

$$\ddot{y} = -2n\dot{x} + u_y, \tag{3.52}$$

$$\ddot{z} = -n^2 z + u_z, \tag{3.53}$$

where $n = 0.00113\text{s}^{-1}$ denotes the mean motion of Target spacecraft circular orbit and u's are the control inputs in each corresponding axis. Consider now the position and velocity vectors $p = [x, y, z]^\top \in \mathbb{R}^3$ and $v = [\dot{x}, \dot{y}, \dot{z}]^\top \in \mathbb{R}^3$. First, redefine the state vector as $x = [p^\top, v^\top]^\top \in \mathbb{R}^6$ and inputs as $u = [u_x, u_y, u_z]^\top \in \mathbb{R}^3$. The output of the system measures the relative state of the Ego with respect to the Target; hence, the nominal output is defined as $y = p \in \mathbb{R}^3$. The problem defined in Eq. (3.47) can now be solved using the discretized linear dynamics of the form Eq. (3.23)-(3.24) for the system defined in Eq. (3.51)-(3.53). The system matrices for the model $x_{t+1} = Ax + Bu;\ y_t = Cx_t$ are defined as,

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

3.5.3 SCvx implementation details

The SCvx algorithm converts the problem Eq. (3.47) into a sequence of convex subproblems of the form,

$$\min_{\delta u_{0:T-1}} \quad L^k(\delta x^k, \delta u^k) \tag{3.54}$$

$$\delta x^k_{t+1} = A_t \delta x^k_t + B_t \delta u^k_t + E v^k_t \quad \forall t = 0, \cdots, T-1,$$

$$s(x^k_t) + S^k \delta x^k_t - s^k_t \leq 0, \tag{3.55}$$

$$\|\delta u^k_t\|_\infty \leq (\Delta_u)^k. \tag{3.56}$$

Here $L^k(\delta x^k, \delta u^k) = \lambda_1 \left(\sum_{t=0}^{T} c(x^k_t + \delta x^k_t, u^k_t + \delta u^k_t) \right) + \lambda_2 \lambda_{\text{obs}} D^\ell_O + \lambda_3 \sum_{t=0}^{T} P(E^k_t v^k_t, s^k_t)$, where λ's are the tuning penalty weights. The keep-out zone constraint in Eq. (3.49) for distance d is written as $s(x) = d^2 - \|Cx\|^2$ with the corresponding Jacobian is $S = \partial s / \partial x = -2C^\top Cx$. The penalty functions on the virtual buffer v^k_t and the constraint violation s^k_t are in terms of the 1-norm. Here, the SCvx algorithm is applied as described in [137]. The penalty terms are chosen as $\lambda_1 = 10^{-1}$, $\lambda_2 = 10^{-1}$ and $\lambda_3 = 10^3$. The implementation results for the scenario shown in Figure 3.5a and the progression of trust regions and the objective function are depicted in Figure 3.7. The trust region is updated to accommodate for a higher trajectory deviation and then converges as shown in Figure 3.7a. The evolution of the objective function can be seen in Figure 3.7b. The objective function grows initially as the penalty terms for the state constraints are high due to the initialized trajectory. Over subsequent iterations, the objective function converges. The heavy penalization of constraints violation terms ensures fast convergence of buffer terms v and s, as seen in Figs. 3.7c and 3.7d.

Simulation setup

The validation experiments and online analysis are performed in a visual simulator. A closed-loop end-to-end pipeline has been developed to simulate high-fidelity spacecraft dynamics and produces photorealistic high-resolution images for the ML-based state estimator. The

rendering engine used here is built as a "level" in Unreal Engine software, where a Low-Earth orbital environment with realistic lighting and textures are simulated. The detailed architecture of the CNN and integrated simulation pipeline has been described in the previous work [20]. The filtered state estimates generated by applying the perception map to the images are then used as feedback for the tracking algorithms. The simulator is a modular setup where we can select appropriate astrodynamics simulators or generate the dynamics in Python. The offline trajectory planning, which is the main optimization problem examined in this chapter, has been developed using CVXPY in Python [59].

3.5.4 Validation experiment Design

In order to provide robustness guarantees, uniform error bounds must be provided for the perception map. A method similar to the procedure suggested by Dean *et al.* [50] can be adapted, where they generate a slope-dependent model of the estimate error near any training point. However, in this work, the uncertainty is modeled as a state-dependent ellipsoidal set. As such, with the assumption that the output uncertainty with respect to each state is bounded, bounds are estimated from the data.

The experiment design requires characterizing the uncertainty as a function of state-dependent parameters. We now need to create the state-to-output uncertainty map as discussed in Eq. 3.5, and approximate its upper bounding convex envelope. In this setting, the perception map is denoted as $y = p(x)$, where p denotes the process of generating the image corresponding to state x and performing inference on the image to generate an uncertain pose output y. Figure 3.8 depicts this perception map, that operates on the agent's state x. The Ego-Target pair is simulated and their relative state is defined by x. An image of the Target is generated by process described in Figure 3.8. This represents the sensors in any other application user wants to define. The ML estimator performs inference on this image to generate the output $y = p(x) \sim U(\mathcal{Y}_x)$, with uniformly distributed uncertainty around $h(x)$ as the nominal output. The following assumption is now invoked:

Assumption 3.5.1. *The perception map $y = p(x)$, samples the measurements with a uniform distribution from a set $\mathcal{Y}_{\S}$. The size of the uncertainty set is given by a locally and globally bounded function $\Lambda(\mathcal{Y}_x)$.*

This is assumption implies that an outlier-rejection algorithm has been applied to the perception map output, keeping the output uncertainty bounded. Moreover, the map should also be immune to adversarial noise, which can affect the output when the image input is not generated strictly from the same distribution as the training dataset [217]. For the case of satellite pose estimation scenario, the adversarial noise is non-existent as the simulator platform is used for testing and training but can occur in practical setting.

For a state x, the uncertainty is defined as $z_x = p(x) - h(x)$. Here, h is the nominal output map for the system process; for the rendezvous for example, it is the relative position measurement for the state x without any uncertainty. With the bounded assumption, z_x is said to be a uniformly distributed random variable. The 2-norm bound on z_x is the size of the uncertainty represented by the function $\Lambda(\mathcal{Y}_x)$, mentioned in Assumption 3.1.1. The bounded random variable z_x can be interpreted as being sub-Gaussian with a variance $\sigma^2 = \Lambda(\mathcal{Y}_x)^2/4$ [218]. Concentration inequalities have been well established for sub-Gaussian random variables, with the consequence that their respective norms can be estimated for N randomly drawn samples, as,

$$d_x = \frac{1}{N}\sum_{i=1}^{N}\|z_x^i\|, \tag{3.57}$$

with probability $1-\delta$, where $\delta \propto 1/N$ [218]. The approximation concentrates at $\sigma^2\sqrt{n}$, where n is the dimension of z. Therefore, the max bound can be computed as $\widehat{\Lambda}(\mathcal{Y}_x) = \sqrt{d_x/\sqrt{n}}$. The datapoints $z_x^i = p(x) - h(x)$ are sampled by randomizing parameters of perception map p that introduce the uncertainty.

For the validation dataset, the enveloping function is generated by computing $\widehat{\Lambda}(\mathcal{Y}_x)$ for M samples of $x \in \mathcal{X}$. Here, x is sampled from $\mathcal{X}$ by generating a dense discretization on $\mathcal{X}$. For the dataset $(x_i, \widehat{\Lambda}(\mathcal{Y}_{x_i}))$, $i = 1, \ldots, M$, a tight quadratic envelope function of the

form $x^\top Qx + b^\top x + c$, where $Q \in \mathbb{S}_+^{n_x}, b \in \mathbb{R}^{n_x}, c \in \mathbb{R}$ is identified by solving the regression problem:

$$\min_{Q \succeq 0} \quad \sum_{i=1}^{M} \|(x_i^\top Q x_i + b^\top x_i + c) - \widehat{\Lambda}(\mathcal{Y}_{x_i})\|_2^2 \tag{3.58}$$

$$\text{s.t} \quad x_i^\top Q x_i + b^\top x_i + c - \widehat{\Lambda}(\mathcal{Y}_{x_i}) \geq 0, \quad i = 1, \ldots, M. \tag{3.59}$$

This problem can be compactly written as,

$$\min_{q} \quad \|L - Xq\|_2^2 \tag{3.60}$$

$$\text{s.t} \quad Xq - L \geq 0, \tag{3.61}$$

$$q > 0. \tag{3.62}$$

Note that the quadratic term has been expanded as $x_i^\top Q x_i = q_{11}x_1^2 + q_{12}x_1x_2 + \ldots + q_{nn}x_n^2 + b_1x_1 + \ldots b_nx_n + c := q^\top x_i'$, for $q = [q_{11}, q_{12}, \ldots, q_{nn}, b_1, \ldots, b_n, c]$, $x_i' = [x_1^2, x_1x_2, \ldots, x_n^2, x_1, \ldots, x_n, 1]$, converting the quadratic term into its linear form for an n dimensional x_i. The data matrices are defined as $L = [\widehat{\Lambda}(\mathcal{Y}_{x_1}), \ldots, \widehat{\Lambda}(\mathcal{Y}_{x_M})]^\top$ and $X = [x_1'^\top, \ldots, x_M'^\top]^\top$. The constrained least-square problem can now be solved as an linear program, in order to generate the enveloping function, parametrized by q.

In this context, the system is given by Eq. (3.51)-(3.53), for the state x representing the relative position and velocity of the Ego and the output y representing the relative position, as captured by the ML estimator. In order to generate a data-point of the dataset $(x_i, \widehat{\Lambda}(\mathcal{Y}_{x_i}))$, a primary dataset is of N points is collected, to solves for $\widehat{\Lambda}(\mathcal{Y}_{x_i})$ using Eq. (3.57) at each x_i. The state x_i is sampled by densely discretizing the space $\mathcal{X}$ with a $n_x = 6$ dimensional grid. For computing Eq. (3.57) at each x_i, we obtain a point $z_{x_i}^j$ by running inference using the process shown in Figure 3.8. The N samples $z_{x_i}^j$ are generated by varying the parameters for the simulation environment, that add randomness to image generated at x_i. For each x_i, the parameters that add uncertainty to the measurements have been classified as Target rotations, camera noise, and the background (Earth placement in the image), as detailed in [20]. A random Target rotation, Earth position and camera noise is sampled

under feasible conditions to compute $z^j_{x_i}$. The process is repeated for M points to generate the secondary dataset $(\widehat{\Lambda}(\mathcal{Y}_{x_i}), x_i)$. Figure 3.9a represents the points x_i sampled from $\mathcal{X}$ within the same distance from the Target (shown in green); this is shown as points (shown in blue) on a sphere. The Ego positions are shown in blue dots show the point x_i while the camera points towards the Target as shown by the red arrows. The Sun direction is shown by the black arrow. The secondary dataset $(x_i, \widehat{\Lambda}(\mathcal{Y}_{x_i}))$, $i = 1, \ldots, M$, is then populated with x_i, sampled from $\mathcal{X}$ for feasible distances from the Target. The enveloping function of the quadratic form is now identified by solving Eq. (3.60). The solution to Eq. (3.60) gives the $n_x = 6$ dimensional, quadratic enveloping function. Figure 3.9b represents a dimension of $\Lambda(\mathcal{Y}_{x_i}) = x^\top Q x$ perpendicular to the Sun vector. The Sun angle is defined as the angle between the camera vector and the Sun vector. Figure 3.9b shows the representation of the quadratic envelope projected onto the 2D plane in-line with the Sun-angle. The dashed line covering the max error bounds represents the envelope. The solid lines represent variation of the max error at Ego positions with the same distance from the Target, with respect to their Sun-angles. Figure 3.9b also illustrates that uncertainty scales sharply as the distance to the Ego increases, as keypoint visibility reduces with distance. This is the main "feature" of this key-point based perception map that is exploited in our estimation aware planning.

Note that the data gathering step of the above process and the corresponding uncertainty parametrization are influenced by the simulation platform and the problem scenario. For other applications where an estimation-aware trajectories need to be generate, the data can be collected in a similar manner by sampling the state space and the uncertainty domain. When the uncertainty set follows Assumption 3.5.3, the validation experiments can be used to define an enveloping function that upper bound the state-dependent uncertainty set.

To summarize, the enveloping function is generated by collecting a primary dataset that solves Eq. (3.60) to give $\widehat{\Lambda}(\mathcal{Y}_x)$ at each x. Then a secondary dataset is collected to compute $\widehat{\Lambda}(\mathcal{Y}_{x_i})$ at each x_i. Here M such samples of x are collected from dense discretization of $\mathcal{X}$. This secondary dataset is used to solve Eq. (3.58) that defines the envelope. In principle any alternate method that computes a tight approximation described in Eq. (3.10) can be

employed for the estimation-aware approach in this chapter to be valid.

For the two scenarios presented in this section, estimation-aware trajectories have been designed using the proposed degree of observability metric. As seen in Figs. 3.3 and 3.6, the state estimation process significantly improve using the prescribed methodology. This proposed approach can be interpreted as improving the sensor conditioning, where the uncertainty in the measurement is mapped to a smaller uncertainty in the state estimation process. Moreover, along the estimation-aware trajectory, the lower-level state feedback controller would generally require less energy due to smaller feedback errors, thus improving the tracking performance. The proposed approach also has application in multi-agent settings, where observations of other agents' behavior for the purpose of coordination has a set-valued uncertainty.

3.6 Discussion

In this chapter, an observability-based metric for set-valued measurements induced by state-dependent errors was presented. This metric can be utilized to design estimation-aware trajectories for set-valued uncertainty measurements when a certain convexity assumption holds. Specifically, the output uncertainty map must be convex in its size with respect to the state Eq. (3.6). In practice, when this map is not smooth, an enveloping convex function must be approximated through a validation step to establish a tight upper bound on the uncertainty. Within this setup, the manuscript demonstrates that when the output maps are *observability-regular* (Definition 3.3.1), a lower bound on the *degree of observability* Eq. (3.22), can be defined. Under reasonable assumptions on the state dependency of the uncertainty set, this lower bound is a concave function. The chapter then proceeded to show that an optimization problem can be formulated to maximize the degree of observability, thereby generating estimation-aware trajectories.

This work has also illustrated the utility of the proposed observability metric in formulating an augmented cost function for trajectory design problems. A case study was considered, in which an ML-based state estimator, implemented by an Ego spacecraft, tracks

an uncooperative Target satellite. The proposed procedure constrains the estimator as a state-dependent set-valued map. It has been shown that the proposed metric can generate a tracking trajectory for the Ego spacecraft, resulting in improved estimation accuracy. Furthermore, this work has demonstrated that, given a nominal trajectory, a neighboring estimation-aware trajectory can be designed by maximizing the observability metric lower bound while imposing exploration constraints in the neighborhood of the nominal trajectory. This case was presented with an example using Dubins car model.

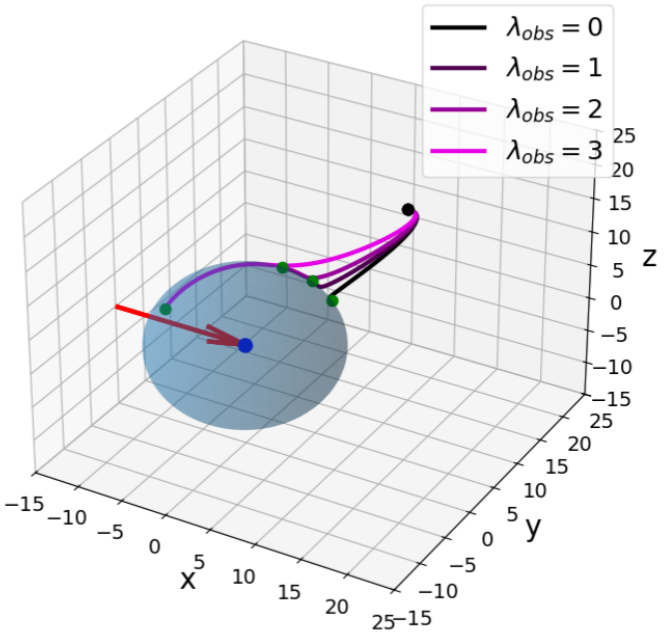
This proposed framework has been formulated for a general class of differentiable nonlinear systems with a set-valued output uncertainty. The optimization problems that use the observability metrics are defined for locally linearizable systems where the size of the output uncertainty is globally convex. The proposed method is effective for designing offline estimation-aware trajectories for observability-regular maps defined in the neighborhood of a nominal trajectory.

Further investigations into output uncertainties with non-ellipsoidal bounds would be a natural extension of this work. The framework can also be extended to cases where the uncertainty size can only be locally approximated as a convex function with respect to the trajectory state, while the optimization problem is solved sequentially. Additionally, further exploration of connections to other uncertainty classes in state-dependent settings and the use of a metric that quantifies output perturbation could follow as a comparative study.

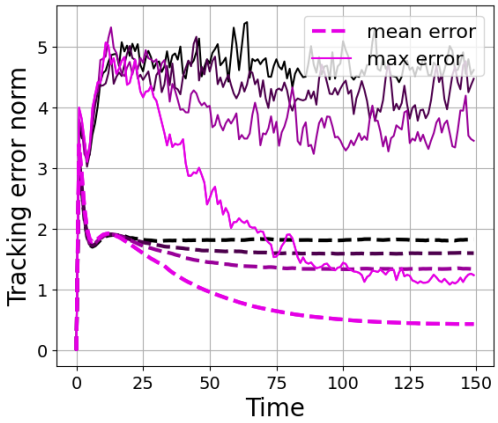
Another facet of this problem is its interpretation as a dynamic sensor placement problem, where sensor trajectories can be designed to enhance an agent's state estimation. If a sensor map is provided and the sensor can be dynamically positioned around the agent it is tracking, the observability metrics can be employed to design an estimation-aware trajectory for the sensor-agent pair. Moreover, a multi-agent sensor placement scenario can be constructed, where sensors interact to maximize combined agent information as they are dynamically positioned to optimize the agent's observability.

Finally, for real-time implementation, a model predictive approach can be adopted, utilizing an updated parametrization of the output uncertainty to optimize the trajectory for

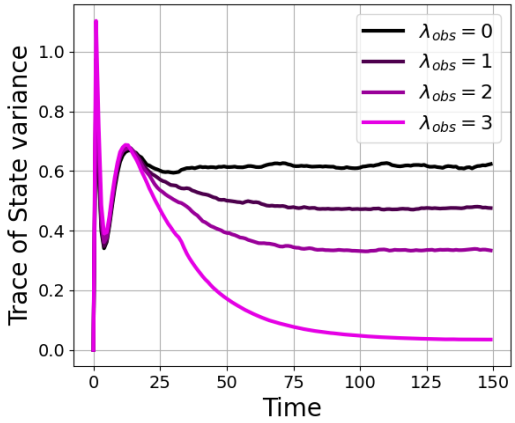
estimation. This approach is applicable when online estimation begins to degrade, and the uncertainty map and model information are known. We demonstrate the use of observability-based metrics using MPC for the Target tracking problem in our work [56].



(a) Estimation-aware trajectories for increasing observability conditions.

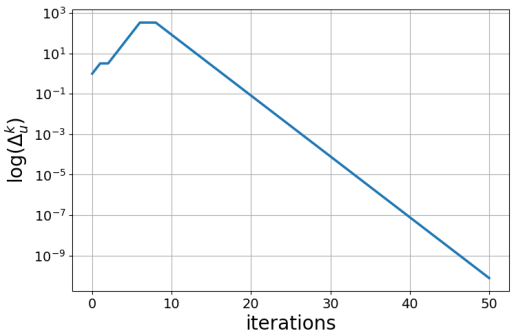


(b) Comparing tracking-error over time for for trajectories show in Figure 3.6a.

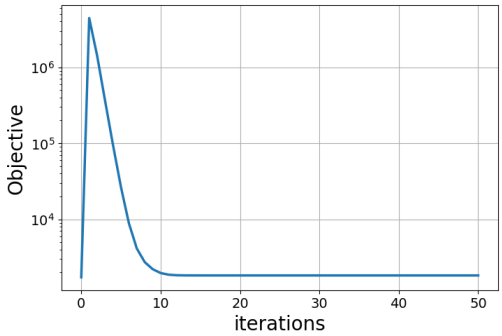


(c) Comparing the state-estimate variance for trajectories over time.

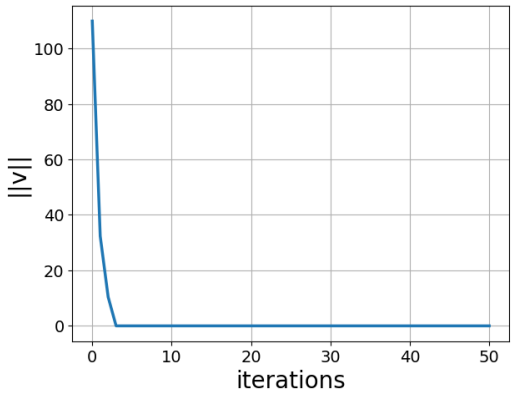
Figure 3.6: Comparative analysis for satellite rendezvous example. Here estimation-aware trajectories and their error analysis for increasing observability condition are presented.



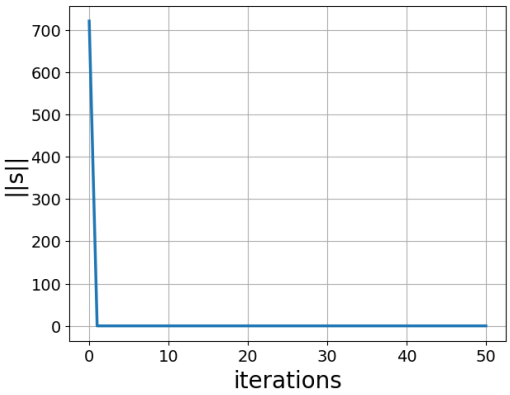
(a) Trust Region size updates per iteration.



(b) Objective function update per iteration.



(c) Virtual control buffer evolution.



(d) Constraint violation term evolution.

Figure 3.7: SCVx implementation statistics for estimation-aware Ego-Target Rendezvous problem.

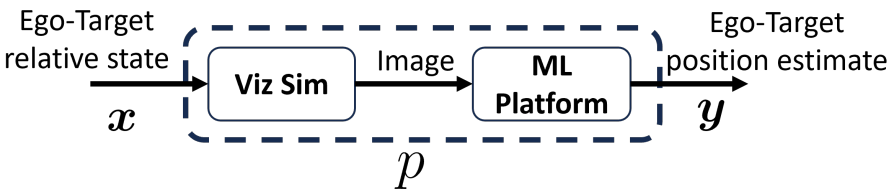
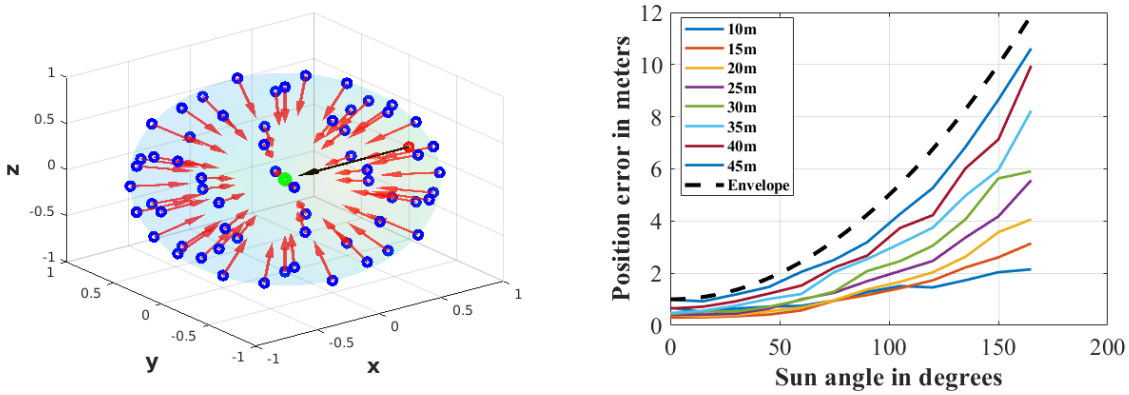


Figure 3.8: Schematic showing relative state to position estimate map



(a) Representation of sampling strategy for validation step showing the Ego positions(blue) relative to Target(green).

(b) 2D representation of envelope(black) compared to maximum output uncertainty size vs Sun angle.

Figure 3.9: Validation experiment for fitting envelope function to output uncertainty.

Part II

NEUROMODULATION

Chapter 4

ANALYSIS OF NEURONAL DYNAMICS AND CONTROL

This chapter discuss the design and analysis of a feedback controller for regulating cortical activity in Mice. Conventionally feedback or closed loop control is not studied as extensively as inference based modeling. Often the primary research focus in Neuroscience is based on understanding models governing the neuronal subsystems. Specifically mechanisms behind systems driving behavior are of interest as modeling these systems and stimulating them could have a quantifiable effect. We start with looking at conventional approaches to closed loop control in neuroscience. Closed-loop methods in neuroscience integrate real-time feedback from neural activity to dynamically adjust inputs. Unlike traditional open-loop experiments, which stimulate and then observe without adaptation, closed-loop approaches respect the inherent feedback mechanisms of the nervous system. In such systems, neural output signals—such as spikes, local field potentials, or behavior—are continually monitored by sensors, processed online to detect events or state changes, and subsequently used to drive adaptive stimulation via electrical, optical, or mechanical actuators.

Closed-loop control in neuroscience enables real-time measurement and manipulation of brain activity, with recent advances allowing unprecedented spatiotemporal precision in the mouse cortex. Optogenetics provides millisecond-precision stimulation without electrical artifacts, making it ideal for such applications [89]. All-optical systems combining two-photon calcium imaging and photostimulation allow on-the-fly detection and modulation of individual neurons in awake mice [153]. These approaches have been used to target hippocampal sharp-wave ripples [73] and to abort seizures by light-based silencing of excitatory neurons [114]. Hardware–software platforms such as CANDO [74] and ORCA [90] integrate high-speed imaging, online analysis, and immediate optogenetic feedback. Wide-field cal-

cium imaging has also been paired with real-time feedback to link cortical states to forelimb movement control [91]. Such closed-loop frameworks enable causal testing of neural dynamics and provide a foundation for adaptive neuromodulation strategies in behaving mice. These closed-loop paradigms establish a foundation for translating neuronal population dynamics into formal control problems, enabling the systematic design of feedback controllers for optogenetic modulation of cortical circuits in behaving mice. Our experimental setup is based on experimental setup at Steinmetz lab. The details of the physiology and methods for data-capture and actuation have been described by Matveev et al [140].

Our investigation specifically focuses on the design of closed-loop controllers targeting spontaneous cortical activity, aiming to leverage these experimental and computational advances to develop robust, real-time control strategies that interact with the intrinsic dynamics of cortical networks. Neuronal dynamics of mice, measured through cortical imaging, can be manipulated by using an optogenetic stimulus. This is an activation that can only inhibit the cortical activity; therefore, it is unidirectional and has an upper bound for safe operations. The observation is a high-dimensional image of the cortex. We specifically want to test the ability of the controller to regulate activity locally in a region of interest using activation at a neighboring region. We design a feedback controller to track the activity of a region near the stimulus to a reference state for a finite horizon. This controller design defines the internal model of the system as it is shown to be a good regulator of the system and helps infer properties of this system.

4.1 System Characterization

The goal of this work is to demonstrate that a feedback controller of a "cortical state" can achieve a good regulation of an output signal for "average" dynamics. The cortical state here is an interpretation of the underlying dynamic neuronal state, which defines the processes in the system.

The output state is measured by partially observing this cortical state. To control or stimulate this state, an input signal is generated using feedback. The output state is the local

readout of a pixel and the input is a laser signal applied to a nearby location. Specifically, control in this context is termed as output regulation, which is the ability of the input sequence to drive the output to some reference signal. The task of tracking the reference is subject to a short horizon due to exposure(input) constraints of the system. To achieve this type of tracking, we want to define a output feedback controller. The system parameters remain unknown and variable since we treat the system as a black box while defining the controller. For the controller to be able to regulate the output under all cortical state conditions, the model must have monotonic input-output characteristics. Moreover, non-stationarity and time-variance can make the linear controller infeasible for output regulation. Given a model, guarantees about dynamic stability can be stated.

4.2 Challenges in Modeling

In general, neuronal dynamics exhibit complex behaviors at the cortical level. In neuroscience, models for cortical activity have been proposed based on experimental analysis and biological insight. The representative dynamics of the cortical state in its general form can be described as a nonlinear system given by:

$$x_{k+1} = f_k(x_k, \rho_k) + g_k(x_k, \rho_k)u_k + d_k \tag{4.1}$$

$$y_k = h_k(x_k, \rho_k) + w_k. \tag{4.2}$$

where f_k, g_k, h_k are time varying continuous functions representing the dynamics and the output maps and x, y, u are the state, input and output vectors with well defined dimensions. The functions f, g, h all depend on the subject parameter ρ that can be partially approximated(such as the synchronized and desynchronized state). The parameter ρ can be perceived as an arbitrarily evolving state with a non-trivial dynamical model. The system is also perturbed by unpredictable exogenous signals d which can be characterized as neuronal activities that affect the subsystem Eq. (4.1-4.2).

These signals are external to the subsystem, but may be generated by some internal uncharacterized neuronal process. The signal v represents the sensor noise in the experiment,

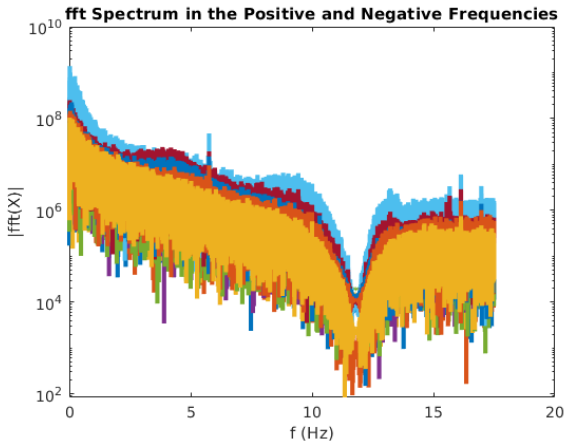


Figure 4.1: Frequency spectrum of first 10 principle moments

and its distribution, which depends on the equipment, can be approximated empirically.

We deal with a subsystem of this large scale cortical system. Here, an optogenetic input is applied to a fixed cortical region, and activity from another neighboring cortical region is measured and considered as an area of interest. A subsystem of the cortical system can be assumed to exist with input/output defined by the activation and measurement locations. We redefine the x, u and y as the state, input and output for the subsystem instead.

The justification for this generalized model can be seen when analyzing the frequency spectrum of the signals by contextualizing available modes of input and output. The evidence of this type of activity can be empirically verified from experiments conducted at Steinmetz Lab and in literature. We start by analyzing frequency spectrum of the output space. The output in this setup is a high dimensional image defined by $z_k = g(y_k)$ we use Principal Component Analysis(PCA) to reduce the dimensions of this high dimensional signal [171]. The spatial components(left) show features of interest in the image and the temporal(right) components represent the time series evolution of this image. We observe that most of the neuronal activity is explained by a small number of temporal principal components, which is a common theme in findings across neurodynamics research [38, 195, 45]. We observe the frequency distribution for data gathered from an experiment session with spontaneous

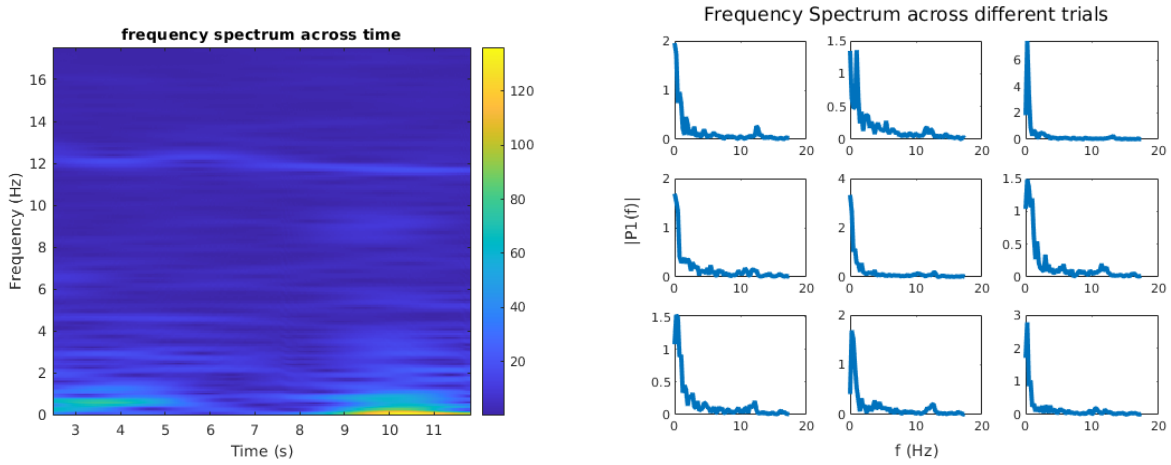


Figure 4.2: Left: Frequency spectrum across time for a session, Right: Freq spectrum over a short trial window across different times

activity where the mouse is left unaltered. We see that there exists a smooth distribution of frequencies as seen in Figure 4.1. For a short horizon we observe variable frequency distribution on non-stimulated spontaneous activity as shown in Figure 4.2. We observe that over short windows features emerge where some frequency modes are dominant but transient. This could be explained as an input with variable frequencies driving an approximately linear system.

The system is analyzed for a finite horizon trajectory called a trial. A trial is an activity of a region of interest on the Brain. The activity which is the output y is computed as the percentage difference from the mean activity denoted as $\%dF/F$ in this context. We observe this output and assume that the latent state x driving the output at any trial is initialized at the start of a trial as $x_0 \in \mathcal{D}$ where $\mathcal{D}$ is the feasible set of the system states. The set $\mathcal{D}$ is bounded as we have a biological system. The distribution of states within this set at initial condition is parameter dependent. The mean of the regional activity may vary over time due to both changes in physiology(parameter change in the system model) or experimental factors(external disturbance). For any given initial condition, an input sequence $u_{[0,T-1]}$ is

applied while disturbance and parameter sequence $d_{[0,T-1]}$ and $\rho_{[0,T-1]}$ also affect the system. The output sequence can be measured for any trial as $\Sigma_{x_0}(u, d, \rho)_{[0,T-1]}$. Note that we do not estimate ρ and d, and we perceive the effect of these variables as variability throughout the expected output trajectory.

For this system, we want to design a controller such that the dynamics can be stabilized and a reference output trajectory can be tracked under the given input constraints. The controller design will depend on the model of this subsystem and the design parameters for tracking. For reliable tracking, the model must be well defined.

In this setting, the system identification for guaranteed predictability is a non-trivial task. This is mostly the case due to non-stationary of the system across a trial. The spontaneous change in parameters and state perturbation through exogenous inputs create the observed non-stationarity.

Commonly used approach of linear subspace identification does not apply to this setup as it non stationarity and as a result stability of the dynamical system for fitting a candidate linear state space model that can be used to forecast noise free dynamics [211, 3, 206]. Moreover the single trial activity in the brain is subject to non-periodic, spontaneous activity which acts as a driving system for the output state that is being tracked. For these inputs to be identifiable, they must be generated through a known model instead of random activation [87]. Conventional system identification remains a challenge due to the variability of output across the same inputs and initial conditions due to the unobserved disturbances and parameter dependence of latent dynamics. But a trend in trial-averaged performance can be observed in the experiments. We observe that by taking an average of trajectories across trials the system behaves like an unbiased, undisturbed system. Using the law of large numbers we observe that for large number of trials we can expect the system to be stationary for finite horizon trajectories. The Figure 4.3 shows reduced trial-by-trial variability(temporal average) as more trials are incorporated. We now look at design approach for controller synthesis and system analysis.

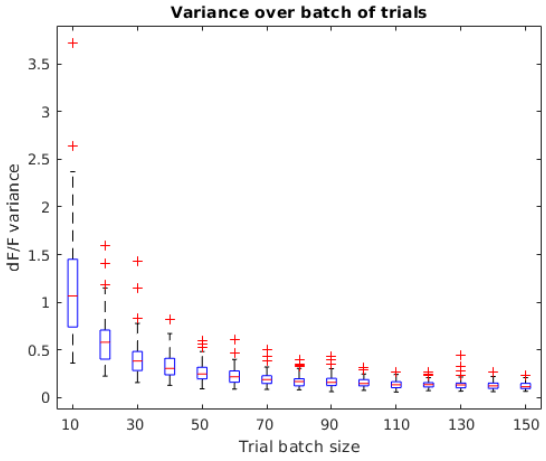


Figure 4.3: The trial-by-trial variability as a function of sample numbers

4.3 System Characterization

The system response to the control stimulus can be defined if the dynamics of the system along with uncertainties are known. For cortical activity, the formulation of such a model under spontaneous activity is non-trivial. Moreover, system identification for prediction metrics is often infeasible when spontaneous exogenous inputs are prevalent. Some works have shown progress in reconstruction and prediction of low-dimensional data with behavioral tasks where spontaneity is less prevalent[1]. For a general investigation, where we want to model dynamic relationship between brain region for facilitating feedback, such models have not yet been shown to be effective.

Now, to manipulating the state and designing a linear state feedback controller with stability guarantees, we need monotonic input-output relationship. With a monotonic input-output system a linear proportional feedback controller will achieve equilibrium [15]. A nonlinear controller could be designed to regulate the desired signal but in that case, controller design needs the model information, specifically the non-linear effects of the system. In optogenetic stimulation, we observe what behavior is exhibited when a given control sig-

[1]Linderman et al, have show switched linear systems can predict activity in C. Elegens.

nal is applied. An experiment is designed to observe the output of the system for varying levels of impulse inputs. In Figure 4.4 we show the trial-averaged output response of the dF/F value of pixel trace for each impulse value to state at which impulse is applied. We observe that the input activation level to output inhibition level ratio is not strictly monotonic but behaves as an approximately monotonic system. This feature is attributed to the variability of the mouse dynamics and the fact that limited number of trials are available due to physiological constraints. We can demonstrate that monotonicity of input-output

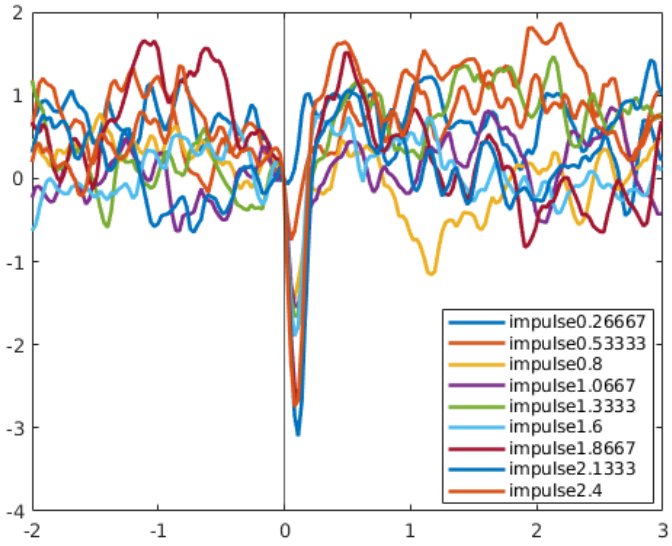


Figure 4.4: Impulse response for optogenetic stimulation

behavior is parameter dependent. We use frequency spectrum as an indicator for parameter where the power spectrum is used as a marker to segregate brain state into phases where response to an input is predictable or unpredictable. It is observed that for trials with lower concentration of low frequency power, responses are more predictable as compared to higher low frequency power. This type of classification used to distinguish known mouse states associated with arousal. The state of arousal or desynchronicity has been observed to have high correlation to output state having low power of low frequency activity whereas higher

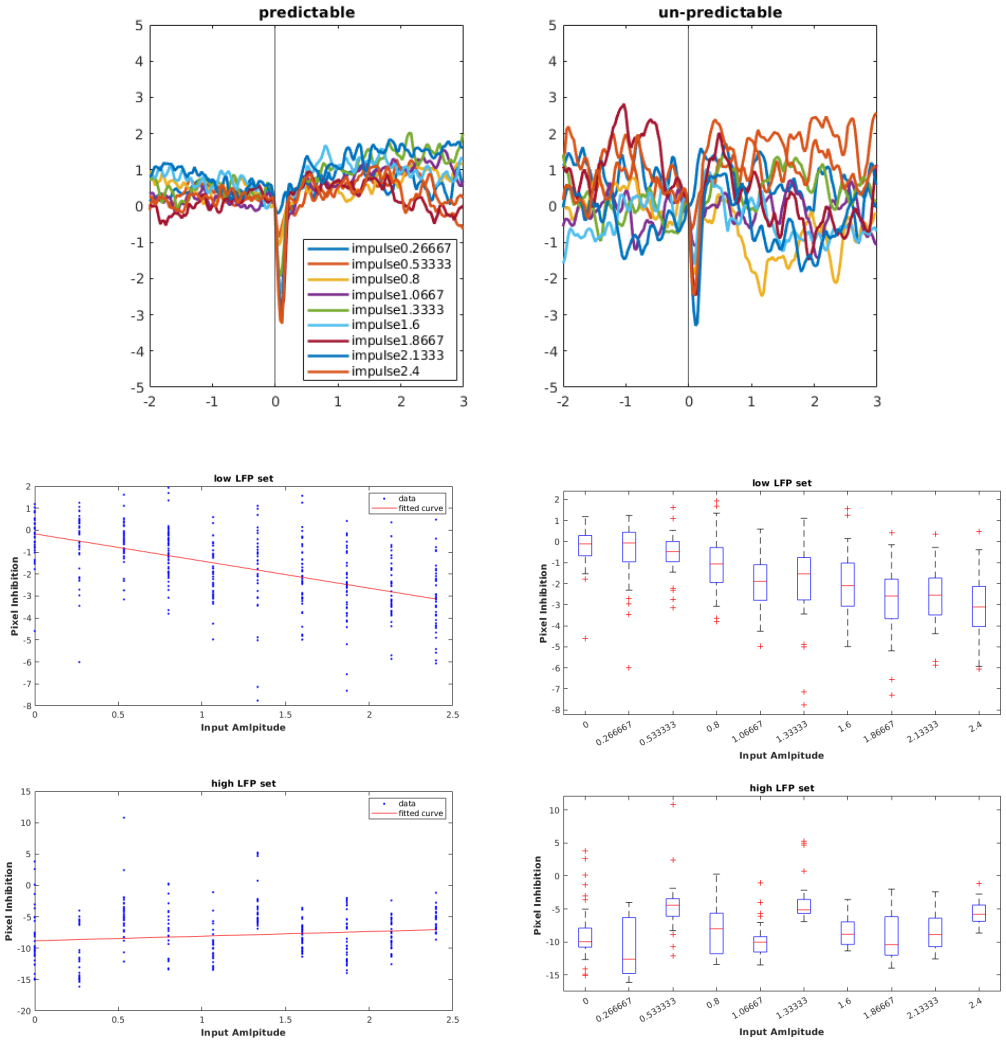
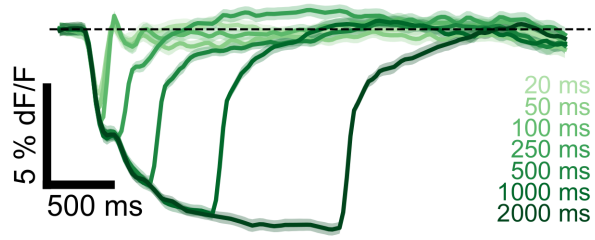


Figure 4.5: Impulse response comparison for signals with low and high concentration of low frequency signal. The low LFP set shows monotonous behavior as compared to high LFP.

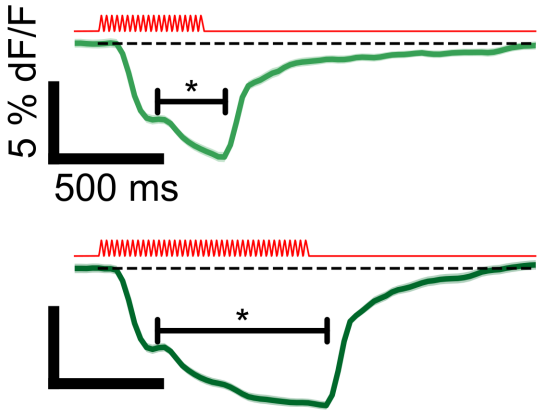
power of low frequency activity is correlated to synchronized/unaroused brain. We observe that under arousal, input actions have monotonous effect as seen in Figure 4.5.

Next we analyze the step response of system to varying level of input inhibition as shown in Figure 4.6.

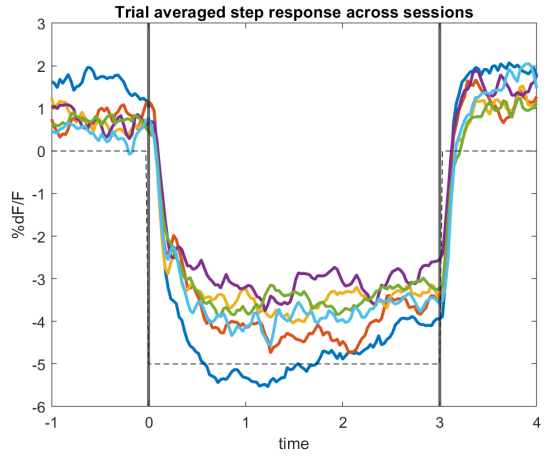
Note that for this system, using modeling methods like frequency sweep or sampling



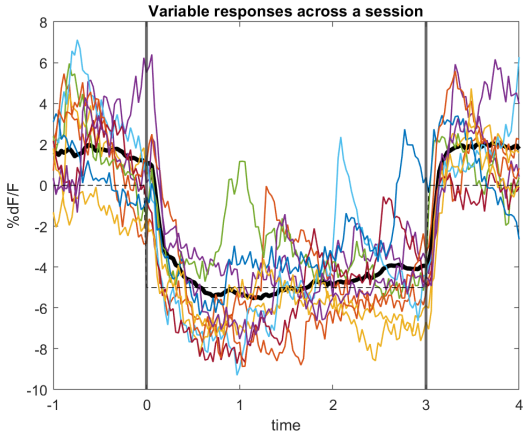
(a) Averaged step response to a sinusoidal activation of varying length of activation.



(b) Step response to a sinusoidal activation



(c) Averaged Response of step inputs across different sessions and mice.



(d) Step response to a sinusoidal activation of constant amplitude for 3 seconds

Figure 4.6: Open loop responses to input stimulation and their trial averaged dynamics with experiments conducted by Matveev [140] and Deole

through persistent excitation is challenging and infeasible in some cases due to physiological input constraints and limit to the amount of data that can be gathered during a session. Moreover, non-stationarity and parameter dependence of input-output characteristics make

system identification through transfer function learning or subspace identification infeasible. We therefore explore a model-free approach to controller design.

4.4 Experiment Setup

The experimental setup and method is described in detail in the work [140]. For context, in our setting, the measurement is derived from wide-field imaging. From this image we select a region of interest and the spatially averaged pixel value represents the output of our system. The input to this system is through optogenetic manipulation. The signal is applied to a fixed part of the brain which is the area of interest. We analyze the system where laser activation location is fixed. The signal used here is a sine-wave at a fixed frequency(35Hz $\approx$ image sampling frequency). The control variable for this signal is the sine-wave amplitude. For consistency, the smallest unit of control(impulse signal) applied is one wavelength.

The goal for initial analysis of control efficacy, is to show that we can track a fixed reference r for inhibition of output state and that a controller can be designed to reject disturbances over a finite horizon. The control objective can be designed as minimization of error norm $\|r - y(t))\|_2$ over $t \in [0, T]$.

4.5 Controller Design and System Analysis

We approach the system design with a model-free design approach. Using results of monotonicity observed from the empirical analysis, we argue that a proportional controller for reference tracking setup will behave as a good regulator for the system.

We have observed that the trial-averaged system exhibits characteristics of a low order system whereas trial by trial samples from the system exhibiting variable responses as seen in Figure 4.6c, 4.6d.

The controller we design is evaluated over its "averaged" tracking performance on the trial window. The trial-to-trial response is treated as disturbance to the average model. The controller is expected to be able to perform disturbance rejection on trial-to-trial basis .The Figure 4.7 shows a closed loop schematic for the averaged system with output feedback

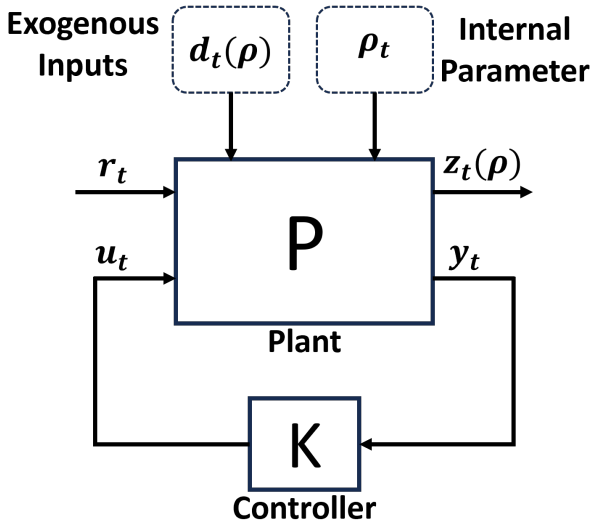


Figure 4.7: Control system for Neuronal dynamics

control. Here system is under the influence of exogenous inputs w_t and we evaluate the L2 norm of controller performance over finite window as $\|z(t)\|_{L_2}$, where $z(t)$ is the tracking error.

For simplicity we assume that a linear system can approximate the dynamics of the subject over a trial. The trial-to-trial variability can be explained by a parameter varying linear system. Here nonlinear models can also be proposed that follow that explain the average and trial to trial behavior but we want to explore set the platform for further research where this system can be analyzed form the lens of linear systems.

Consider the following candidate Linear parameter-varying system:

$$x_{t+1} = A(\rho)x_t + B(\rho)u_t + w_t + d_t \tag{4.3}$$

$$y_t = Cx_t + v_t \tag{4.4}$$

where a candidate model for the system matrices can be expressed as

$$A(\rho) = \sum_{k=1}^{n} \rho_k A_k,$$
$$B(\rho) = \sum_{k=1}^{n} \rho_k B_k.$$

Here the parameter ρ contributes to the variability we observe on a trial-to-trial basis along with the exogenous signal d_t. The apparently arbitrary nature of parameter ρ and disturbances d makes the linear system identification problem to be infeasible. We also know that the observed disturbances are bounded as spontaneous brain activity is bounded. The noise terms w_t, v_t are assumed to be zero mean i.i.d. and the disturbance term is also assumed to be stationary. The parameters ρ are also assumed to be drawn from a stationary distribution[2]. Finally, we also know that the system is marginally stable as the dynamics have oscillatory dynamics.

The assumptions of stationarity of parameters and averaging dynamics can be formulated as:

$$\mathbb{E}_\rho[A(\rho)] = A(\bar{\rho}) := \bar{A}$$
$$\mathbb{E}_\rho[B(\rho)] = B(\bar{\rho}) := \bar{B},$$
$$\mathbb{E}[d] = 0, \quad \mathbb{E}[\rho] = \bar{\rho}$$

We state the averaging assumption as:

Assumption 4.5.1. *The trial-averaged response over a set of feasible initial conditions can be closely approximated as a stationary linear system.*
$\mathbb{E}\left[\Sigma_{x_0}(u, d, \rho)_{0:T-1}\right] = \bar{\Sigma}_{\bar{x}_0} u_{0:T-1}$*, where* $\bar{\Sigma}$ *generates the output sequence for input* $u_{0:T-1}$ *for average dynamics* $\rho = \bar{\rho}, d_k = 0$ *and* $\mathbb{E}[x_0] = \bar{x}_0$ *for system (4.3-4.4). Here* $\rho \in P$ *and* $d \in Q$ *represent the parameter and disturbance set.*

---

[2]The output is a percentage change in pixel trace over the mean trace value denoted by dF/F. The disturbances and parameters are considered to be stationary with respect to the dF/F value.

Here, on a trial-by-trial basis, the dynamics are interpreted as the averaged system injected with exogenous disturbances d and the controller regulates the transient dynamics and rejects disturbances to the average system$(\bar{A}, \bar{B})$. The parameters ρ, that physiologically represent the brain states, are sampled from Q which represents all possible brain states. The probability distribution of this set is unknown and is heavily dependent on experimental parameters and physiology. The disturbances d which represent the brain interactions from other subsystems are drawn from set Q and are also drawn with an unknown distribution as factors like motion and other unmodeled/unobserved activity. We also assume that the estimation of trial-average is computed using enough trials such that the approximated average is close to the true trial average, that is the approximated average is unbiased.

4.5.1 Controller Structure

For manipulating the output(trace of a cortical region) to a reference trajectory, we propose a closed-loop structure for the output feedback controller with a proportional-integral form. A linear feedforward controller(open loop) is designed for the outer loop to derive the nominal input required to track the reference signal.

Open Loop Controller Design (Feedforward Control)

We introduce a linear proportional term for feedforward control defined by:

$$\bar{u}_t = K_{\mathrm{r}}(y_{\mathrm{ref}} - y_t) \tag{4.5}$$

where y_{ref} is the reference signal, y is the output and K_r is the feedforward term. The feedforward term is computed for each subject through a calibration step where trial-averaged open-loop responses are measured. The step inputs are sampled from the feasible input set and each input is repeated for N steps. Given that Assumption 4.5.1 holds, a stationary response can be generated over N trials.

Calibration

We design an experiment to calibrate the linear feedforward controller. The step impulse inputs are sampled evenly spaced feasible inputs between $0 - 2.5$v. Each step input is repeated N times and a randomly applied on a trial. This gives us a data pair (u_i, z_i), where z_i is the max inhibition achieved by trial averaged output sequence $y_{0:T}$. We apply a linear regression step on the dataset (u_i, z_i). The linear feedforward controller must satisfy the constraint $K_r\mathcal{Y} = \mathcal{U}$, where $\mathcal{Y} = [(y_1 - y_{\text{ref}}), \ldots, (y_k - y_{\text{ref}})]$ and $\mathcal{U} = [u_1, \ldots, u_k]$. Then we solve a regression problem that generates the linear feedforward controller $K_r = \mathcal{U}\mathcal{Y}^{\dagger}$. The feedforward controller needs to be generated every time experimental parameters change.

Closed Loop Controller Design (Feedback Controller)

The feedback controller is designed to closely track a desired reference signal. We propose a simple linear feedback controller with a proportional and an integral term. The proportional control term is a linear error correction term. The integral term is introduced to compensate for bias and activation non-linearities under a step response as seen in Figure 4.6c. Therefore, our discrete-time controller has the form:

$$u_t = \bar{u}_t + K_p(y_{\text{ref}} - y_t) + K_i \sum_{k=0}^{t} (y_{\text{ref}} - y_k)dt \tag{4.6}$$

We express this controller as an output feedback controller of the form $u = \bar{u} + Ky$.

Given that Assumption 4.5.1 holds, we define the tracking objective function as the L2 norm for finite horizon which is equivalently the MSE of tracking given by

$$J(K) = \mathbb{E}_{x_0 \sim \mathcal{D}} \left[\sum_{t=0}^{T} \|z(t)\|_2 \;\middle|\; K \right]; \qquad z(t) = y_t - y_{\text{ref}} \tag{4.7}$$

Where $\|.\|$ is the Euclidean norm, K is the output feedback controller applied to generate the sequence $u_{0:T-1}$ with parameters (K_p, K_i). The trials$(t = 0, \ldots, T)$ are randomly instantiated in an experiment so as to randomize the initial condition $x_0 \in \mathcal{D}$ and generate

an unbiased sampling of parameters ρ and disturbances d. We test the efficacy of candidate controllers by running a validation experiment. The expected value of $J(K)$ can be approximated by taking trial average over N trials as $\hat{J}(K)$ defined by:

$$\hat{J}(K) = \sum_{i=1}^{N} \left[\sum_{t=0}^{T} \|z(t)\|_2 \right] \tag{4.8}$$

The assumptions that trial-averaged dynamics are approximately linear, hold true if Eq. (4.8) converges to Eq. (4.7) as N increases. Here, number of trials is a limiting factor for evaluation as we are limited with experiment time; therefore, a biased estimation of the trial-averaged system will not be generated if brain states are not evenly distributed. Brain states are often unmodeled, specifically for this scenario, and change spontaneously. Neuroscience research has led to model identification and correlation studies between brain states and observed data, which can be applied here, but is a non-trivial task in this work due to experimental constraints.

We now look at an optimization approach for controller design.

4.5.2 Controller Optimization

The regularizability of a controller can be improved as compared to manually selected controller, if the controller is optimized to reduce the L2 norm of error as defined in Eq (4.7). Assuming that enough number of trials are available to assume stationarity, we design a zeroth-order optimization algorithm for the average dynamics of the system Eq. (4.3, 4.4).

The landscape of Eq (4.7) is unknown with respect to K as the model is unknown and J or $\nabla_K J$ cannot be computed analytically as experiment length puts a hard constraint on a number of trials. Therefore, gradient approximation is infeasible in this scenario unless stationarity of brain states can be guaranteed. If so then algorithms like model-free gradient descent on LTI systems can be applied [68]. In this experiment though number of trials are limited and the brain is not stationary for given number of trials

Therefore we design a validation experiment to map J as a function of K. This experiment can inform us about feasibility of optimization routines as shown in Figure 4.9.

We design a data driven optimization routine on J landscape using online zero-order information. This is a sampling based approach where controller are sampled using a update rule defined by Algorithm 3.

Consider the averaged dynamics of the system as:

$$x_{t+1} = \bar{A}x_t + \bar{B}u_t \tag{4.9}$$

$$y_t = Cx_t \tag{4.10}$$

Note here that open-loop step response of the system starts to exhibit a nonlinear behavior even under trial-averaged response as shown in Figure 4.6c. This suggests that the load applied to the system produces inconsistent output. The load can decrease over trial length due to internal processes in certain experiments. Moreover, calibration errors or process changes can occur while computing open loop. We assume there exists an integral state constructed using Riemann sum, i.e.

$$\xi_t = \sum_{k=0}^{t} (x_k - x_{\mathbf{ref}})dt$$

. Here for small dt, we have $\xi = \int_0^t (x_s - x_{\mathbf{ref}})ds$. Now the augmented internal system for state $s = [x^\top,\ \xi^\top]^\top$ can be defined as:

$$\begin{bmatrix} x_{t+1} \\ \xi_{t+1} \end{bmatrix} = \begin{bmatrix} \bar{A} & 0 \\ 1 & dt \end{bmatrix} \begin{bmatrix} x_t \\ \xi_t \end{bmatrix} + \begin{bmatrix} \bar{B} \\ 0 \end{bmatrix} u_t \tag{4.11}$$

$$y_t = \begin{bmatrix} C & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ \xi_t \end{bmatrix}. \tag{4.12}$$

The output feedback controller is defined as $y = KC's$, where $K = [Kp, Ki]$ and C' is the output mapping from Eq. (4.12).

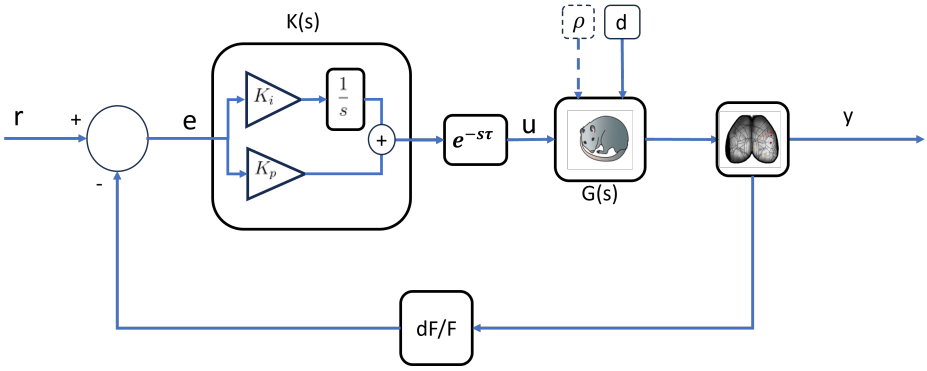


Figure 4.8: Control architecture for Neuronal dynamics

4.6 Controller Analysis

Model-Free Optimization Routine

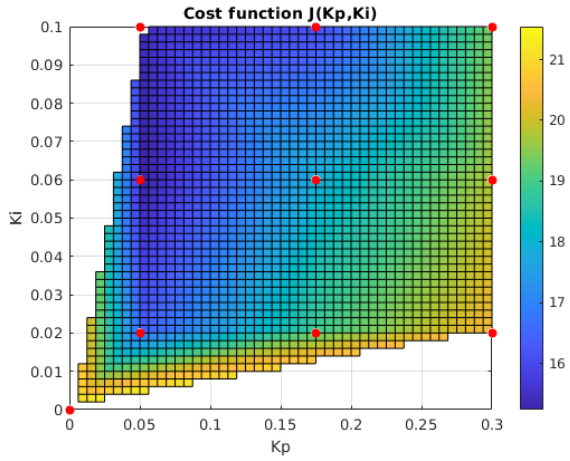


Figure 4.9: Trial averaged performance evaluated for controllers sampled from a feasible space.

We analyze the controller performance over a feasible space by sampling from a grid of

viable controllers and evaluating their performance by computing Eq (4.7). Here the mouse is assumed to follow stationarity rule. The Figure 4.9 shows the landscape of Eq. (4.7) with respect to K for a trial-averaged system. A model free controller optimization routine has been applied here as described in 3. The optimization routine samples a controller $K = [K_p, K_i]$ from the feasible set and evaluates the performance using (4.7). A new candidate controller is then sampled randomly from the surface of a norm ball around K. The size of this ball is the step-size for controller update. The controller is updated if performance is improved. If controller cannot be updated then the step size is reduced by a factor of 0.5.

Algorithm 3 Naive Zero-order Search

Initialize : Controller $K = K_0$, step size = r

for i = 0 to I **do**

$K_i = K_i + rU$ ▷ $U = [K_p, K_i]$ sampled from unit norm ball

for j = 1:N **do**

Sample a trial of trajectory length T

Compute: $C_j = \sum_{t=1}^{T} c_t(x_t, u_t)$

end for

Compute $\hat{J}(K_i) = \frac{1}{N} \sum_{j=1}^{N} C_j$

Update K_i:

$$K_i = \left\{ \begin{array}{ll} K_i, & \text{if } J(K_i) <= J(K_{i-1}) \\ K_{i-1}, & \text{if } J(K_i) > J(K_{i-1}) \end{array} \right\}$$

end for

Here, for N iterations, the approximation regret as given below, must be less than some tolerance level that allows for the optimization landscape to be smooth–so that update steps closely approximate the true descent direction.

$$R(N.K) = \hat{J}(K) - J(K) = \sum_{i=1}^{N} \left[\sum_{t=0}^{T} c_t(x_t, u_t) \;\middle|\; K, x_0 \sim \mathcal{D} \right] - \mathbb{E}[\sum_{t=0}^{T} c_t(x_t, u_t) | K] \quad (4.13)$$

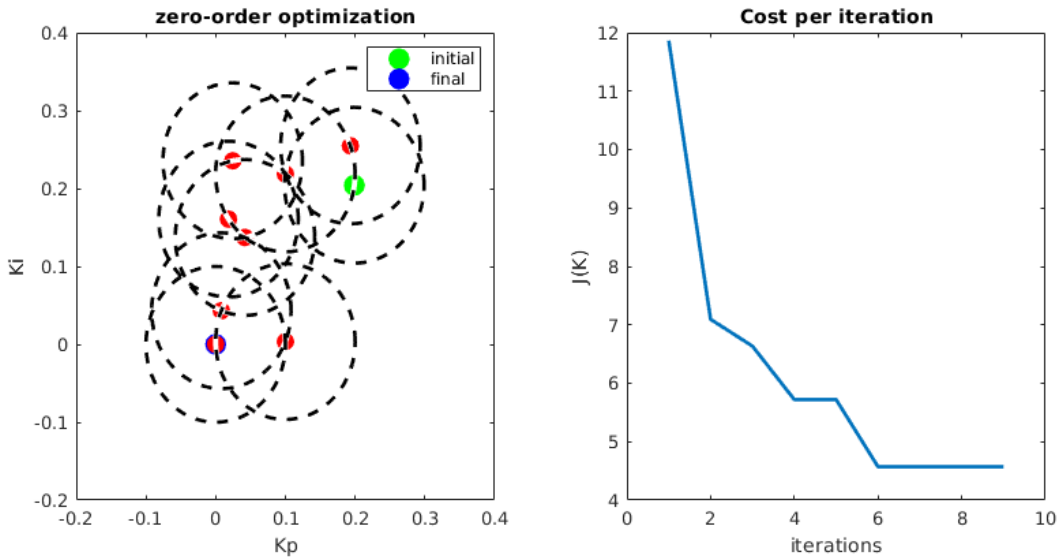


Figure 4.10: Model-free optimization of controller as described in Algorithm 3

Without prior knowledge of uncertainty, we cannot define a minimum value for N such that approximation regret is bounded below the desired threshold. The evaluation of Eq (4.7) for a given controller is often biased if a particular brain state persists for the majority duration of controller evaluation and shifts to a different state while evaluating a different controller. A possible solution to this problem is to use an external signal to force a stationary brain state for the entirety of the experiment. Due to inconsistencies of such a stimulus we do not claim the stationarity of brain state.

The result of session on optimization routine is shown in Figure 4.10

The controller in this set up is designed such that it behaves as a good regulator for the system. The system will be considered a good regulator if tracking performance is improved for closed loop as compared to open loop, that is $\hat{J}(K) \leq \hat{J}(0)$ where $\hat{J}(0)$ is the open loop controller.

Controller Analysis

To analyze controller performance for a subject, we set up the closed loop controller as described in Section 4.5.1. A candidate controller $K = [K_p,\ K_i]$ is selected within the feasible set. The feasible set is defined by input constraints based on the physiology of the subject. The gains are selected such that inputs do not saturate. We run a validation analysis where a comparison between open loop and closed loop controllers is presented. Both the controllers are applied randomly for N trials in each case. We assume that the initial conditions at trial start have been sampled from the bounded set $\mathcal{D}$ which spans the space of all feasible states. Moreover, we assume that for N trials the parameters ρ and system disturbances d are sampled evenly across Q, P. The controller performance is then analyzed by approximating Eq .4.8 for open loop and closed loop controllers.

Definition 4.6.1. *We define a system of the form Eq. 4.3,4.4 to be output regularizable for a controller K if $J(K) \leq J(0)$.*

We show the analysis for an experiment session in Fig. 4.11 where we compare open loop and closed loop performance over trials. Based on experimental result, we observe that closed loop controllers make the system output regularizable. The Fig. 4.11 shows that tracking performance is improved by the controller and the single trial case shows the action of the controller in blue. The Figure 4.13 shows the comparison of variance across trials, which shows that disturbances induced by exogenous interactions, which persistently excite the local dynamics, are significantly subdued in the closed loop case. On a trial-by-trial basis we observe that the closed loop controller is effective at suppressing disturbances as shown in Figure. 4.12, where the worst case performance of closed loop trials is better than open loop trials.

We claim that such a controller that improves the tracking performance over open loop and performs disturbance rejection is a *good regulator* for the system over finite time as it makes the system, output regularizable as stated by Definition 4.6.1. This forms a basis for future study where an internal model principle for finite horizon trials can be formulated.

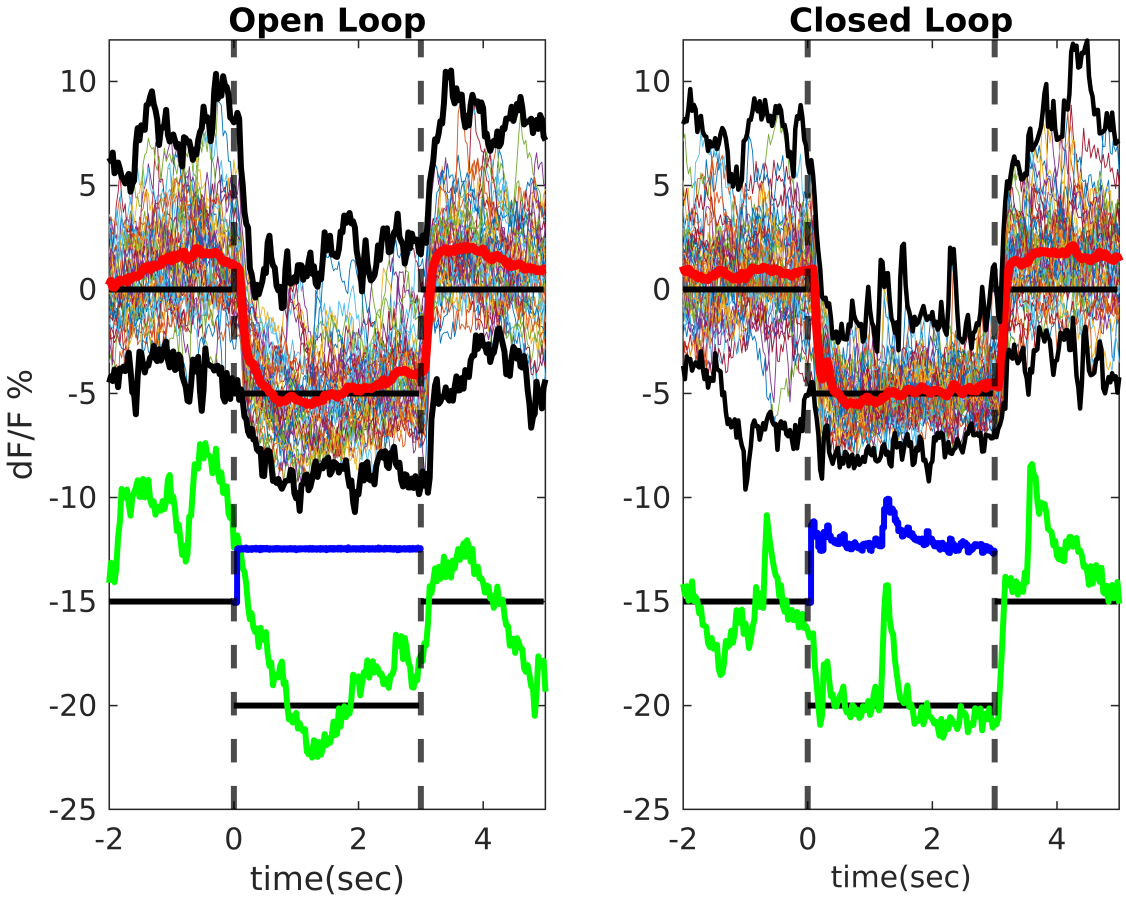


Figure 4.11: Comparison of Tracking control for Feedforward and Feedback scenarios where closed loop controller is a good regulator of the system.

We claim that given that the controller is a good regulator of the system, the controller contains the necessary information about the system and the assumptions made about the stationarity are true.

4.6.1 Controller Performance and Future directions

The controller evaluation across sessions as shown in Figure 4.14. This demonstrates that the general assumption about averaged linear dynamics begin stationary holds true across sessions. Note that this is only true if monotonicity of input-output relationship also holds

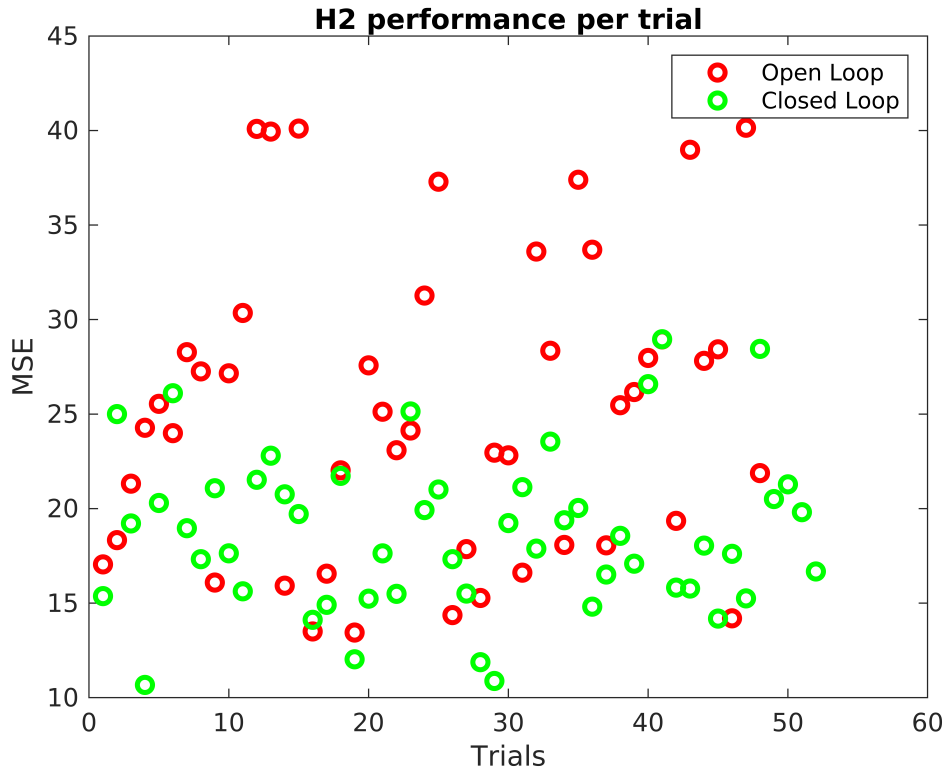


Figure 4.12: Trial by Trial comparison of open loop and closed loop experiments

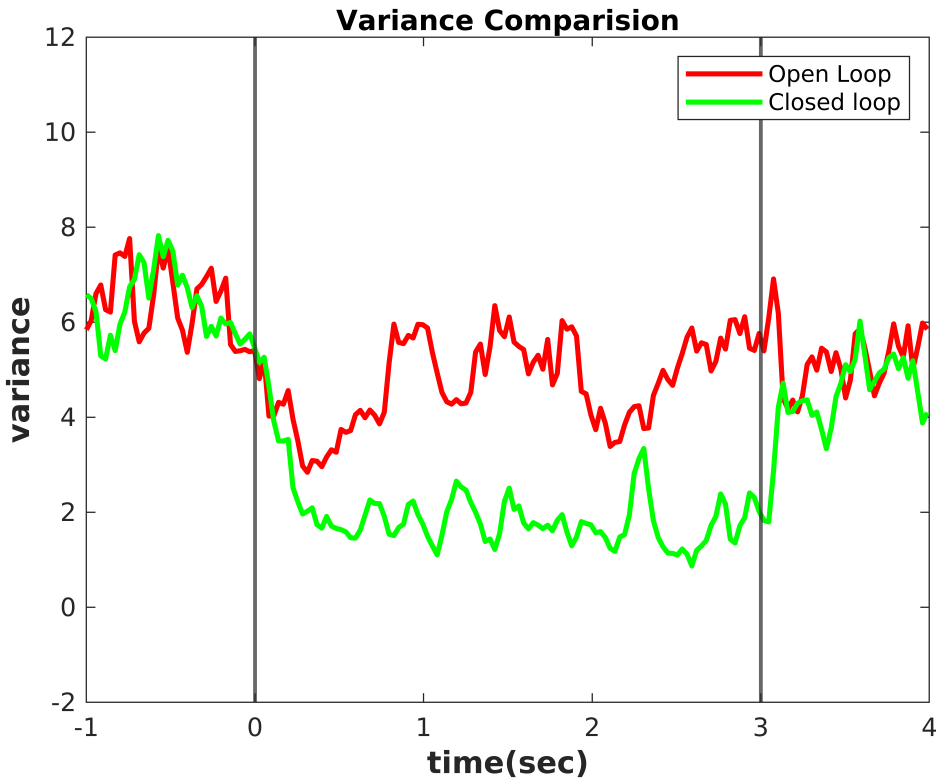


Figure 4.13: Variance Reduction from open loop to closed loop.

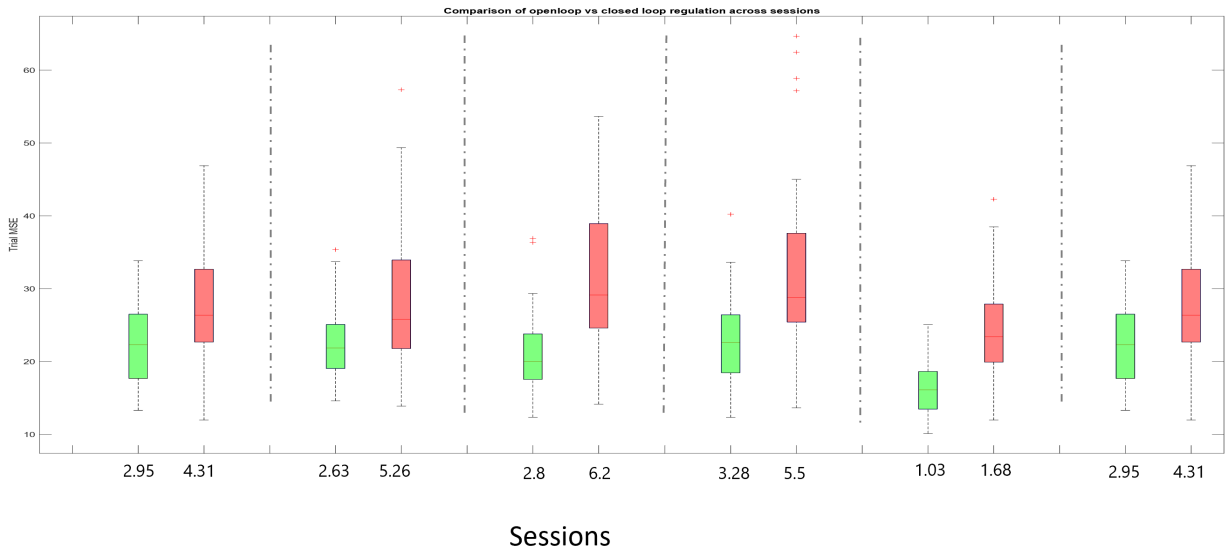


Figure 4.14: Variance Reduction and regulation comparison of controllers across experiment sessions. The boxplot shows trial-by trial statistics with open loop trials in red and closed loop trials in red. The x axis shows total variance across trials for each controller

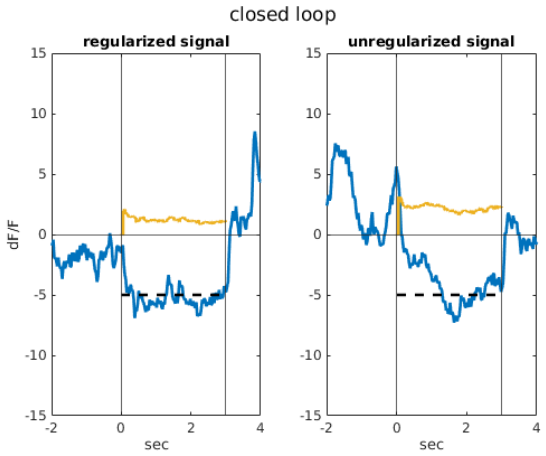
true.

Now we propose that using the model Eq 4.3, 4.4 we can further analyze the system and segregate individual trials into regularizable and non-regularizable based on a threshold value for MSE. In Figure 4.15 we show a comparison of regularizable and un-regularizable trials for both open loop and closed loop case.

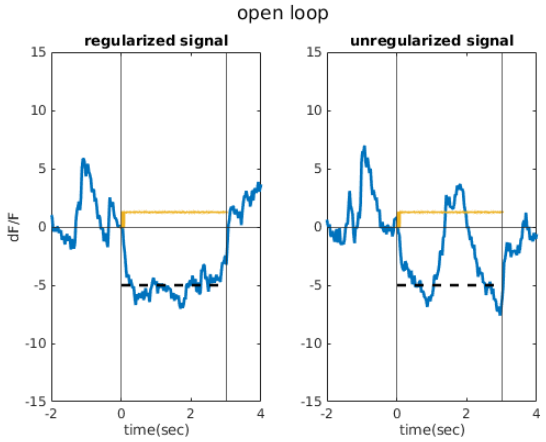
For experiments, we observe that on a trial-to-trial basis the variation introduced by exogenous inputs can exhibit specific properties both under open loop and closed loop setup.

We can see that on a trial-to-trial basis disturbances and parameter dependence can lead to unpredictable behavior even when the controller is a good regulator. We can then justify the model Eq 4.3, 4.4 by expressing variation from the average model as the disturbance or parameter pertaining to a given trial.

As seen in Figure 4.15, we segregated the trials based on MSE as regularizable or non-regularizable. The next direction of this work explores the viability of prediction of reg-



(a) Closed loop regularizability.



(b) Open loop regularizability.

Figure 4.15: Comparing Regularizability of controller per trial

ularizability on a trial-to-trial basis. If such a predictive model can be generated, then a model-based predictive controller can be designed to regularize an otherwise unregularizable trial. Thereby, further decreasing the trial-to-trial variability of the system. Our ongoing work attempts to address those questions and we try to build a model for internal dynamics of the cortex as described by the system in Figure:4.7. The scope of this chapter was to ex-

plore modeling approach for the cortical dynamics such that a controller could be designed to regulate local signal in a region of interest. Further applications to this work will explore trajectory tracking using the internal low-level controller used in this work. The trajectory generation could then be related to some behavioral marker that improves task accuracy in the subject or could be related to a bio-marker for neuronal deficiency that needs to be "stabilized" using desired trajectory.

BIBLIOGRAPHY

[1] *Kalman filter generalizations*, chapter 7, pages 183–227. John Wiley Sons, Ltd, 2006.

[2] *Least squares estimation*, chapter 3, pages 79–105. John Wiley Sons, Ltd, 2006.

[3] *Information Systems Development: Towards a Service Provision Society*. Springer US, 2010.

[4] Markus W. Achtelik, Simon Lynen, Stephan Weiss, Margarita Chli, and Roland Siegwart. Motion- and uncertainty-aware path planning for micro aerial vehicles. *Journal of Field Robotics*, 31(4):676–698, 2014.

[5] Behçet Açıkmeşe, John M Carson III, and David S Bayard. A robust model predictive control algorithm for incrementally conic uncertain/nonlinear systems. *International Journal of Robust and Nonlinear Control*, 21(5):563–590, 2011.

[6] Mehdi Aghagolzadeh and Wilson Truccolo. Latent state-space models for neural decoding. *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2014:3033–3036, 2014.

[7] Atiye Alaeddini and Kristi A. Morgansen. Trajectory design for a nonlinear system to insure observability. In *2014 European Control Conference (ECC)*, pages 2520–2525, 2014.

[8] Atiye Alaeddini, Kristi A. Morgansen, and Mehran Mesbahi. Augmented state feedback for improving observability of linear systems with nonlinear measurements. *Systems & Control Letters*, 133:104520, 2019.

[9] A. Alessandri and P. Coletta. Switching observers for continuous-time and discrete-time linear systems. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, volume 3, pages 2516–2521 vol.3, 2001.

[10] Kyle T. Alfriend, Srinivas R. Vadali, Pini Gurfil, Jonathan P. How, and Louis S. Breger. *Chapter 5 - Linear Equations of Relative Motion*. Butterworth-Heinemann, Oxford, 2010.

[11] Faraj Alhwarin, Alexander Ferrein, and Ingrid Scholl. IR stereo kinect: Improving depth images by combining structured light with IR stereo. *Lecture Notes in Computer*

Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8862:409–421, 2014.

[12] Alexander Amini, Wilko Schwarting, et al. Deep evidential regression. In *NeurIPS*, 2020.

[13] Brandon Anderson et al. Certified robustness via locally biased randomized smoothing. In *AISTATS*, 2022.

[14] Travis Askham and J. Nathan Kutz. Variable projection methods for an optimized dynamic mode decomposition, 2017.

[15] Karl Johan Astrom and T Hagglund. *PID controllers*. ISA, Research Triangle Park, 2 edition, June 1995.

[16] K.J. Åström and R. Murray. *Feedback Systems: An Introduction for Scientists and Engineers, Second Edition*. Princeton University Press, 2021.

[17] Trevor Avant and Kristi A. Morgansen. Observability Properties of Object Pose Estimation. In *2019 American Control Conference (ACC)*, pages 5134–5140, 2019.

[18] Behçet Açikmese and Martin J. Corless. Stability analysis with quadratic lyapunov functions: Some necessary and sufficient multiplier conditions. *Syst. Control. Lett.*, 57:78–94, 2008.

[19] A. Balluchi, A. Bicchi, A. Balestrino, and G. Casalino. Path tracking control for dubin's cars. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3123–3128 vol.4, 1996.

[20] Jonathan Becktor, William Seto, Aditya Deole, Saptarshi Bandyopadhyay, Niyousha Rahimi, shahriar Talebi, Mehran Mesbahi, and Amir Rahmani. Robust vision-based multi-spacecraft guidance navigation control using cnn-based pose estimation. In *2022 IEEE Aerospace Conference*, pages –. IEEE, 2022.

[21] Viktor Bengs et al. Are uncertainty quantification capabilities of evidential deep learning reliable? *OpenReview*, 2025. ICLR submission.

[22] Akshara Bhatia, Aravind Rajeswaran, et al. Failure prediction with statistical guarantees for vision-based policies. In *Robotics: Science and Systems (RSS)*, 2022.

[23] Lars Blackmore, Senthooran Rajamanoharan, and Brian C. Williams. Active estimation for jump markov linear systems. *IEEE Transactions on Automatic Control*, 53(10):2223–2236, 2008.

[24] M F Bolus, A A Willats, C J Rozell, and G B Stanley. State-space optimal feedback control of optogenetically driven neural activity. *Journal of Neural Engineering*, 18(3):036006, mar 2021.

[25] Christophe Bonnal, Jean-Marc Ruault, and Marie-Christine Desjean. Active debris removal: Recent progress and current trends. *Acta Astronautica*, 85:51–60, 2013.

[26] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, January 1994.

[27] Rohith Boyinine, Rajnikant Sharma, and Kevin Brink. Observability based path planning for multi-agent systems to aid relative pose estimation. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 912–921, 2022.

[28] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[29] Yannik Bukschat and Marco Vetter. Efficientpose: Efficient human pose estimation with neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 0–1, 2021.

[30] Ali Tevfik Büyükkoçak, Yingjie Hu, Azizollah Taheri, Derya Aksaray, and Demoz Gebre-Egziabher. *State-Estimation-Aware Planning for Autonomous Systems with Temporal Logic Specifications*, chapter 1, page 1. AIAA, 2023.

[31] Lorenzo Pasqualetto Cassinis, Robert Fonod, Eberhard Gill, Ingo Ahrns, and Jesus Gil Fernandez. Cnn-based pose estimation system for close-proximity operations around uncooperative spacecraft. In *AIAA Scitech 2020 Forum*, page 1, 2020.

[32] Bo Chen, Jiewei Cao, Alvaro Parra, and Tat Jun Chin. Satellite pose estimation with deep landmark regression and nonlinear pose refinement. *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, pages 2816–2824, 2019.

[33] F.L. Chernousko. Ellipsoidal state estimation for dynamical systems. *Nonlinear Analysis: Theory, Methods and Applications*, 63(5):872–879, 2005. Invited Talks from the Fourth World Congress of Nonlinear Analysts (WCNA 2004).

[34] Ping-yeh Chiang, Michael J. Curry, Ahmed Abdelkader, Aounon Kumar, John Dickerson, and Tom Goldstein. Detection as regression: Certified object detection by median smoothing. *arXiv:2007.03730*, 2020.

[35] Ping-yeh Chiang, Michael J. Curry, Ahmed Abdelkader, Aounon Kumar, John P. Dickerson, and Tom Goldstein. Certified object detection by median smoothing. In *NeurIPS*, 2020.

[36] Alaa Eddine Chriat and Chuangchuang Sun. High-order control lyapunov–barrier functions for real-time optimal control of constrained non-affine systems. *Mathematics*, 12(24), 2024.

[37] Moo K. Chung. Gaussian kernel smoothing, 2020.

[38] Mark M. Churchland, John P. Cunningham, Matthew T. Kaufman, Justin D. Foster, Paul Nuyujukian, Stephen I. Ryu, and Krishna V. Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, June 2012.

[39] WH Clohessy and RS Wiltshire. Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, 27(9):653–658, 1960.

[40] WH Clohessy and RS Wiltshire. Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, 27(9):653–658, 1960.

[41] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust Region Methods*, chapter 7, page 169–248. Society for Industrial and Applied Mathematics, January 2000.

[42] Ryan Cosner et al. Measurement-robust control barrier functions: Certainty in safety despite uncertainty in state. Technical report, Caltech, 2021.

[43] Matteo Cucchi, Steven Abreu, Giuseppe Ciccone, Daniel Brunner, and Hans Kleemann. Hands-on reservoir computing: a tutorial for practical implementation. *Neuromorphic Computing and Engineering*, 2(3):032002, aug 2022.

[44] John Cunningham and Byron Yu. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17, 08 2014.

[45] John P Cunningham and Byron M Yu. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17(11):1500–1509, August 2014.

[46] Christopher W. Curtis, D. Jay Alford-Lago, Erik Bollt, and Andrew Tuma. Machine learning enhanced hankel dynamic-mode decomposition, 2023.

[47] Inc. CVX Research. CVX: Matlab software for disciplined convex programming, version 2.0. `https://cvxr.com/cvx`, August 2012.

[48] Luis D'Alto and Martin Corless. Incremental quadratic stability. *Numerical Algebra, Control & Optimization*, 3(1):175, 2013.

[49] Simone D'Amico. ESA pose estimation challenge, July 2019.

[50] Sarah Dean, Nikolai Matni, Benjamin Recht, and Vickie Ye. Robust guarantees for perception-based control. *Proceedings of Machine Learning Research*, pages 1–21, 2019.

[51] Sarah Dean, Nikolai Matni, Benjamin Recht, and Vickie Ye. Robust guarantees for perception-based control. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger, editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 350–360. PMLR, 10–11 Jun 2020.

[52] Sarah Dean, Stephen Tu, Nikolai Matni, and Benjamin Recht. Safely Learning to Control the Constrained Linear Quadratic Regulator. pages 1–27, 2018.

[53] Steve Dean et al. Robust safety under stochastic uncertainty with discrete-time control barrier functions. In *Robotics: Science and Systems (RSS)*, 2023.

[54] Yu Deng, Jiahong Xue, Teng Cao, Yingxing Zhang, Lanxi Wen, and Yiyang Chen. Superpose: Improved 6d pose estimation with robust tracking and mask-free initialization. *arXiv preprint arXiv:2409.19986*, 2024.

[55] M. Denninger et al. Blenderproc: Reducing the reality gap by procedural scene generation. *arXiv:1911.01911*, 2019.

[56] Aditya Deole, Beniamino Pozzan, and Mehran Mesbahi. Mpc-based estimation-aware trajectory generation for uncontrolled satellite pose tracking. *AIAA SCITECH 2024 Forum*, January 2024.

[57] Aditya Deole, Shahriar Talebi, Spencer Kraisler, Niyousha Rahimi, Mehran Mesbahi, Saptarshi Bandyopadhyay, Vinod P. Gehlot, William Seto, Amir Rahmani, and Jonathan Becktor. Multi-agent passivity-based control for perception-based guidance. 2023.

[58] Yan Di, Fabian Manhardt, Gu Wang, Xiangyang Ji, Nassir Navab, and Federico Tombari. SO-Pose: Exploiting Self-Occlusion for Direct 6D Pose Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12396–12405, October 2021.

[59] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[60] Yu Ding, Yirui Cong, and Xiangke Wang. Set-membership filtering-based cooperative state estimation for multi-agent systems. In *2023 42nd Chinese Control Conference (CCC)*, pages 5791–5796, 2023.

[61] A. L. Dontchev and W. W. Hager. An Inverse Mapping Theorem for Set-Valued Maps. *Proceedings of the American Mathematical Society*, 121(2):481–489, 1994.

[62] A. L. Dontchev and A. S. Lewis. Perturbations and metric regularity. *Set-Valued Analysis*, 13(4):417–438, December 2005.

[63] A. Dosovitskiy et al. Carla: An open urban driving simulator. In *Conference on Robot Learning (CoRL)*, 2017.

[64] O. Egeland and J.-M. Godhavn. Passivity-based adaptive attitude control of a rigid spacecraft. *IEEE Transactions on Automatic Control*, 39(4):842–846, 1994.

[65] Alex Ellery, Joerg Kreisel, and Bernd Sommer. The case for robotic on-orbit servicing of spacecraft: Spacecraft reliability is a myth. *Acta Astronautica*, 63(5):632–648, 2008.

[66] Jin Erdong and Sun Zhaowei. Passivity-based control for a flexible spacecraft in the presence of disturbances. *International Journal of Non-Linear Mechanics*, 45(4):348–356, 2010.

[67] Zhaoxin Fan, Yazhi Zhu, Yulin He, Qi Sun, Hongyan Liu, and Jun He. Deep learning on monocular object pose detection and tracking: A comprehensive overview. *ACM Comput. Surv.*, 55(4), November 2022.

[68] Maryam Fazel, Rong Ge, Sham M. Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for linearized control problems. *CoRR*, abs/1801.05039, 2018.

[69] Mahyar Fazlya, Alexander Robey, Hamed Hassani, Manfred Morari, and George J. Pappas. Eficient and Accurate Estimation of Lipschitz Constants for deep neural networks. *arXiv*, (NeurIPS), 2019.

[70] Mahyar Fazlyab, Manfred Morari, and George J. Pappas. Probabilistic Verification and Reachability Analysis of Neural Networks via Semidefinite Programming. *Proceedings of the IEEE Conference on Decision and Control*, 2019-Decem:2726–2731, 2019.

[71] Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George J. Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. *CoRR*, abs/1906.04893, 2019.

[72] Yu Feng, He Zhen, Qiao Bing, and Yu Xiaoting. Stereo-vision-based relative pose estimation for the rendezvous and docking of non-cooperative satellites. *Mathematical Problems in Engineering*, 2014.

[73] A. Fernandez-Ruiz, A. Oliva, G. A. Nagy, A. P. Maurer, A. Berenyi, K. Mizuseki, and G. Buzsaki. Long-duration hippocampal sharp wave ripples improve memory. *Science*, 364:1082–1086, 2018.

[74] N. Firfilionis, J. Smith, K. Lee, and A. Thomas. Cando: Closed-loop optogenetic neuromodulation platform. *Journal of Neural Engineering*, 18(4):046021, 2021.

[75] R. A. Fisher. Theory of statistical estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 22(5):700–725, 1925.

[76] Emilien Flayac, Karim Dahia, Bruno Hérissé, and Frédéric Jean. Nonlinear fisher particle output feedback control and its application to terrain aided navigation. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1566–1571, 2017.

[77] Hélène Frankowska and Marc Quincampoix. Hölder metric regularity of set-valued maps. *Mathematical Programming*, 132(1–2):333–354, August 2010.

[78] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, 2012.

[79] Xiao Shan Gao, Xiao Rong Hou, Jianliang Tang, and Hang Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 2003.

[80] V. Gilja, P. Nuyujukian, C. A. Chestek, J. P. Cunningham, B. M. Yu, J. M. Fan, M. M. Churchland, M. T. Kaufman, J. C. Kao, S. I. Ryu, and K. V. Shenoy. A high-performance neural prosthesis enabled by control algorithm design. *Nature Neuroscience*, 15:1752–1757, 2012.

[81] G. Girardeau, K. Benchenane, S. I. Wiener, G. Buzsaki, and M. B. Zugaro. Selective suppression of hippocampal ripples impairs spatial memory. *Nature Neuroscience*, 12:1222–1223, 2009.

[82] Thomas Glotzbach, Naveena Crasta, and Christoph Ament. Observability analyses and trajectory planning for tracking of an underwater robot using empirical gramians. *IFAC Proceedings Volumes*, 47(3):4215–4221, 2014.

[83] Ananya Gowal et al. Robust control barrier functions using uncertainty estimation with matched and unmatched uncertainties. *arXiv:2401.01881*, 2024.

[84] Christopher Grebe, Emmett Wise, and Jonathan Kelly. Observability-aware trajectory optimization: Theory, viability, and state of the art. In *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 1–8, 2021.

[85] Michael Greenacre, Patrick J F Groenen, Trevor Hastie, Alfonso Iodice D'Enza, Angelos Markos, and Elena Tuzhilina. Principal component analysis. *Nat. Rev. Methods Primers*, 2(1), December 2022.

[86] Tristan D. Griffith and James E. Hubbard. System identification methods for dynamic models of brain activity. *Biomedical Signal Processing and Control*, 68:102765, 2021.

[87] Tristan D. Grifith, Vinod P. Gehlot, and Mark J. Balas. Adaptive estimation of unknown inputs with weakly nonlinear dynamics. In *2022 American Control Conference (ACC)*, pages 5043–5049, 2022.

[88] Tristan D. Grifith, Vinod P. Gehlot, and Mark J. Balas. Adaptive estimation of unknown inputs with weakly nonlinear dynamics. In *2022 American Control Conference (ACC)*, pages 5043–5049, 2022.

[89] Marshel J. H. Grosenick, L. and K. Deisseroth. Closed-loop and activity-guided optogenetic control. *Neuron*, 86(1):106–139, 2015.

[90] Z. V. Guo, N. Li, D. Huber, E. Ophir, D. A. Gutnisky, J. T. Ting, G. Feng, and K. Svoboda. Orca: Online real-time calcium imaging analysis. *Nature Methods*, 16:213–222, 2019.

[91] A. Gupta, M. Sharma, and G. Rothschild. Auditory feedback controls motor cortical activity in mice. *Nature Communications*, 13:900, 2022.

[92] Fredrik Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7):53–82, 2010.

[93] Kenneth D. Harris and Alexander Thiele. Cortical state and attention. *Nature Reviews Neuroscience*, 12(9):509–523, August 2011.

[94] Alexei Harvard, Vincenzo Capuano, Eugene Y. Shao, and Soon-Jo Chung. Spacecraft pose estimation from monocular images using neural network based keypoints and visibility maps. In *AIAA Scitech 2020 Forum*, 2020.

[95] Alexei Harvard, Vincenzo Capuano, Eugene Y. Shao, and Soon-Jo Chung. Spacecraft pose estimation from monocular images using neural network based keypoints and visibility maps. In *AIAA Scitech 2020 Forum*, page 1, 2020.

[96] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[97] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):386–397, 2020.

[98] R. Hermann and A. Krener. Nonlinear Controllability and Observability. *IEEE Transactions on Automatic Control*, 22(5):728–740, 1977.

[99] James Hobbs, Sam Coogan, and Jeremy Weimer. Run time assurance for safety-critical systems. *IEEE Control Systems Magazine*, 2022.

[100] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. van der Smagt, and J. P. Donoghue. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485:372–375, 2012.

[101] Lars Hörmander. *The Analysis of Linear Partial Differential Operators I*. Springer Berlin Heidelberg, 2003.

[102] Haimin Hu, Mahyar Fazlyab, Manfred Morari, and George J. Pappas. Reach-SDP: Reachability Analysis of Closed-Loop Systems with Neural Network Controllers via Semidefinite Programming. *arXiv*, pages 1–15, 2020.

[103] Haimin Hu, Mahyar Fazlyab, Manfred Morari, and George J. Pappas. Reach-sdp: Reachability analysis of closed-loop systems with neural network controllers via semidefinite programming, 2020.

[104] Elina A.K. Jacobs, Nicholas A. Steinmetz, Andrew J. Peters, Matteo Carandini, and Kenneth D. Harris. Cortical state fluctuations during sensory decision making. *Current Biology*, 30(24):4944–4955.e7, 2020.

[105] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[106] Marine Jouin, Rafael Gouriveau, Daniel Hissel, Marie-Cécile Péra, and Noureddine Zerhouni. Particle filter-based prognostics: Review, discussion and perspectives. *Mechanical Systems and Signal Processing*, 72-73:2–31, 2016.

[107] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(1):35–45, 03 1960.

[108] Patrick W. Kenneally, Scott Piggott, and Hanspeter Schaub. Basilisk: A flexible, scalable and modular astrodynamics simulation framework. *Journal of Aerospace Information Systems*, 17(9):496–507, 2020.

[109] Hassan K Khalil. *Nonlinear Systems*, volume 3. Upper Saddle River, NJ: Prentice hall, 2002.

[110] Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5562–5571. PMLR, 18–24 Jul 2021.

[111] Arthur J. Krener and Kayo Ide. Measures of Unobservability. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 6401–6406, 2009.

[112] Clemens Kreutz and Jens Timmer. *Encyclopedia of Systems Biology*, chapter Optimal Experiment Design, Fisher Information, pages 1576–1579. Springer New York, New York, NY, 2013.

[113] Vikram Krishnamurthy and Dejan V. Djonin. Structured threshold policies for dynamic sensor scheduling—a partially observed markov decision process approach. *IEEE Transactions on Signal Processing*, 55(10):4938–4957, 2007.

[114] E. Krook-Magnuson, C. Armstrong, A. D. Bui, S. E. Lew, J. O. McNamara, and I. Soltesz. On-demand optogenetic control of spontaneous seizures in temporal lobe epilepsy. *Nature Communications*, 4:1376, 2013.

[115] J. Nathan Kutz, Xing Fu, and Steven L. Brunton. Multi-resolution dynamic mode decomposition, 2015.

[116] Sanjay Lall, Jerrold E. Marsden, and Sonja Glavaški. Empirical model reduction of controlled nonlinear systems. *IFAC Proceedings Volumes*, 32(2):2598–2603, 1999.

[117] Taeyeop Lee, Bowen Wen, Minjun Kang, Gyuree Kang, In So Kweon, and Kuk-Jin Yoon. Any6d: Model-free 6d pose estimation of novel objects. *arXiv preprint arXiv:2503.18673*, 2025.

[118] Jing Lei. Passivity-Based Output Feedback Control for Systems With Nonlinear Outputs Via High-Gain Observers. *Journal of Dynamic Systems, Measurement, and Control*, 144(4), 01 2022. 041002.

[119] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*, 81(2), 2009.

[120] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep Iterative Matching for 6D Pose Estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[121] Yuxin Li, Gu Wang, Xiangyu Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018.

[122] Zelin Li, Gu Wang, Xiangyu Ji, Yu Xiang, and Dieter Fox. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 7678–7687, 2019.

[123] C. F. Lillie. On-orbit assembly and servicing of future space observatories. In John C. Mather, Howard A. MacEwen, and Mattheus W. M. de Graauw, editors, *Space Telescopes and Instrumentation I: Optical, Infrared, and Millimeter*, volume 6265, pages 767 – 778. International Society for Optics and Photonics, SPIE, 2006.

[124] Lars Lindemann, Matthew Cleaveland, Yiannis Kantaros, and George J. Pappas. Robust motion planning in the presence of estimation uncertainty. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 5205–5212, 2021.

[125] Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J. Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 2023.

[126] S. Little, A. Pogosyan, S. Neal, B. Zavala, L. Zrinzo, M. Hariz, T. Foltynie, P. Limousin, K. Ashkan, J. FitzGerald, A. Green, T. Aziz, and P. Brown. Adaptive deep brain stimulation in advanced parkinson disease. *Annals of Neurology*, 74(3):449–457, 2013.

[127] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O. Hero III, and Pramod K. Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.

[128] Yifan Liu et al. Sam-6d: Segment anything for zero-shot 6d object pose estimation. *arXiv preprint arXiv:2311.15707*, 2023.

[129] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017.

[130] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[131] L. Lux, C. Bernhard, A. Lörincz, and J. Csicsvari. Content-specific disruption of hippocampal replay during sleep impairs memory consolidation. *Neuron*, 105:485–493, 2020.

[132] J.M. Maciejowski. Guaranteed stability with subspace methods. *Systems and Control Letters*, 26(2):153–156, 1995.

[133] V. Makoviychuk et al. Isaac gym: High performance gpu-based physics simulation for robotics. *arXiv:2108.10470*, 2021.

[134] Danylo Malyuta, Taylor P. Reynolds, Michael Szmuk, Thomas Lew, Riccardo Bonalli, Marco Pavone, and Behçet Açıkmeşe. Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently. *IEEE Control Systems Magazine*, 42(5):40–113, 2022.

[135] Yuanqi Mao, Daniel Dueri, Michael Szmuk, and Behçet Açıkmeşe. Successive convexification of non-convex optimal control problems with state constraints. *IFAC-PapersOnLine*, 50(1):4063–4069, 2017.

[136] Yuanqi Mao, Daniel Dueri, Michael Szmuk, and Behçet Açıkmeşe. Successive convexification of non-convex optimal control problems with state constraints. *IFAC-PapersOnLine*, 50(1):4063–4069, 2017. 20th IFAC World Congress.

[137] Yuanqi Mao, Michael Szmuk, and Behçet Açıkmeşe. Successive convexification of non-convex optimal control problems and its convergence properties. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 3636–3641, 2016.

[138] F. Marbach and A. Zador. Directed, but not random, feedback shapes neural coding in cortex. *Nature Neuroscience*, 21:184–193, 2018.

[139] A. Marmont and K. S. Cole. The voltage clamp technique. *The Journal of General Physiology*, 32:567–588, 1949.

[140] Pascha Matveev, Anna J. Li, Zhiwen Ye, Anna J. Bowen, Ximena Opitz-Araya, Jonathan T. Ting, and Nicholas A. Steinmetz. Simultaneous mesoscopic measurement and manipulation of mouse cortical activity. *bioRxiv*, 2024.

[141] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.

[142] Henrique M. T. Menegaz, João Y. Ishihara, Geovany A. Borges, and Alessandro N. Vargas. A systematization of the unscented kalman filter theory. *IEEE Transactions on Automatic Control*, 60(10):2583–2598, 2015.

[143] Varun Murali, Igor Spasojevic, Winter Guerra, and Sertac Karaman. Perception-aware trajectory generation for aggressive quadrotor flight using differential flatness. In *American Control Conference*, pages 3936–3943, 2019.

[144] M. Nallasamy, Sanjit A. Seshia, et al. Soter: A runtime assurance framework for programming safe robotics systems. In *DSN*, 2019.

[145] Seiied Saeed Nasrolahi and Farzaneh Abdollahi. Lyapunov stability analysis for non-linear satellite attitude control in the presence of states measurement error. In *2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA)*, pages 64–68, 2016.

[146] Tuan Nguyen et al. Yolopose v2: Transformer-based end-to-end 6d object pose estimation. *arXiv preprint arXiv:2307.11550*, 2023.

[147] Van Nguyen Nguyen, Stephen Tyree, Andrew Guo, Mederic Fourmy, Anas Gouda, Taeyeop Lee, Sungphill Moon, Hyeontae Son, Lukas Ranftl, Jonathan Tremblay, Eric Brachmann, Bertram Drost, Vincent Lepetit, Carsten Rother, Stan Birchfield, Jiri Matas, Yann Labbe, Martin Sundermeyer, and Tomas Hodan. Bop challenge 2024 on model-based and model-free 6d object pose estimation, 2025.

[148] S. R. O. Nilsson, N. L. Goodwin, J. J. Choong, S. Hwang, H. R. Wright, C. D. Norona, A. Dutta, J. W. Shaevitz, and A. M. Leifer. Real-time markerless posture tracking for behavioral experiments. *Nature Methods*, 17:82–89, 2020.

[149] Y. Nishimura, S. I. Perlmutter, and E. E. Fetz. Restoration of upper limb movement via artificial cortico-spinal-spinal connection after spinal cord injury. *Nature Communications*, 4:1972, 2013.

[150] Raimund J. Ober. The Fisher information matrix for linear systems. *Systems and Control Letters*, 47(3):221–226, 2002.

[151] Kazuhide Okamoto and Panagiotis Tsiotras. Optimal stochastic vehicle path planning using covariance steering. *IEEE Robotics and Automation Letters*, 4(3):2276–2281, 2019.

[152] I. Opris, L. Santos, G. A. Gerhardt, R. E. Hampson, and S. A. Deadwyler. Closed-loop stimulation of prefrontal cortex improves task performance. *Neurobiology of Learning and Memory*, 97:396–403, 2012.

[153] A. M. Packer, L. E. Russell, H. W. Dalgleish, and M. Häusser. Simultaneous all-optical manipulation and recording of neural circuit activity with cellular resolution in vivo. *Nature Methods*, 12(2):140–146, 2015.

[154] Shiye Pan and Xinmei Wang. A survey on perspective-n-point problem. In *2021 40th Chinese Control Conference (CCC)*, pages 2396–2401, 2021.

[155] Kiru Park, Timothy Patten, and Markus Vincze. Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[156] Patricia Pauli, Dennis Gramlich, Julian Berberich, and Frank Allgöwer. Linear systems with neural network nonlinearities: Improved stability analysis via acausal zames-falb multipliers, 2021.

[157] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4561–4570, 2019.

[158] Lucas Pinto, Michael J Goard, Daniel Estandian, Min Xu, Alex C Kwan, Seung-Hee Lee, Thomas C Harrison, Guoping Feng, and Yang Dan. Fast modulation of visual perception by basal forebrain cholinergic neurons. *Nature Neuroscience*, 16(12):1857–1863, October 2013.

[159] Angel Ricardo Plastino and Angelo Plastino. What's the Big Idea? Cramér–Rao Inequality and Rao Distance. *Significance*, 17(4):39–39, 07 2020.

[160] Nathan D. Powel and Kristi A. Morgansen. Empirical observability gramian rank condition for weak observability of nonlinear systems with control. In *2015 54th IEEE Conference on Decision and Control (CDC)*, volume 1, pages 6342–6348, 2015.

[161] A. A. Prinz, L. F. Abbott, and E. Marder. The dynamic clamp comes of age. *Trends in Neurosciences*, 27(4):218–224, 2004.

[162] Pedro F. Proenca and Yang Gao. Deep learning for spacecraft pose estimation from photorealistic rendering. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 6007–6013, 2020.

[163] Pedro F. Proença and Yang Gao. Deep learning for spacecraft pose estimation from photorealistic rendering. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6007–6013, 2020.

[164] W. Qiu et al. Unrealcv: Connecting computer vision to unreal engine. In *ACM Multimedia*, 2017.

[165] Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3828–3836, 2017.

[166] Mohammadhussein Rafieisakhaei, Suman Chakravorty, and P. R. Kumar. On the use of the observability gramian for partially observed robotic path planning problems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 1523–1528, 2017.

[167] Niyousha Rahimi. *Machine Learning in Feedback Systems: Provable Methods for Safe and Robust Autonomy*. PhD thesis, 2024. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2024-10-08.

[168] Niyousha Rahimi, Shahriar Talebi, Aditya Deole, Mehran Mesbahi, Saptarshi Bandyopadhyay, and Amir Rahmani. Robust controller synthesis for vision-based spacecraft guidance and control. In *AIAA SCITECH 2022 Forum*. American Institute of Aeronautics and Astronautics, January 2022.

[169] C. Radhakrishna Rao. *Breakthroughs in Statistics: Foundations and Basic Theory*, chapter Information and the Accuracy Attainable in the Estimation of Statistical Parameters, pages 235–247. Springer New York, New York, NY, 1992.

[170] K. Reif, S. Gunther, E. Yaz, and R. Unbehauen. Stochastic stability of the discrete-time extended kalman filter. *IEEE Transactions on Automatic Control*, 44(4):714–728, 1999.

[171] Chi Ren and Takaki Komiyama. Wide-field calcium imaging of cortex-wide activity in awake, head-fixed mice. *STAR Protocols*, 2(4):100973, December 2021.

[172] Taylor P. Reynolds, Danylo Malyuta, Mehran Mesbahi, Behçet Açıkmeşe, and John M. Carson. Funnel synthesis for the 6-dof powered descent guidance problem. *AIAA Scitech 2021 Forum*, pages 1–21, 2021.

[173] Taylor Patrick Reynolds. *Computational Guidance and Control for Aerospace Systems*. PhD thesis, University of Washington, 2020.

[174] A. W. Roberts and D. E. Varberg. Another proof that convex functions are locally lipschitz. *The American Mathematical Monthly*, 81(9):1014–1016, November 1974.

[175] G. Ros et al. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[176] Joel A Rosenfeld and Rushikesh Kamalapurkar. Dynamic mode decomposition with control liouville operators. *arXiv preprint arXiv:2101.02620*, 2021.

[177] E. Feron S. Boyd, L. El Ghaoui and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 3. Studies in Applied Mathematics, SIAM, Philadelphia, PA, 1994.

[178] Kevin Scaman and Aladin Virmaux. Lipschitz regularity of deep neural networks: analysis and efficient estimation, 2019.

[179] D. Scheinost, C. Lacadie, M. Hampson, R. T. Constable, and L. R. Ment. Alterations in spontaneous cortical activity networks and their modulation by neurofeedback. *NeuroImage*, 146:199–207, 2017.

[180] Stefan Scherer et al. A verification framework for runtime assurance of autonomous uas. Technical report, NASA, 2024.

[181] J. Schiemann, P. Puggioni, J. E. Dacre, M. Pelko, A. P. Domanski, M. C. W. van Rossum, and I. Duguid. Adaptive closed-loop control of neural circuits using reinforcement learning. *Nature Communications*, 8:2194, 2017.

[182] S. Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *International Symposium on Field and Service Robotics*, 2017.

[183] Sumant Sharma, Connor Beierle, and Simone D'Amico. Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks. In *2018 IEEE Aerospace Conference*, pages 1–12, 2018.

[184] Sumant Sharma, Connor Beierle, and Simone D'Amico. Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks. In *2018 IEEE Aerospace Conference*, pages 1–12, 2018.

[185] Enrico Silani and Marco Lovera. Magnetic spacecraft attitude control: a survey and some new results. *Control Engineering Practice*, 13(3):357–371, 2005. Aerospace IFAC 2002.

[186] Craig S. Sims. *A class of reduced order filters designed for state dependent noise*, page 855–863. Springer-Verlag.

[187] Abhay Singh and Juergen Hahn. Determining Optimal Sensor Locations for State and Parameter Estimation for Stable Nonlinear Systems. *Industrial and Engineering Chemistry Research*, 44, June 2005.

[188] Shubham D. Sonawani, Ryan Alimo, Renaud Detry, Daniel Jeong, Andrew Hess, and Heni Ben Amor. Assistive relative pose estimation for on-orbit assembly using convolutional neural networks. *CoRR*, abs/2001.10673, 2020.

[189] Shubham D. Sonawani, Ryan Alimo, Renaud Detry, Daniel Jeong, Andrew Hess, and Heni Ben Amor. Assistive relative pose estimation for on-orbit assembly using convolutional neural networks. *CoRR*, abs/2001.10673, 2020.

[190] Liang Song, Zhi Li, and Xingrui Ma. Autonomous rendezvous and docking of an unknown tumbling space target with a monocular camera. In *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, pages 1008–1013, 2014.

[191] Shichao Song, Andy Zeng, Johnny Lee, and Thomas Funkhouser. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 431–440, 2020.

[192] Gene W Sparrow and Douglas B. Price. *Derivation of Approximate Equations for Solving the Planar Rendezvous Problem.* National Aeronautics and Space Administration, District of Columbia, 1968.

[193] A. Srinivasan, S. Sahoo, K. E. Decker, and C. E. Vargas-Irwin. Functional optogenetic stimulation of peripheral nerves for restoring movement. *Frontiers in Neuroscience*, 12:72, 2018.

[194] Enrico Stoll, Jürgen Letschnik, Ulrich Walter, Jordi Artigas, Phillip Kremer, Carsten Preusche, and Gerd Hirzinger. On-orbit servicing. *IEEE Robotics & Automation Magazine*, 16(4):29–33, December 2009.

[195] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Matteo Carandini, and Kenneth D. Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765):361–365, June 2019.

[196] Martin Sundermeyer et al. Zs6d: Zero-shot 6d object pose estimation using local feature matching with vision transformers. *arXiv preprint arXiv:2309.11986*, 2023.

[197] Amirhossein Taghvaei and Prashant G. Mehta. An optimal transport formulation of the ensemble kalman filter. *IEEE Transactions on Automatic Control*, 66(7):3052–3067, 2021.

[198] Shahriar Talebi, Siavash Alemzadeh, Niyousha Rahimi, and Mehran Mesbahi. On regularizability and its application to online control of unstable LTI systems. *IEEE Transactions on Automatic Control*, pages 1–1, 2021.

[199] Wentao Tang, Zhenhua Wang, Qinghua Zhang, and Yi Shen. Set-membership estimation for linear time-varying descriptor systems. *Automatica*, 115:108867, 2020.

[200] Yukai Tang, Jean-Bernard Lasserre, and Heng Yang. Uncertainty quantification of set-membership estimation in control and perception: Revisiting the minimum enclosing ellipsoid. *Proceedings of Machine Learning Research*, 2024.

[201] Unity Technologies. Unity perception package. `https://github.com/Unity-Technologies/com.unity.perception`. Accessed May 26, 2025.

[202] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[203] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 292–301, 2018.

[204] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[205] George Terzakis and Manolis Lourakis. A Consistently Fast and Globally Optimal Solution to the Perspective-n-Point Problem. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12346 LNCS, 2020.

[206] Laurent Thiry and Bernard Thirion. Functional (meta)models for the development of control software. *IFAC Proceedings Volumes*, 41(2):8449–8454, 2008.

[207] P. Tsiotras. Further passivity results for the attitude control problem. *IEEE Transactions on Automatic Control*, 43(11):1597–1600, 1998.

[208] Panagiotis Tsiotras. Stabilization and optimality results for the attitude control problem. *Journal of Guidance, Control, and Dynamics*, 19(4):772–779, 1996.

[209] Panagiotis Tsiotras. Stabilization and optimality results for the attitude control problem. *Journal of Guidance, Control, and Dynamics*, 19(4):772–779, 1996.

[210] Panagiotis Tsiotras. Optimal regulation and passivity results for axisymmetric rigid bodies using two controls. *Journal of Guidance, Control, and Dynamics*, 20(3):457–463, 1997.

[211] Peter Van Overschee and Bart De Moor. Subspace algorithms for the stochastic identification problem. *Automatica*, 29(3):649–660, 1993.

[212] Peter Van Overschee and Bart De Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, January 1994.

[213] Peter Van Overschee and Bart De Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994. Special issue on statistical signal processing and control.

[214] Henk J. van Waarde. Beyond persistent excitation: Online experiment design for data-driven modeling and control, 2021.

[215] Henk J. van Waarde, M. Kanat Camlibel, and Mehran Mesbahi. From noisy data to feedback controllers: Nonconservative design via a matrix s-lemma. *IEEE Transactions on Automatic Control*, 67(1):162–175, 2022.

[216] Henk J. van Waarde, Jaap Eising, Harry L. Trentelman, and M. Kanat Camlibel. Data informativity: A new perspective on data-driven analysis and control. *IEEE Transactions on Automatic Control*, 65(11):4753–4768, 2020.

[217] Vijay Veerabadran, Josh Goldman, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alexey Kurakin, Ian Goodfellow, Jonathon Shlens, Jascha Sohl-Dickstein, Michael C. Mozer, and Gamaleldin F. Elsayed. Subtle adversarial image manipulations influence both human and machine perception. *Nature Communications*, 14(1), August 2023.

[218] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.

[219] Can Wang, Yifei Liu, Zhigang Yu, Yisheng Huang, and Yanning Yang. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16611–16621, 2021.

[220] Haoran Wang et al. Corr2distrib: Correspondence to distribution for 6d object pose estimation. *arXiv preprint arXiv:2505.02489*, 2025.

[221] He Wang, Shaofei Zhang, Xiangyu Ji, Jiajun Sun, and Hujun Bao. Self6d: Self-supervised monocular 6d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 108–125, 2020.

[222] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2019.

[223] Yichen Wang, Xiang Liu, Yu Li, Yu Xiang, and Dieter Fox. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3343–3352, 2019.

[224] Zhiguo Wang, Xiaojing Shen, and Yunmin Zhu. Posterior cramér-rao bounds for non-linear dynamic system with colored noises. *Journal of Systems Science and Complexity*, 32(6):1526–1543, December 2019.

[225] Andrew D. Wilson, Jarvis A. Schultz, and Todd D. Murphey. Trajectory synthesis for fisher information maximization. *IEEE Transactions on Robotics*, 30(6):1358–1370, December 2014.

[226] A. Witt, A. Van Ooyen, and C. Boucsein. Computational modeling of closed-loop optogenetic stimulation. *Frontiers in Computational Neuroscience*, 7:84, 2013.

[227] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*, 2018.

[228] Liang Xu, Ruixin Niu, and Erik P. Blasch. Uncertainty aware ekf: A tracking filter learning lidar measurement uncertainty. In *2022 25th International Conference on Information Fusion (FUSION)*, pages 1–8, 2022.

[229] Y. Xu et al. Certified adversarial robustness via partition-based randomized smoothing. *arXiv:2409.13546*, 2024.

[230] Yan Xu, Kwan-Yee Lin, Guofeng Zhang, Xiaogang Wang, and Hongsheng Li. RNNPose: Recurrent 6-DoF Object Pose Refinement With Robust Correspondence Field Estimation and Pose Optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14880–14890, June 2022.

[231] Keigo Yamada, Yasuo Sasaki, Takayuki Nagata, Kumi Nakai, Daisuke Tsubakino, and Taku Nonomura. Efficient sensor node selection for observability gramian optimization. *Sensors*, 23(13), 2023.

[232] Shuo Yang, George J. Pappas, Rahul Mangharam, and Lars Lindemann. Safe perception-based control under stochastic sensor uncertainty using conformal prediction. *arXiv:2304.00194*, 2023.

[233] W. Yang and R. Yuste. All-optical interrogation of neural circuits. *Current Opinion in Neurobiology*, 50:1–10, 2020.

[234] Yuxiao Yang, Allison T Connolly, and Maryam M Shanechi. A control-theoretic system identification framework and a real-time closed-loop clinical simulation testbed for electrical brain stimulation. *Journal of Neural Engineering*, 15(6):066007, Sep 2018.

[235] Xinyi Yu et al. 6dope-gs: Gaussian splatting for efficient 6d object pose estimation. *arXiv preprint arXiv:2403.10683*, 2024.

[236] Yue Yu, Shahriar Talebi, Henk J. van Waarde, Ufuk Topcu, Mehran Mesbahi, and Behçet Açıkmeşe. On controllability and persistency of excitation in data-driven control: Extensions of willems' fundamental lemma. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6485–6490, 2021.

[237] Zhigang Yu, Yisheng Huang, Yue Zhang, Minglun Zhang, and Yanning Yang. Pixel-level voting network for 6dof pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 831–846, 2020.

[238] Ming Zhou et al. Fast gdrnpp: Accelerating 6d pose estimation with pruning and knowledge distillation. *arXiv preprint arXiv:2409.12720*, 2024.

[239] Daphney-Stavroula Zois, Marco Levorato, and Urbashi Mitra. Active classification for pomdps: A kalman-like state estimator. *IEEE Transactions on Signal Processing*, 62(23):6209–6224, 2014.

[240] Daphney-Stavroula Zois and Urbashi Mitra. Active state tracking with sensing costs: Analysis of two-states and methods for n-states. *IEEE Transactions on Signal Processing*, 65(11):2828–2843, 2017.