

Algorithm Design for Safe and Efficient Societal-Scale Navigation

Chih-Yuan Chiu



Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2023-267

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-267.html>

December 11, 2023

Copyright © 2023, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Algorithm Design for Safe and Efficient Societal-Scale Navigation

by

Chih-Yuan Chiu

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering—Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor S. Shankar Sastry, Chair

Professor Claire Tomlin

Professor Murat Arcaç

Professor Koushil Sreenath

Winter 2023

Algorithm Design for Safe and Efficient Societal-Scale Navigation

Copyright 2023
by
Chih-Yuan Chiu

Abstract

Algorithm Design for Safe and Efficient Societal-Scale Navigation

by

Chih-Yuan Chiu

Doctor of Philosophy in Engineering—Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor S. Shankar Sastry, Chair

In modern urban centers, traffic networks increasingly experience higher densities of both human-operated and self-driving vehicles. Unfortunately, high traffic loads can increase the likelihood of accidents and gridlock at individual intersections and roads, and produce societal-scale externalities in the form of pollution and excessive commute times. To address these issues, modern navigation and transportation technologies increasingly leverage machine learning-enabled components to safely and efficiently guide commuters towards their desired destinations. Examples range from learning-enabled perception and motion planning algorithms deployed on self-driving vehicles, to recommendation systems for route planning in large-scale transportation networks. Unfortunately, learning-based algorithms can exhibit unexpected and alarming behavior when deployed in real-world environments. For instance, computer vision modules in self-driving vehicles frequently fail to correctly identify traffic signs and predict pedestrian motion, with fatal consequences. Meanwhile, route recommendation platforms can induce high congestion levels, by directing self-interested travelers to overcrowd paths of least perceived latency. These phenomena highlight that, despite their promise, modern ML algorithms for localized and societal-scale navigation remain unable to operate robustly in real-world traffic scenarios.

To overcome these challenges, this thesis draws from tools in control theory, estimation theory, numerical optimization, game theory, and mechanism design to design algorithms that ensure safe and efficient navigation in modern transportation systems. In particular, the thesis consists of the following parts, each of which targets a different facet of decision making in traffic navigation: (I) A unified optimization-based state estimation algorithm for autonomous agents; (II) Game-theoretic motion planners that characterize multi-agent interactions in local traffic scenarios; (III) A dynamic tolling scheme and learning updates for large traffic networks. We conclude by describing promising avenues of future work.

Contents

Contents	i
1 Introduction	1
I SLAM: A Generalized Optimization Framework	5
2 Static SLAM	6
2.1 Introduction	6
2.2 SLAM: Formulation on Euclidean Spaces	8
2.3 SLAM: Formulation on Manifolds	9
2.4 Main Algorithm	11
2.5 Gauss-Newton Descent	15
2.6 Marginalization of States	16
2.7 Main Algorithm on Manifolds	18
3 Equivalence of Filtering and Optimization	22
3.1 Extended Kalman Filter (EKF) on Euclidean Spaces, Standard Formulation	22
3.2 Extended Kalman Filter (EKF) on Euclidean Spaces, in an Optimization Framework	24
3.3 Multi-State Constrained Kalman Filter (MSCKF), Standard Formulation . .	31
3.4 Multi-State Constrained Kalman Filter (MSCKF), on Manifolds	36
3.5 State-of-the-Art SLAM Algorithms	45
3.6 Experiments	46
3.7 Discussion	49
II Game-Theoretic Motion Planning for Autonomous Vehicles	50
4 Defensive Driving	51
4.1 Related Work	54
4.2 Preliminaries	56
4.3 Methods	58

4.4	Implementation Details: ILQGames	60
4.5	Results	60
4.6	Discussion	65
5	Game-Theoretic Priors for SLAM	66
5.1	Related Work	67
5.2	Setup and Notation	68
5.3	Methods	70
5.4	Experiment Results	74
5.5	Discussion	77
III Adaptive Tolling for Transportation Networks		82
6	Adaptive Tolling for Arc-Based Traffic Assignment	83
6.1	Preliminaries	85
6.2	Optimal Toll: Existence and Uniqueness	94
6.3	Dynamics and Convergence	100
6.4	Experiment Results	113
6.5	Discussion	113
7	Online Learning for Adaptive Tolling	117
7.1	Preliminaries	118
7.2	Main Algorithm	121
7.3	Lemmas for Regret Analysis	130
7.4	Regret Analysis	133
7.5	Experiments	143
7.6	Discussion	144
IV Future Work		145
8	Conclusion	146
8.1	Part 1	146
8.2	Part 2	147
8.3	Part 3	149
Bibliography		151

Acknowledgments

I would like to thank my advisor, Professor Shankar Sastry, for his constant advice and enthusiasm for research throughout my Ph.D. journey. It is safe to say that I would not be where I am today without his insightful professional feedback. I would also like to thank Professors Claire Tomlin, Gireeja Ranade, David Fridovich-Keil, Forrest Laine, Yi Ma, Eric Mazumdar, and Lillian Ratliff, for their support and encouragement throughout my Ph.D. experience. I would also like to thank my collaborators (for both past and ongoing work, listed in alphabetical order of surname) Dr. Daniel Calderone, Benjamin Chasnov, Sampada Deglurkar, Jingqi Li, Chinmay Maheshwari, Anish Muthali, Druv Pai, Lasse Peters, Amay Saxena, Haotian (David) Shen, Ritika Shrivastava, Pan-Yang Su, Victoria Tuck, and Dr. Manxi Wu for excellent collaborations I have enjoyed at UC Berkeley, some of which are described in this thesis. Special thanks to Dr. Tyler Westenbroek and Professor David Fridovich-Keil, for providing perspectives and conversations that were singularly helpful during the final year of my PhD experience. Finally, I would like to thank my family and friends for their support during my graduate school years.

Chapter 1

Introduction

In our modern era of artificial intelligence, navigation and transportation technologies increasingly leverage learning-enabled components to perform rapid data collection and processing, to facilitate autonomous decision making in multi-agent interactions. These interactions can occur in *atomic* settings involving a small number of agents in a local environment, such as autonomous navigation, or in *non-atomic* settings in which the contribution of each agent’s behavior is infinitesimal, such as route-planning in large-scale transportation networks. Unfortunately, learning-based methods can exhibit unexpected and alarming behavior once deployed in real-world environments. For instance, learning-enabled computer vision modules in self-driving vehicles frequently fail to correctly identify traffic signs, with fatal consequences. Meanwhile, learning-based route recommendation platforms often induce high congestion levels, by directing self-interested travelers to overcrowd paths of least perceived latency. These phenomena highlight that, despite their promise, modern machine learning (ML) methods for localized and societal-scale navigation continue to be plagued by the following fundamental flaws:

1. **Lack of Robustness and Interpretability**—Current state-of-the-art ML-based navigation algorithms often struggle to robustly process noisy observations of an autonomous agent’s surroundings, a crucial prerequisite for local motion planning. As such, these navigation algorithms are brittle, incapable of sustained and trustworthy operation when deployed for localized navigation in challenging real-world environments. This leads to unpredictable and undesirable consequences, such as autonomous driving-related fatalities.
2. **Socially Undesirable Outcomes**—Current ML-based route recommendation platforms are not endowed with concepts relevant to collective social optimality or strategic human decision-making. As such, when guiding travelers across large-scale transportation networks, they often fail to account for interactions between the profit-maximizing, yet individually unpredictable, agents using these networks. This results in the aforementioned excessive congestion.

To address the above issues, this dissertation presents principled mathematical methods for designing algorithms for assured autonomous navigation on localized and societal scales.

Part I: A Unified Framework for Simultaneous Localization and Mapping (SLAM)

An autonomous vehicle navigating a previously unseen traffic scenario must accurately represent its environment and locate itself within that representation. While purely learning-based perception is increasingly deployed in autonomy stacks, such methods can exhibit unexpected and alarming behavior in real-world traffic. For instance, computer vision modules in self-driving vehicles often fail to correctly identify traffic signs and predict pedestrian motion, with fatal consequences.

To address this issue, my work presents a unified optimization-based framework for Simultaneous Localization and Mapping (SLAM) algorithms, which construct geometric maps describing an autonomous agent’s surroundings, while locating the agent within the established map. In the existing literature, SLAM methods are categorized into filtering-based algorithms, which are usually computationally faster, and optimization-based methods, which tend to be more accurate. My colleagues, **Amay Saxena, Ritika Shrivastava, Dr. Joseph Menke, Professor Shankar Sastry** and I reconciled the strengths of these two methods by formulating a unified optimization-based framework that encompasses a large class of existing, state-of-the-art SLAM algorithms for static scenes [95]. Our framework allows the robustness and accuracy of these methods to be interpreted as the consequence of algorithmic design choices, and easily contrasted across SLAM benchmark datasets. Moreover, our framework facilitates the design of new algorithms whose performance flexibly interpolates those of current methods. Altogether, our work provides a unified framework for perception and estimation, to inform immediate decision-making in autonomous navigation.

Part II: Game-Theoretic Motion Planning

When designing a safe trajectory for a given (“ego”) autonomous vehicle, the motion planning literature typically models the other (“non-ego”) vehicles in the scene as dynamic obstacles with fixed trajectories that were independently predicted upstream in the autonomy stack. Unfortunately, this assumption precludes the ego agent’s ability to account for non-ego agents’ reactions when executing their own planned trajectories.

To address this issue, my colleagues **Professor Claire Tomlin, David Fridovich-Keil**, and I focused on developing game-theoretic motion planners, which explicitly encode the intent of other non-ego agents, as well as their potential reactions to the ego agent’s trajectory, as coupled optimization problems. These algorithms are capable of capturing complex multi-agent interactions that frequently arise in real-life traffic scenarios. These

include defensive driving, in which the ego agent guards themselves against the possibility that non-ego agents in close proximity are temporarily distracted, and may unintentionally exhibit adversarial behavior.

In the literature, the objectives of self-interested agents interacting in a shared environment are often encoded as a set of coupled Hamilton-Jacobi equations whose solutions yield local Nash equilibrium strategies [100, 99]. These strategies can then be computed numerically via state space discretization. However, these algorithms inevitably require computational cost and memory that scale exponentially with the state dimension (the “curse of dimensionality”), and are thus unsuitable for modeling the high-dimensional, multi-player interactions in real-world traffic [9]. To overcome these computational challenges, our work uses ILQGames [46], a recently developed iterative linear-quadratic algorithm, as our primary game solver. ILQGames iteratively solves linear-quadratic games and incurs computational complexity cubic in the number of players and linear in the time horizon.

We conclude this part by presenting GTP-SLAM, a game-theoretic framework for joint localization, mapping, prediction, and planning. The GTP-SLAM algorithm offers a rapprochement between the game-theoretic path planning paradigm explored in Part II with the SLAM algorithms described in Part I. More precisely, we formulate a novel SLAM algorithm for multi-player scenes, motivated by iterative best response. GTP-SLAM aims to jointly estimate the dynamic states and control inputs of all players in the scene, as well as landmark positions. It does so from the ego player’s perspective, while accounting for noncooperative, game-theoretic interactions between the players.

Part III: Congestion Management via Dynamic Tolling

Dynamic tolling schemes provide a practical method for reducing traffic congestion in transportation networks, in which self-interested travelers select routes that minimize travel times to their destinations. Traffic assignment models (TAMs) play a critical role in congestion modeling, by predicting travelers’ arc selection decisions based on the traffic load and imposed toll on each arc. TAMs operate on the core principle that self-interested travelers select routes with minimal perceived cost, which can be modeled as deterministic or stochastic.

Part III of this thesis presents joint work with **Chinmay Maheshwari, Pan-Yang Su, and Professor Shankar Sastry**, in which we developed timescale-separated stochastic dynamics models for the traffic flow and imposed tolls on a traffic network of arbitrary structure and scale. Many traffic assignment models (TAMs) in the literature are route-based, which assumes that travelers make a single decision among routes that connect their origin and destination. However, the comprehensive enumeration of routes in a traffic network is often computationally intractable, since the number of available routes can be exponential in the number of arcs. To tackle this issue, Chapter 4 presents arc-based TAMs that model traffic network congestion generated by self-interested travelers who sequentially select arcs based on their perceived latency on the network. Moreover, we use perturbed best-response dynamics to characterize the selfish, yet not entirely predictable, routing and rerouting decisions

of travelers in the transportation system, as they learn and respond to the fluctuating load-dependent latency and toll on each arc in the network. We prove that arc flows generated by these dynamics converge to a neighborhood of the associated equilibrium flow allocation. At the same time, we implement a dynamic tolling scheme to ensure convergence to a socially desirable traffic load distribution.

However, for the tolling mechanism to be effectively implemented, the central traffic authority must possess knowledge of a wide range of system parameters. These include parameters for arc latency functions that relate travel time to traffic load, as well as dispersion (temperature) and noise parameters that capture degrees of stochasticity and rationality in the travelers' aggregate decision-making. All of these parameters may a priori be unknown to the traffic authority, and difficult to estimate. To address this issue, Chapter 5 presents an online learning algorithm that estimates unknown system parameters while implementing effective tolls, thus generalizing the dynamic tolling scheme introduced in Chapter 4.

Overall, this dissertation thesis presents the first steps towards a unified theory of socially optimal autonomous navigation that blends both learning- and model-based methods for efficient data-processing and decision-making. We conclude by discussing promising directions of future research.

Acknowledgements Needless to say, my research has benefited tremendously from the fruitful collaborations in which I was privileged to partake during my PhD experience. These collaborations feature prominently in the content of this thesis. In particular, Part I draws heavily from collaborations with Amay Saxena, Ritika Shrivastava, Dr. Joseph Menke, and Professor Shankar Sastry [95, 17, 16]. Part II is based on papers published in collaboration with Professors Claire Tomlin and David Fridovich-Keil [21, 18]. Finally, Part III is derived from a productive collaboration with Chinmay Maheshwari, Pan-Yang Su, and Professor Sastry over the course of the past year and a half [22, 20].

Part I

SLAM: A Generalized Optimization Framework

Chapter 2

Static SLAM

2.1 Introduction

A critical function for any autonomous vehicle navigating a previously unseen traffic scenario is to accurately represent its environment and locate itself within that representation. While purely learning-based perception is increasingly used to model an autonomous agent's environment, such methods can exhibit unexpected and alarming behavior when deployed in real-world environments. For instance, even when equipped with state-of-the-art computer vision algorithms, self-driving vehicles often fail to correctly identify traffic signs and predict pedestrian motion, sometimes with fatal consequences.

To address these issues, Chapters 2 and 3 focus on Simultaneous Localization and Mapping (SLAM), a well-studied problem in robotics, computer vision, and estimation theory. In SLAM, a robot builds a representation for a previously unexplored environment while locating itself in the constructed map [66]. Applications of SLAM range widely, from military applications such as map construction and search-and-rescue missions, to civilian applications such as augmented and virtual reality, and 3D scene capture [14, 24, 25, 26].

The first step towards solving the SLAM problem is to locate and identify landmarks in the environment, and extract features to construct a map. The robotic agent then uses its dynamics model, in conjunction with these feature measurements, to pinpoint the position and orientation (pose) of the robotic agent relative to the map. Measurements of new features from newly identified landmarks, and new measurements of old features, can be used to iteratively update the SLAM states—that is, landmark positions and estimates of the robot's pose. This reduces errors in original estimates of landmark positions and the robot pose, due to either measurement noise or fluctuations in the environment.

Modern SLAM algorithms usually include *front* and *back ends*. The front end performs feature extraction, data association, and outlier rejection on raw sensor data. This includes feature extraction, data association, and outlier rejection, to match features across feature data, filter out spurious feature matches (outliers), process IMU data, and associate all of this information with relevant SLAM states. The back end then performs inference over the

processed data, using underlying dynamics and measurement models to produce a compatible state estimate. Back end algorithms classified as *Gaussian filtering* or *batch optimization* based. Filtering methods iteratively update the distribution of recently observed states under a Gaussian distribution assumption [98, 79, 69], while optimization methods iteratively estimate states as solutions to an optimization problem, with the optimization objective constructed from sensor error, such as error terms collected from inertial measurement units (IMU) and image reprojection. In particular, factor graph-based approaches efficiently solve optimization problems over past variables via factorization schemes that maintain the sparsity of the underlying least squares problem [28, 57, 30, 56]. Keyframe-based methods are optimization-based approaches that retain only a small subset of maximally informative frames (“keyframes”) spaced arbitrarily far apart in time in the optimization window, while dropping all other poses [67]. Keyframes are selected to maximize the information captured in optimization windows of limited size. Empirically, algorithms from both categories have achieved state-of-the-art performance, though the latter often attain higher accuracy at the cost of longer compute times [14, 67, 34].

Prior literature contrasted theoretical and empirical properties of filtering and batch optimization algorithms. Scaramuzza and Fraundorfer performed a comparison study of filtering and bundle adjustment-based methods for visual odometry [96, 44]. Frese et al. provided a survey, from a practitioner’s perspective, of grid-based and pose graph SLAM algorithms [45]. By contrast, Huang and Dissanyake studied the consistency, accuracy, and computational speed of filtering, optimization-based, and pose-graph SLAM from a theoretical perspective [52]. Khosoussi et al. exploited sparsity in SLAM problems by conditioning on estimates of robot orientations [61]. Strasdat et al. conducted Monte Carlo experiments on visual SLAM algorithms [101], revealed that including more features in the back end increased accuracy more (compared to including more frames), and concluded that bundle adjustment (BA) outperforms filtering, since its computation time increases less drastically with the number of features.

In this chapter and the next, we present a unified template for filtering and optimization-based SLAM approaches, by building upon methods surveyed by the publications presented above. Our aim is to facilitate the design of new algorithms whose computational speed and performance flexibly interpolate those of existing, state-of-the-art algorithms. We then use this framework to elucidate the tradeoff between the accuracy and computational complexity of filtering-based and batch optimization-based algorithms in different environments. Specifically, we recast the Extended Kalman Filter (EKF), Multi-State Constrained Kalman Filter (MSCKF), and Open Keyframe Visual-Inertial SLAM algorithm (OKVIS) as optimization-based back-end algorithms, using our unified SLAM framework. We then compare the empirical performance of the reformulated MSCKF with that of sliding window optimization-based back-end algorithms, including the keyframe-based approach of Open Keyframe Visual-Inertial SLAM [67]. Somewhat surprisingly, the MSCKF outperforms sliding window filters (SWF) of comparable sizes on several datasets, despite not performing multiple Gauss-Newton updates. We use our unified framework to analyze these empirical results. More details can be found in a published paper, Saxena*, Chiu*, Shrivastava,

Menke, and Sastry “Simultaneous Localization and Mapping: Through the Lens of Nonlinear Optimization” [95] (*Equal contribution).

2.2 SLAM: Formulation on Euclidean Spaces

SLAM on Euclidean Spaces

The SLAM problem involves estimating two types of variables: *states* and *features*. The state at each time t , denoted $x_t \in \mathbb{R}^{d_x}$, describes physical attributes of the robot, e.g., camera positions and orientations (poses). Feature positions available at time t in a global frame, denoted $\{f_k | j = 1, \dots, p\} \subset \mathbb{R}^{d_f}$, can be obtained by analyzing information from image measurements $\{z_{t,k} | j = 1, \dots, p\} \subset \mathbb{R}^{d_z}$ and state estimates; these describe the relative position of the robot in its environment. States and features are described by an infinitely differentiable (i.e., C^∞) dynamics map $g : \mathbb{R}^{d_x} \rightarrow \mathbb{R}^{d_x}$ and a C^∞ measurement map $h : \mathbb{R}^{d_x} \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_z}$, via additive noise models:

$$x_{t+1} = g(x_t) + w_t, \quad w_t \sim \mathcal{N}(0, \Sigma_w), \quad (2.1)$$

$$z_{t,k} = h(x_t, f_k) + v_{t,k}, \quad v_{t,k} \sim \mathcal{N}(0, \Sigma_v), \quad (2.2)$$

where $\Sigma_w \in \mathbb{R}^{d_x \times d_x}$, $\Sigma_w \succeq 0$ and $\Sigma_v \in \mathbb{R}^{d_z \times d_z}$, $\Sigma_v \succeq 0$.

For localization and mapping, SLAM algorithms maintain a *full state (vector)* $\bar{x}_t \in \mathbb{R}^d$, in which a number of past states and feature positions are concatenated. The exact number and time stamps of these states and features vary with the design choice of each SLAM algorithm. For example, sliding window filters (SWFs) may define the full state $\bar{x}_t := (x_{t-n+1}, \dots, x_t, f_{p-q+1}, \dots, f_p) \in \mathbb{R}^d$, with $d := d_x n + d_f q$, to be a sliding window of the most recent n states, consisting of one pose each, and the most recent estimates, at time t , of a collection of q features [98, 79]. Batch optimization methods, on the other hand, maintain all states and features encountered in the problem up to the current time [30, 28, 57, 56].

Note that (2.1) and (2.2) do not involve overparameterized state variables, e.g., quaternion representations for poses. These natural extensions are discussed in Section 2.3.

SLAM as an Optimization Problem on Euclidean Spaces

SLAM estimates state and feature positions that best enforce constraints posed by given dynamics and measurement models, as well as noisy state and feature measurements collected over time. This is formulated as the minimization of the sum of weighted residual terms representing these constraints. For example, weighted residuals associated with the prior distribution over $\bar{x}_t \in \mathbb{R}^d$, the dynamics constraints between states $x_i, x_{i+1} \in \mathbb{R}^{d_x}$, and the reprojection error of feature $f_j \in \mathbb{R}^{d_f}$ corresponding to the state $x_i \in \mathbb{R}^{d_x}$ and image measurement $z_{i,j} \in \mathbb{R}^{d_z}$, may be given by $\Sigma_0^{-1/2}(\bar{x}_t - \mu_0) \in \mathbb{R}^d$, $\Sigma_w^{-1/2}(x_{i+1} - g(x_i)) \in \mathbb{R}^{d_x}$, and $\Sigma_v^{-1/2}(z_{i,j} - h(x_i, f_j)) \in \mathbb{R}^{d_z}$, respectively (here, $1 \leq i \leq n-1$, $1 \leq j \leq q$). We define

the running cost, $c : \mathbb{R}^{d_x n + d_f q} \rightarrow \mathbb{R}$, as the sum of weighted norm squares of these residuals. For example, for a SWF algorithm for SLAM:

$$c(\bar{x}_t) := \|\bar{x}_t - \mu_0\|_{\Sigma_0^{-1}}^2 + \sum_{i=t-n+1}^{t-1} \|x_{i+1} - g(x_i)\|_{\Sigma_w^{-1}}^2 + \sum_{j=p-q+1}^p \sum_{i=t-n+1}^t \|z_{i,j} - h(x_i, f_k)\|_{\Sigma_v^{-1}}^2, \quad (2.3)$$

where $\|v\|_A^2 := v^\top A v$ for any real vector v and real matrix A of compatible dimension.

To formulate SLAM as a nonlinear least-squares problem, we stack all residual terms into one residual vector $C(\bar{x}_t)$. For example, for the SWF given above:

$$\begin{aligned} C(\bar{x}_t) := & \left[(\Sigma_0^{-1/2}(\bar{x}_t - \mu_0))^\top \right. \\ & (\Sigma_w^{-1/2}(x_{t-n+1} - g(x_{t-n})))^\top \cdots (\Sigma_w^{-1/2}(x_t - g(x_{t-1})))^\top \\ & (\Sigma_v^{-1/2}(z_{t-n+1,p-q+1} - h(x_{t-n+1}, f_{p-q+1})))^\top \cdots \\ & (\Sigma_v^{-1/2}(z_{t-n+1,p} - h(x_{t-n+1}, f_p)))^\top \cdots \\ & (\Sigma_v^{-1/2}(z_{t,p-q+1} - h(x_t, f_{p-q+1})))^\top \cdots \\ & \left. (\Sigma_v^{-1/2}(z_{t,p} - h(x_t, f_p)))^\top \right]^\top \in \mathbb{R}^{(2n-1)d_x + nqd_z}. \end{aligned}$$

Thus, $c(\bar{x}_t) = C(\bar{x}_t)^\top C(\bar{x}_t)$, and the SLAM problem is now reduced to the nonlinear least squares problem below:

$$\min_{\bar{x}_t} .c(\bar{x}_t) = \min_{\bar{x}_t} .C(\bar{x}_t)^\top C(\bar{x}_t) \quad (2.4)$$

Section 2.4 introduces the main algorithmic submodules used to find an approximate solution to (2.4).

2.3 SLAM: Formulation on Manifolds

Here, we generalize the SLAM formulation in Section 2.2 to the case where dynamical states are defined on smooth manifolds rather than Euclidean spaces. SLAM often involves estimating the orientations of rigid bodies, which evolve on a smooth manifold embedded in an ambient space, e.g., rotation matrices expressed as unit quaternions. In such situations, we use *boxplus* (\boxplus) and *boxminus* (\boxminus) operators, defined below, to perform composition and difference operations in the iterative algorithm presented in Section 2.4, while enforcing constraints imposed by the manifold's geometric structure.

Suppose the full state x evolves on a smooth manifold \mathcal{M} , with $\dim(\mathcal{M}) = n$. For each $x \in \mathcal{M}$, let $\pi_x : U_x \rightarrow V_x$ be a diffeomorphic chart from an open neighborhood $U_x \subset \mathcal{M}$ of

$x \in \mathcal{M}$ to an open neighborhood $V_x \subset \mathbb{R}^n$ of $0 \in \mathbb{R}^n$. Without loss of generality, suppose $\pi_x(x) = 0$. The operators $\boxplus : U_x \times V_x \rightarrow U_x$ and $\boxminus : U_x \times U_x \rightarrow V_x$ are defined by:

$$x \boxplus \delta = \pi_x^{-1}(\delta) \quad (2.5)$$

$$y \boxminus x = \pi_x(y) \quad (2.6)$$

In essence, \boxplus adds a perturbation $\delta \in \mathbb{R}^n$, in local coordinates, to a state $x \in \mathcal{M}$, while \boxminus extracts the difference $\delta \in \mathbb{R}^n$, in local coordinates, between states $x, x' \in \mathcal{M}$ covered by the same chart. Below, “ δ ” often describes an error or increment to a nominal state on the manifold.

Manifold Examples

This subsection gives examples of the \boxplus , \boxminus and π operators for manifolds that occur widely in SLAM: the set of unit quaternions, \mathbb{H}_u , and the set of rotation matrices, $SO(3)$.

Each $q \in \mathbb{H}_u$ is expressed as $q = (q_u, \vec{q}_v)$ where $q_u \in \mathbb{R}$ and $\vec{q}_v \in \mathbb{R}^3$ denote the scalar and vector (imaginary) parts, respectively, with $\|q\| = \sqrt{q_u^2 + \|\vec{q}_v\|_2^2} = 1$ (JPL convention). Here, the coordinate map $\pi : \mathbb{H}_u \rightarrow \mathbb{R}^3$ is defined as the Log map on \mathbb{H}_u ; its inverse π^{-1} is the Exp map. Specifically, we write each $q \in \mathbb{H}_u$ as $q = (\cos(\frac{\theta}{2}), \sin(\frac{\theta}{2})\vec{\omega})$ for some $\theta \in [0, \pi]$, $\vec{\omega} \in \mathbb{R}^3$ with $\|\vec{\omega}\| = 1$, i.e., the quaternion q implements a rotation about the axis $\vec{\omega}$ by θ radians counterclockwise. Then, $\pi : \mathbb{H}_u \rightarrow \mathbb{R}^3$ and $\pi^{-1} : B_\pi(0) \rightarrow \mathbb{H}_u$ are defined by: ($B_\pi(0) := \{x \in \mathbb{R}^3 : \|x\|_2 < \pi\}$ denotes the image of π)

$$\begin{aligned} \pi(q) &= \text{Log}(q) = \theta\vec{\omega}, \\ \pi^{-1}(\theta\vec{\omega}) &= \text{Exp}(\theta\vec{\omega}) = (\cos(\theta/2), \sin(\theta/2)\vec{\omega}). \end{aligned}$$

The \boxplus and \boxminus maps are then implemented via the quaternion product $\star : \mathbb{H}_u \times \mathbb{H}_u \rightarrow \mathbb{H}_u$:

$$\begin{aligned} q_a \boxplus \vec{\omega} &= q_a \star \text{Exp}(\vec{\omega}) \\ q_a \boxminus q_b &= \text{Log}(q_b^{-1} \star q_a) \end{aligned}$$

For $SO(3)$, we define \boxplus and \boxminus similarly, i.e.,

$$\begin{aligned} R_a \boxplus \vec{\omega} &= R_a \text{Exp}(\vec{\omega}) \\ R_a \boxminus R_b &= \text{Log}(R_b^T R_a) \end{aligned}$$

Often, the full state in a SLAM problem exists in the Cartesian product of a finite collection of manifolds, since it contains poses and features on their own manifolds. For a product manifold $\mathcal{M}_1 \times \mathcal{M}_2$, with projection, increment, and difference maps already defined on \mathcal{M}_1 and \mathcal{M}_2 , we define \boxplus and \boxminus on $\mathcal{M}_1 \times \mathcal{M}_2$ by:

$$\begin{aligned} (g_1, g_2) \boxplus (\xi_1, \xi_2) &= (g_1 \boxplus \xi_1, g_2 \boxplus \xi_2) \\ (g_1, g_2) \boxminus (h_1, h_2) &= (g_1 \boxminus h_1, g_2 \boxminus h_2) \end{aligned}$$

SLAM as an Optimization Problem on Manifolds

The SLAM problem can be formulated on manifolds using modified cost functions, where plus and minus operations are replaced with \boxplus and \boxminus when necessary. We provide a more explicit description of this process below.

To formulate SLAM on a manifold, we must alter our definitions of the state variables, features, image positions, dynamics map, and measurement map. Let \mathcal{X} be a smooth manifold of dimension d_x , on which the system state are defined. Similarly, let \mathcal{F} be a smooth manifold of dimension d_f , on which features are defined, and let \mathcal{Z} be the smooth manifold of dimension d_z , on which image measurements are defined (Often, $\mathcal{F} = \mathbb{R}^{d_f}$ and $\mathcal{Z} \in \mathbb{R}^{d_z}$, e.g., with $d_f = 3$ and $d_z = 2$). We then have:

$$\begin{aligned} x_{t+1} &= g(x_t) \boxplus w_t, & w_t &\sim \mathcal{N}(0, \Sigma_w), \\ z_{t,k} &= h(x_t, f_k) \boxplus v_{t,k}, & v_{t,k} &\sim \mathcal{N}(0, \Sigma_v). \end{aligned}$$

where $x_t \in \mathcal{X}$ denotes the state at time t , $g : \mathcal{X} \rightarrow \mathcal{X}$ denotes the discrete-time dynamics map, and $w_t \in \mathbb{R}^{d_x}$ denotes the dynamics noise, with covariance $\Sigma_w \in \mathbb{R}^{d_x \times d_x}$, $\Sigma_w \succ 0$. Moreover, $f_k \in \mathcal{F}$ denotes feature position j estimated at the camera pose at time t , $z_{t,k} \in \mathcal{Z}$ denotes the image measurement of feature j measured from the camera pose at time t , $h : \mathcal{X} \times \mathcal{F} \rightarrow \mathcal{Z}$ denotes the measurement map, and $v_t \in \mathbb{R}^{d_z}$ denotes the measurement noise, with covariance $\Sigma_v \in \mathbb{R}^{d_z \times d_z}$, $\Sigma_v \succ 0$.

As before, SLAM concerns an optimization problem over a collection of poses and features, e.g., a sliding window of the most recent poses in the states $\{x_i \in \mathcal{X} | i = t-n+1, \dots, t\}$ and features $\{f_k \in \mathcal{F} | j = p-q+1, \dots, p\}$:

$$\bar{x}_t := (x_{t-n+1}, \dots, x_t, f_{p-q+1}, \dots, f_p) \in \mathcal{X}^n \times \mathcal{F}^q.$$

We assume that \bar{x}_t is associated with a prior distribution with mean $\mu_0 \in \mathcal{X}^n \times \mathcal{F}^q$ and covariance $\Sigma_0 \in \mathbb{R}^{(nd_x+qd_f) \times (nd_x+qd_f)}$.

2.4 Main Algorithm

Algorithm Overview

Below, we describe in detail the submodules of a straightforward, general SLAM algorithm, using the state variables and cost terms defined in Sections 2.2 and 2.3. As before, we first focus on the case where the state space is Euclidean. As before, denote the state and concatenated cost vector by $\bar{x}_t \in \mathbb{R}^d$ and $C : \mathbb{R}^d \rightarrow \mathbb{R}^{d_C}$, respectively. (e.g., the sliding window filter in Section 2.2 would correspond to $d = d_x n + d_f q$ and $d_C = (2n-1)d_x + qd_f + nqd_z$). Recall from Chapter 2.2 that the SLAM problem is equivalent to solving the nonlinear least-squares problem (2.4), reproduced below:

$$\min_{\bar{x}_t} .c(\bar{x}_t) = \min_{\bar{x}_t} .\|C(\bar{x}_t)\|_2^2.$$

Constructing the Optimization Problem

The first step of the algorithm is to construct the objective function whose minimization defines the SLAM problem. From available visual and inertial measurements, we construct and concatenate a collection of residual terms to form a residual vector $C(\bar{x}_t)$ of the form introduced in Section 2.2. The SLAM problem is then equivalent to solving the nonlinear least-squares problem (2.4), reproduced below with the indices considered here:

$$\min_{\bar{x}_t} .c(\bar{x}_t) = \min_{\bar{x}_t} .C(\bar{x}_t)^\top C(\bar{x}_t).$$

Recall that $\bar{x}_t \in \mathbb{R}^{d_x n + d_f q}$ consists of the most recent n poses and position estimates of the q most salient features, measured at the most recent pose (at time t). Recall also that $C : \mathbb{R}^{d_x n + d_f q} \rightarrow \mathbb{R}^{d_x n + d_f q + d_x(n-1) + d_z n q}$ is defined by:

$$C(\bar{x}_t) := \begin{bmatrix} C_0(\bar{x}_t)^\top & C_{g_{t-n+1}}^\top & \cdots & C_{g_{t-n+1}}^\top & C_{h_{p-q+1, t-n+1}}^\top & \cdots & C_{h_{t, p-q+1}}^\top & \cdots \\ & C_{h_{t-n+1, p}}^\top & \cdots & C_{h_{t, p}}^\top & & & & \end{bmatrix}^\top.$$

with weighted residual terms $C_0 : \mathbb{R}^{d_x n + d_f q} \rightarrow \mathbb{R}^{d_x n + d_f q}$, and $C_{g_i} : \mathbb{R}^{2d_x} \rightarrow \mathbb{R}^{d_x}$, $C_{h_{ij}} : \mathbb{R}^{d_x + d_f} \rightarrow \mathbb{R}^{d_z}$ for each $i \in \{t-n+1, \dots, t-1\}$ and $j \in \{p-q+1, \dots, p\}$ given by:

$$C_0(\bar{x}_t) := \Sigma_0^{-1/2}(\bar{x}_t - \mu_0), \quad (2.7)$$

$$C_{g_i}(x_i, x_{i+1}) := \Sigma_w^{-1/2}(x_{i+1} - g(x_i)), \quad (2.8)$$

$$C_{h_{ij}}(x_i, f_{t,j}) := \Sigma_v^{-1/2}(z_{i,j} - h(x_i, f_{t,j})). \quad (2.9)$$

Gauss-Newton Descent and Linear Approximation

Next, we recursively update of the full state $\bar{x}_t \in \mathbb{R}^{d_x n + d_f q}$ by performing Gauss-Newton descent steps using the cost function. More precisely, let $J \in \mathbb{R}^{(d_x n + d_f q + d_x(n-1) + d_z n q) \times (d_x n + d_f q)}$ denote the Jacobian of C with respect to \bar{x}_t . Starting from an initial estimate $\bar{x}_t^{(0)} \in \mathbb{R}^{d_x n + d_f q}$ of \bar{x}_t , we recursively update iterates of our estimate $\{\bar{x}_t^{(k)} | k \geq 0\}$ via the Gauss-Newton algorithm:

$$\bar{x}_t^{(k+1)} \leftarrow \bar{x}_t^{(k)} - (J^\top J)^{-1} J^\top C(\bar{x}_t^{(k)}). \quad (2.10)$$

These Gauss-Newton steps are iteratively applied until the current iterate $\bar{x}_t^* := \bar{x}_t^{(k)}$, for some sufficiently large $k \in \mathbb{N}$, is believed to correspond to a sufficiently small cost $c(\bar{x}_t^*)$. Then, it is fixed, and all or part of the original SLAM optimization problem is replaced with the following linear least squares optimization problem (2.4):

$$\begin{aligned} \min_{\bar{x}_t} .c(\bar{x}_t) &= \min_{\bar{x}_t} .C(\bar{x}_t)^\top C(\bar{x}_t) \\ &= \min_{\bar{x}_t} . \left[(\bar{x}_t - \mu)^\top \Sigma^{-1} (\bar{x}_t - \mu) + o(\bar{x}_t - \bar{x}_t^*) \right] \end{aligned}$$

$$\approx \min_{\bar{x}_t} . (\bar{x}_t - \mu)^\top \Sigma^{-1} (\bar{x}_t - \mu) \quad (2.11)$$

where $\mu \in \mathbb{R}^{d_x n + d_f q}$ and $\Sigma \in \mathbb{R}^{(d_x n + d_f q) \times (d_x n + d_f q)}$ are given by:

$$\begin{aligned} \mu &\leftarrow \bar{x}_t^* - (J^\top J)^{-1} J^\top C(\bar{x}_t^*) \\ \Sigma &\leftarrow (J^\top J)^{-1}. \end{aligned}$$

Remark 2.4.1. *Alternatives to the Gauss-Newton algorithm, for computing incremental improvements to the initial guess are available, exist in abundance. These include the standard gradient descent algorithms [90], the Levenberg-Marquardt algorithm [75], or Powell’s dog leg method [88, 71]. However, we focus on the Gauss-Newton algorithm because, as illustrated above, it has a meaningful interpretation in the case of filtering based SLAM algorithms. Moreover, for well-conditioned problems where a good initial estimate is available, the Gauss-Newton algorithm tends to be faster than other methods [80].*

Marginalization

The *marginalization* step reduces the number of variables present in the SLAM problem to reduce computation time. In the context of the above setup, these variables are estimates of the n pose positions and q feature positions, encapsulated in the overall state \bar{x}_t , which are selected to optimize the overall cost $c(\bar{x}_t)$. Intuitively, the marginalization procedure involves the following steps to reduce the number of pose and feature position estimates in our optimization problem. First, we partition the overall state $\bar{x}_t \in \mathbb{R}^{d_x n + d_f q}$ into *marginalized* and *non-marginalized* components. Likewise, we rewrite the overall cost $c(\bar{x}_t) \in \mathbb{R}$ as the sum of two cost terms, one of which depends only on the non-marginalized components, while the other depends on both the marginalized and non-marginalized components. Finally, the marginalization step is completed by approximating the latter cost term as an explicit function of the non-marginalized cost term. This process is described mathematically below.

Among the n poses and p features present in the overall state \bar{x}_t , let the marginalized state $\bar{x}_M \in \mathbb{R}^{d_x M_x + d_f M_f}$ encapsulate the M_x pose positions and M_f feature positions that we wish to discard from our optimization problem, where $1 \leq M_x \leq n - 1$ and $1 \leq M_f \leq q - 1$, and collect the remaining pose and feature position estimates into the non-marginalized state $\bar{x}_K \in \mathbb{R}^{d_x(n - M_x) + d_f(q - M_f)}$. The only state components kept in the optimization problem after marginalization, encapsulated in \bar{x}_K , are poses and feature position estimates that are sufficiently recent or informative to be considered irreplaceable in the optimization problem.

Next, we wish to approximate $c(\bar{x}_t)$ using a cost function that depends entirely on \bar{x}_K . To do this, we first recall that $c(\bar{x}_t)$ is the sum of squared residual terms. By collecting all terms which depend only on the non-marginalized state components \bar{x}_K , we can rewrite $c(\bar{x}_t)$ as the sum of two costs:

$$\begin{aligned} c(\bar{x}_t) &= c(\bar{x}_K, \bar{x}_M) = c_1(\bar{x}_K) + c_2(\bar{x}_K, \bar{x}_M) \\ &= C_1(\bar{x}_K)^\top C_1(\bar{x}_K) + C_2(\bar{x}_K, \bar{x}_M)^\top C_2(\bar{x}_K, \bar{x}_M), \end{aligned}$$

where $c_1 : \mathbb{R}^{d_x(n-M_x)+d_f(q-M_f)} \rightarrow \mathbb{R}$ describes the sum of squared residuals in $c(\bar{x}_t)$ with no dependence on \bar{x}_M , and $C_1 : \mathbb{R}^{d_x(n-M_x)+d_f(q-M_f)} \rightarrow \mathbb{R}^{d_{c,1}}$ denotes the concatenation of such squared residual terms, i.e., $c_1(\bar{x}_K) = C_1(\bar{x}_K)^\top C_1(\bar{x}_K)$, while $c_2 : \mathbb{R}^{d_x n + d_f q} \rightarrow \mathbb{R}$ and $C_2 : \mathbb{R}^{d_x n + d_f q} \rightarrow \mathbb{R}^{d_{c,2}}$ correspond to the remaining terms. The dimensions c_1 and c_2 depend on the specific way in which the state variables in \bar{x}_t and the cost terms in $c(\bar{x}_t)$ are partitioned. For example, consider the cost function (2.3):

$$\begin{aligned} c(\bar{x}_t) &= (\bar{x}_t - \mu_0)^\top \Sigma_0^{-1} (\bar{x}_t - \mu_0) + \sum_{i=t-n+1}^{t-1} (x_{i+1} - g(x_i))^\top \Sigma_w^{-1} (x_{i+1} - g(x_i)) \\ &\quad + \sum_{j=p-q+1}^p \sum_{i=t-n+1}^t (z_{i,j} - h(x_i, f_{t,j}))^\top \Sigma_v^{-1} (z_{i,j} - h(x_i, f_{t,j})). \end{aligned}$$

Suppose we partition the full state vector by $\bar{x}_t := (\bar{x}_K, \bar{x}_M)$, with:

$$\begin{aligned} \mathcal{M}_x &:= \{t-n+1, \dots, t-n+M_x\}, & (\text{poses to marginalize}), \\ \mathcal{M}_f &:= \{p-q+1, \dots, p-q+M_f\}, & (\text{features to marginalize}), \\ \bar{x}_K &:= (x_{t-n+M_x+1}, \dots, x_t, f_{t,p-q+M_f+1}, \dots, f_{t,p}) \in \mathbb{R}^{d_x(n-M_x)+d_f(p-M_f)}, \\ \bar{x}_M &:= (x_{t-n+1}, \dots, x_{t-n+M_x}, f_{t,p-q+1}, \dots, f_{t,p-q+M_f}) \in \mathbb{R}^{d_x M_x + d_f M_f}. \end{aligned}$$

and the cost function by $c(\bar{x}_t) = c_1(\bar{x}_K) + c_2(\bar{x}_M, \bar{x}_K)$, with:

$$\begin{aligned} C_1(\bar{x}_K) &:= \begin{bmatrix} C_{g_{t-n+M_x}}^\top & \cdots & C_{g_{t-1}}^\top & C_{h_{t-n+M_x+1,p-q+M_f+1,t}}^\top & \cdots & C_{h_{t-n+M_x+1,p,t}}^\top & \cdots \\ & & C_{h_{t,p-q+M_f+1,t}}^\top & \cdots & C_{h_{t,p,t}}^\top & & \end{bmatrix}^\top \in \mathbb{R}^{d_x(n-M_x)+d_z(n-M_x)(q-M_f)}, \\ C_2(\bar{x}_K, \bar{x}_M) &:= \begin{bmatrix} C_0(\bar{x}_t)^\top & C_{g_{t-n+1}}^\top & \cdots & C_{g_{t-n+M_x-1}}^\top & C_{h_{t-n+1,p-q+1,t}}^\top & \cdots & C_{h_{t-n+1,p,t}}^\top & \cdots \\ & C_{h_{t-n+M_x,p-q+1,t}}^\top & \cdots & C_{h_{t-n+M_x,p}}^\top & C_{h_{t-n+M_x+1,p-q+1,t}}^\top & \cdots & C_{h_{t-n+M_x+1,p-q+M_f,t}}^\top \\ & & & C_{h_{t,p-q+1,t}} & \cdots & C_{h_{t,p-q+M_f,t}} & & \end{bmatrix}^\top \in \mathbb{R}^{d_x n + d_f q + d_x(M_x-1) + d_z(M_x q + M_f n - M_x M_f)}, \\ c_1(\bar{x}_K) &= C_1(\bar{x}_K)^\top C_1(\bar{x}_K) \in \mathbb{R}, \\ c_2(\bar{x}_K, \bar{x}_M) &= C_2(\bar{x}_K, \bar{x}_M)^\top C_2(\bar{x}_K, \bar{x}_M) \in \mathbb{R}, \end{aligned}$$

where C_0 and each C_{g_i} and $C_{h_{i,j,t}}$ are given by (2.7), (2.8), and (2.9), respectively. Note that in this case, the dimensions of C_1 and C_2 are given by $d_{c,1} = d_x(n-M_x) + d_z(n-M_x)(q-M_f)$ and $d_{c,2} = d_x n + d_f q + d_x(M_x-1) + d_x(M_x q + M_f n - M_x M_f)$, respectively.

The SLAM problem can now be written as:

$$\min_{\bar{x}_t} .c(\bar{x}_t) = \min_{\bar{x}_K} . \left[C_1(\bar{x}_K)^\top C_1(\bar{x}_K) + \min_{\bar{x}_M} . C_2(\bar{x}_K, \bar{x}_M)^\top C_2(\bar{x}_K, \bar{x}_M) \right]$$

To complete the marginalization step, we replace the output of the inner minimization, $\min_{\bar{x}_M} .C_2(\bar{x}_K, \bar{x}_M)^\top C_2(\bar{x}_K, \bar{x}_M)$, with an explicit function of \bar{x}_K . To do so, we apply first-order approximation to the cost term $C_2(\bar{x}_K, \bar{x}_M)^\top C_2(\bar{x}_K, \bar{x}_M)$ to obtain:

$$C_2(\bar{x}_K, \bar{x}_M) = C_2(\bar{x}_K^*, \bar{x}_M^*) + [J_K^\top \quad J_M^\top] \begin{bmatrix} \bar{x}_K - \bar{x}_K^* \\ \bar{x}_M - \bar{x}_M^* \end{bmatrix} + o(\bar{x}_K - \bar{x}_K^*, \bar{x}_M - \bar{x}_M^*), \quad (2.12)$$

where $J_K \in \mathbb{R}^{c_2 \times (d_x(n-M_x) + d_f(q-M_f))}$ denotes the Jacobian of C_2 with respect to \bar{x}_K and $J_M \in \mathbb{R}^{c_2 \times (d_x M_x + d_f M_f)}$ denotes the Jacobian of C_2 with respect to \bar{x}_M . Using (2.12) to approximate the inner minimization in the above optimization problem, we arrive at the optimization problem below, which depends only on the non-marginalized state \bar{x}_K :

$$\begin{aligned} \min_{\bar{x}_t} .c(\bar{x}_t) &= \min_{\bar{x}_K} . \left[C_1(\bar{x}_K)^\top C_1(\bar{x}_K) + \min_{\bar{x}_M} .C_2(\bar{x}_K, \bar{x}_M)^\top C_2(\bar{x}_K, \bar{x}_M) \right] \\ &\approx \min_{\bar{x}_K} . \left(C_1(\bar{x}_K)^\top C_1(\bar{x}_K) + (\bar{x}_K - \mu_K)^\top \Sigma_K^{-1} (\bar{x}_K - \mu_K) \right) \end{aligned} \quad (2.13)$$

where the algorithm defines the mean $\mu_K \in \mathbb{R}^{d_x(n-M_x) + d_f(q-M_f)}$ and the covariance matrix $\Sigma_K \in \mathbb{R}^{(d_x(n-M_x) + d_f(q-M_f)) \times (d_x(n-M_x) + d_f(q-M_f))}$ of \bar{x}_K by assigning:

$$\mu_K \leftarrow \bar{x}_K^* - \Sigma_K J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] C_2(\bar{x}^*) \quad (2.14)$$

$$\Sigma_K \leftarrow (J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] J_K)^{-1}, \quad (2.15)$$

In this chapter and the next, we mathematically prove the optimality of the approximations in (2.12) and (2.13), and examines the implications of varying the frequency in executing the *Gauss-Newton Descent*, *Linear Approximation*, and *Marginalization* steps. In particular, in Chapter 3, we will interpret a selection of mainstream filtering-based SLAM algorithms as the repeated iteration of the above three steps at different rates. Moreover, in Section 3.6, we will illustrate that, by varying the frequencies at which each of the above three steps is performed, we can construct novel SLAM algorithms whose accuracy and computational time interpolate smoothly between those of existing algorithms.

In the sections below, we consider the Gauss-Newton Descent, Linear Approximation, and Marginalization steps in more detail.

2.5 Gauss-Newton Descent

Gauss-Newton descent involves solving for the minimization of $c(\bar{x}_t)$ via Gauss-Newton steps, an iterative linearization method that approximates $c(\bar{x}_t)$ about a given linearization point \bar{x}_t^* by a linear least-squares cost term, i.e.,

$$\min_{\bar{x}_t} .c(\bar{x}_t) = \min_{\bar{x}_t} . \|\bar{x}_t - \mu_t\|_{\Sigma_t^{-1}}^2 + o(\bar{x}_t - \bar{x}_t^*) \quad (2.16)$$

for some $\mu_t \in \mathbb{R}^d$ and $\Sigma_t \in \mathbb{R}^{d \times d}$. The linearization procedure required to obtain $\mu_t \in \mathbb{R}^d$ and $\Sigma_t \in \mathbb{R}^{d \times d}$, as well as the approximation involved, are detailed in the theorem below.

Theorem 2.5.1. (Gauss-Newton Step) Let $\bar{x}_t^* \in \mathbb{R}^d$ denote a given linearization point, and suppose $J := \frac{\partial C}{\partial \bar{x}_t} \in \mathbb{R}^{d_C \times d}$ has full column rank. Then applying a Gauss-Newton step to the cost $c(\bar{x}_t)$, about $\bar{x}_t^* \in \mathbb{R}^d$ yields the new cost:

$$c(\bar{x}_t) = \|\bar{x}_t - \mu_t\|_{\Sigma_t}^2 + o(\bar{x}_t - \bar{x}_t^*),$$

where $\mu_t \in \mathbb{R}^d$ and $\Sigma_t \in \mathbb{R}^{d \times d}$ are given by:

$$\begin{aligned} \Sigma_t &\leftarrow (J^\top J)^{-1}, \\ \mu_t &\leftarrow \bar{x}_t^* - (J^\top J)^{-1} J^\top C(\bar{x}_t^*). \end{aligned}$$

Proof. We have:

$$\begin{aligned} c(\bar{x}_t) &= C(\bar{x}_t)^\top C(\bar{x}_t) \\ &= [C(\bar{x}_t^*) + J(\bar{x}_t - \bar{x}_t^*)]^\top [C(\bar{x}_t^*) + J(\bar{x}_t - \bar{x}_t^*)] + o(\bar{x}_t - \bar{x}_t^*) \\ &= (\bar{x}_t - \mu_t)^\top \Sigma_t^{-1} (\bar{x}_t - \mu_t) + c_0(\bar{x}_t^*) + o(\bar{x}_t - \bar{x}_t^*), \end{aligned}$$

where $c_0(\bar{x}_t^*) \in \mathbb{R}$ denotes a scalar-valued function of \bar{x}_t^* that is independent of the variable \bar{x}_t . This concludes the proof. \square

Algorithm 1: Gauss-Newton Step.

Data: Objective $C^\top C$, linearization point x_t^* .

Result: Mean μ , covariance Σ after a Gauss-Newton step.

- 1 $J \leftarrow \frac{\partial C}{\partial x_t} \Big|_{\mu_t}$
 - 2 $\Sigma_t \leftarrow J^\top J$
 - 3 $\mu_t \leftarrow x_t^* - (J^\top J)^{-1} J^\top C(x_t^*)$
 - 4 **return** μ_t, Σ_t
-

2.6 Marginalization of States

The marginalization step reduces the state dimension in our SLAM algorithm, which helps to reduce the computation time. First, we partition the overall state $\bar{x}_t \in \mathbb{R}^d$ into a *marginalized component* $\bar{x}_{t,M} \in \mathbb{R}^{d_M}$, to be discarded from \bar{x}_t , and a *non-marginalized component* $\bar{x}_{t,K} \in \mathbb{R}^{d_K}$, to be kept ($d = d_K + d_M$.) Then, we partition $c(\bar{x}_t)$ into two cost terms: $c_1(\bar{x}_{t,K})$, which depends only on non-marginalized state components, and $c_2(\bar{x}_{t,K}, \bar{x}_{t,M})$ which depends on both marginalized and non-marginalized state components:

$$\begin{aligned} c(\bar{x}_t) &= c(\bar{x}_K, \bar{x}_M) = c_1(\bar{x}_K) + c_2(\bar{x}_K, \bar{x}_M) \\ &= \|C_1(\bar{x}_K)\|_2^2 + \|C_2(\bar{x}_K, \bar{x}_M)\|_2^2. \end{aligned}$$

Here, $C_1(\overline{x_K}) \in \mathbb{R}^{d_{C,1}}$ and $C_2(\overline{x_K}, \overline{x_M}) \in \mathbb{R}^{d_{C,2}}$ denote the concatenation of residuals associated with $c_1(\overline{x_K})$ and $c_2(\overline{x_K}, \overline{x_M})$ (with $d_C = d_{C,1} + d_{C,2}$). To remove $\overline{x_{t,M}} \in \mathbb{R}^{d_M}$ from the optimization problem, observe that:

$$\begin{aligned} \min_{\overline{x_t}} c(\overline{x_t}) &= \min_{\overline{x_{t,K}}, \overline{x_{t,M}}} \left(c_1(\overline{x_{t,K}}) + c_2(\overline{x_{t,K}}, \overline{x_{t,M}}) \right) \\ &= \min_{\overline{x_{t,K}}} \left(\|C_1(\overline{x_{t,K}})\|_2^2 + \min_{\overline{x_{t,M}}} \|C_2(\overline{x_{t,K}}, \overline{x_{t,M}})\|_2^2 \right). \end{aligned}$$

To remove $\overline{x_{t,M}}$, it suffices to approximate the solution to the inner minimization problem by a linear least-squares cost, i.e.:

$$\min_{\overline{x_{t,K}}} \|C_2(\overline{x_{t,K}}, \overline{x_{t,M}})\|_2^2 \approx \|\overline{x_{t,K}} - \overline{\mu}_{t,K}\|_{\overline{\Sigma}_{t,K}^{-1}}^2$$

for some $\overline{\mu}_{t,K} \in \mathbb{R}^{d_K}$ and $\overline{\Sigma}_{t,K} \in \mathbb{R}^{d_K \times d_K}$. Since $\|C_2(\overline{x_{t,K}}, \overline{x_{t,M}})\|_2^2$ is in general non-convex, we obtain $\overline{\mu}_{t,K}$ and $\overline{\Sigma}_{t,K}$ by minimizing the first-order Taylor expansion of $\|C_2(\overline{x_{t,K}}, \overline{x_{t,M}})\|_2^2$ about some linearization point, instead of minimizing $\|C_2(\overline{x_{t,K}}, \overline{x_{t,M}})\|_2^2$ directly. Below, Theorem 2.6.1 details the derivation of $\overline{\mu}_{t,K}$ and $\overline{\Sigma}_{t,K}$.

Theorem 2.6.1 (Marginalization Step). *Let $\overline{x_t^*} \in \mathbb{R}^d$ denote a given linearization point, and suppose $J := \frac{\partial C}{\partial \overline{x_t}} \in \mathbb{R}^{d_C \times d}$ has full column rank. Define $J_K := \frac{\partial C}{\partial \overline{x_{t,K}}} \in \mathbb{R}^{d_C \times d_K}$ and $J_M := \frac{\partial C}{\partial \overline{x_{t,M}}} \in \mathbb{R}^{d_C \times d_M}$. If $C(\overline{x_{t,M}}, \overline{x_{t,K}})$ were a linear function of $\overline{x_t} = (\overline{x_{t,M}}, \overline{x_{t,K}})$, then applying a Marginalization step to the cost $c(\overline{x_t})$, about the linearization point $\overline{x_t^*} = (\overline{x_{t,K}^*}, \overline{x_{t,M}^*}) \in \mathbb{R}^d$ yields:*

$$\min_{\overline{x_{t,M}}} c(\overline{x_{t,K}}, \overline{x_{t,M}}) = \min_{\overline{x_{t,K}}} \left(c_1(\overline{x_{t,K}}) + \min_{\overline{x_{t,M}}} c_2(\overline{x_{t,K}}, \overline{x_{t,M}}) \right), \quad (2.17)$$

where $\Sigma_{t,K} \in \mathbb{R}^{d_K \times d_K}$ and $\mu_{t,K} \in \mathbb{R}^{d_K}$ are given by:

$$\Sigma_{t,K} := \left(J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] J_K \right)^{-1}, \quad (2.18)$$

$$\mu_{t,K} := \overline{x_{t,K}^*} - \Sigma_{t,K} J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] C_2(\overline{x_t^*}). \quad (2.19)$$

Proof. It suffices to show that:

$$\min_{\overline{x_{t,M}}} c_2(\overline{x_{t,K}}, \overline{x_{t,M}}) = (\overline{x_{t,K}} - \mu_K)^\top \Sigma_K^{-1} (\overline{x_{t,K}} - \mu_K) + c'(\overline{x_t^*}).$$

To do so, we first note that since $C_2(\overline{x_t})$ is linear in $\overline{x_t}$:

$$\begin{aligned} c_2(\overline{x_t}) &= \|C_2(\overline{x_t})\|_2^2 = \|C_2(\overline{x_t^*}) + J_2 \Delta \overline{x_t}\|_2^2 \\ &= \|C_2(\overline{x_t^*}) + J_K \Delta \overline{x_{t,K}} + J_M \Delta \overline{x_{t,M}}\|_2^2. \end{aligned}$$

By the method of least-squares, the optimal $\Delta \overline{x_{t,M}}$ is given by the normal equation:

$$\Delta \overline{x_{t,M}} = -(J_M^\top J_M)^{-1} J_M^\top (C_2(\overline{x_t^*}) + J_K \Delta \overline{x_{t,K}})$$

Substituting back into our expression for $c(\bar{x}_t)$, we have:

$$\begin{aligned}
\min_{\bar{x}_{t,M}} .c_2(\bar{x}_t) &= \|(I - J_M(J_M^\top J_M)^{-1} J_M^\top)(C_2(\bar{x}_t^*) + J_K \Delta \bar{x}_{t,K})\|_2^2 \\
&= (C_2(\bar{x}_t^*) + J_K \Delta \bar{x}_{t,K})^\top [I - J_M(J_M^\top J_M)^{-1} J_M^\top] (C_2(\bar{x}_t^*) + J_K \Delta \bar{x}_{t,K}) \\
&= (\bar{x}_{t,K} - \bar{x}_{t,K}^*)^\top \underbrace{J_K^\top [I - J_M(J_M^\top J_M)^{-1} J_M^\top] J_K}_{:= \Sigma_K^{-1}} (\bar{x}_{t,K} - \bar{x}_{t,K}^*) \\
&\quad + 2(\bar{x}_{t,K} - \bar{x}_{t,K}^*)^\top J_K^\top [I - J_M(J_M^\top J_M)^{-1} J_M^\top] C_2(\bar{x}_t^*) \\
&\quad + C_2(\bar{x}_t^*)^\top [I - J_M(J_M^\top J_M)^{-1} J_M^\top] C_2(\bar{x}_t^*) \\
&= (\bar{x}_{t,K} - \bar{x}_{t,K}^* + \underbrace{\Sigma_K J_K^\top [I - J_M(J_M^\top J_M)^{-1} J_M^\top] C_2(\bar{x}_t^*)}_{:= -\mu_K})^\top \Sigma_K^{-1} \\
&\quad \underbrace{(\bar{x}_{t,K} - \bar{x}_{t,K}^* + \Sigma_K J_K^\top [I - J_M(J_M^\top J_M)^{-1} J_M^\top] C_2(\bar{x}_t^*))}_{:= -\mu_K} \\
&\quad + C_2(\bar{x}_t^*) (I - J_M(J_M^\top J_M)^{-1} J_M^\top) \\
&\quad + c'(\bar{x}_t^*) \\
&= (\bar{x}_{t,K} - \mu_K)^\top \Sigma_K^{-1} (\bar{x}_{t,K} - \mu_K) + c'(\bar{x}_t^*).
\end{aligned}$$

with Σ_K and μ_K as defined in the theorem statement, and $c'(\bar{x}_t^*) \in \mathbb{R}$ independent of \bar{x}_t . \square

Algorithm 2: Marginalization

Data: Objective $f = C^\top C$, vector of variables to marginalize $x_{t,M}$, linearization point x_t^* .

Result: Mean $\mu_{t,K}$ and covariance $\Sigma_{t,K}$ of non-marginalized variables $\bar{x}_{t,K}^*$.

- 1 $C \leftarrow$ subvector of C containing entries dependent on x_M .
 - 2 $J := \begin{bmatrix} J_K & J_M \end{bmatrix} \leftarrow \begin{bmatrix} \frac{\partial C}{\partial x_{t,K}} \big|_{x^*} & \frac{\partial C}{\partial x_{t,M}} \big|_{x^*} \end{bmatrix}$.
 - 3 $\Sigma_{t,K} \leftarrow (J_K^\top [I - J_M(J_M^\top J_M)^{-1} J_M^\top] J_K)^{-1}$
 - 4 $\mu_{t,K} \leftarrow \bar{x}_{t,K}^* - \Sigma_{t,K} J_K^\top [I - J_M(J_M^\top J_M)^{-1} J_M^\top] C(x^*)$
 - 5 **return** $\mu_{t,K}, \Sigma_{t,K}$
-

2.7 Main Algorithm on Manifolds

Our SLAM framework, formulated above on Euclidean spaces, can be straightforwardly extended to a formulation on manifolds. This involves using manifold-related concepts in Section 2.3 to modify the Euclidean-space dynamics and measurement maps in Section 2.2,

as well as the cost functions, Gauss-Newton steps, and marginalization steps in Sections 2.5, 2.6. In particular, when appropriate, plus and minus operations must be replaced with the \boxplus and \boxminus operators.

More precisely, we interpret the SLAM problem on manifolds as the optimization of a cost function $c : \mathcal{X}^n \times \mathcal{F}^q \rightarrow \mathbb{R}$, constructed from residual terms of the same dimension of the minimal coordinates of \bar{x}_t , x_t and z_t . In particular, we must generalize (2.3) to the case where the states, dynamics and measurement maps are defined on and between manifolds. This involves replacing $+$ and $-$ operators with \boxplus and \boxminus operators, when necessary. For example, the sliding window filter window presented in Section 2.2, would be associated with the cost $c : \mathcal{X}^n \times \mathcal{F}^q \rightarrow \mathbb{R}$, given by:

$$\begin{aligned} c(\bar{x}_t) &= \|\bar{x}_t \boxminus \mu_0\|_{\Sigma_0^{-1}}^2 + \sum_{i=t-n+1}^{t-1} \|x_{i+1} \boxminus g(x_i)\|_{\Sigma_w^{-1}}^2 (x_{i+1} \boxminus g(x_i)) \\ &+ \sum_{j=p-q+1}^p \sum_{i=t-n+1}^t \|z_{ij} \boxminus h(x_i, f_k)\|_{\Sigma_v^{-1}}^2 \end{aligned}$$

Similar to Section 2.2, we stack all residual terms into a single residual vector $C(\bar{x}_t)$. For example, for the sliding window filter above, we have:

$$\begin{aligned} C(\bar{x}_t) &:= \left[(\Sigma_0^{-1/2}(\tilde{x}_t \boxminus \mu_0))^\top (\Sigma_w^{-1/2}(x_{t-n+1} \boxminus g(x_{t-n})))^\top \cdots (\Sigma_w^{-1/2}(x_t \boxminus g(x_{t-1})))^\top \right. \\ &\quad \left. (\Sigma_v^{-1/2}(z_{t-n+1,p-q+1} \boxminus h(x_{t-n+1}, f_{p-q+1})))^\top \cdots (\Sigma_v^{-1/2}(z_{t-n+1,p} \boxminus h(x_{t-n+1}, f_p)))^\top \right. \\ &\quad \left. \cdots (\Sigma_v^{-1/2}(z_{t,p-q+1} \boxminus h(x_t, f_{p-q+1})))^\top \cdots (\Sigma_v^{-1/2}(z_{t,p} \boxminus h(x_t, f_p)))^\top \right]^\top \\ &\in \mathbb{R}^{(2n-1)d_x + pd_f + nqd_z}. \end{aligned}$$

As a result, $c(\bar{x}_t) = C(\bar{x}_t)^\top C(\bar{x}_t)$, and the SLAM problem is now reduced to the following nonlinear least squares problem:

$$\min_{\bar{x}_t} .c(\bar{x}_t) = \min_{\bar{x}_t} .C(\bar{x}_t)^\top C(\bar{x}_t) \quad (2.20)$$

Below, we introduce the main algorithmic submodules used to find an approximate solution to (2.20). Let \bar{x}^* be a chosen linearization point. Let $\widehat{C}_{\bar{x}^*} := C \circ \pi_{\bar{x}^*}^{-1}$ be the coordinate representation of the function C near \bar{x}^* . Recall that $\widehat{C}_{\bar{x}^*}$ is simply a function from one Euclidean space to another. We can now Taylor expand to write:

$$C(\bar{x}) = (C \circ \pi_{\bar{x}^*}^{-1})(\pi_{\bar{x}^*}(\bar{x})) = \widehat{C}_{\bar{x}^*}(\Delta\bar{\chi}) = \widehat{C}_{\bar{x}^*}(0) + J\Delta\bar{\chi} + o(\Delta\bar{\chi}),$$

where $\Delta\bar{\chi} = \bar{x} \boxminus \bar{x}^*$ and J is the Jacobian of $\widehat{C}_{\bar{x}^*}$ with respect to $\Delta\bar{\chi}$ evaluated at zero. We then apply a modified version of the algorithms from Section 2.4:

1. *Gauss-Newton Descent*: Used to update the current linearization point $\bar{x}^{\{k\}}$ to a new linearization point $\bar{x}^{\{k+1\}}$.

$$\bar{x}^{(k+1)} \leftarrow \bar{x}^{(k)} \boxplus \left(- (J^T J)^{-1} J^T C(\bar{x}^{(k)}) \right) \quad (2.21)$$

After Gauss-Newton steps have been taken, the linearization point \bar{x}^* is fixed, and all or part of the original optimization problem, i.e., $\min_{\bar{x}} c(\bar{x}) = \min_{\bar{x}} C(\bar{x})^\top C(\bar{x})$, is replaced with the following linear least squares optimization problem:

$$\min_{\bar{x}} .(\bar{x} \boxminus \mu)^\top \Sigma^{-1} (\bar{x} \boxminus \mu) \quad (2.22)$$

where the algorithm assigns:

$$\begin{aligned} \mu &\leftarrow \bar{x}^* \boxplus (J^\top J)^{-1} J^\top C(\bar{x}^*) \\ \Sigma &\leftarrow (J^\top J)^{-1}. \end{aligned}$$

2. *Marginalization*: Used to remove variables \bar{x}_M from the optimization problem by applying linear approximation to C —in particular, the optimization problem $\min_{\bar{x}} .c(\bar{x})$ is approximated by:

$$\min_{\bar{x}_K} .(\bar{x}_K \boxminus \mu_K)^\top \Sigma_K^{-1} (\bar{x}_K \boxminus \mu_K),$$

where the algorithm assigns:

$$\begin{aligned} \mu_K &\leftarrow \bar{x}_K^* \boxplus \left(- \Sigma_K J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] C_2(\bar{x}^*) \right) \\ \Sigma_K &\leftarrow \left(J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] J_K \right)^{-1}, \end{aligned}$$

For characterizations of the behavior of Jacobian matrices under the boxplus \boxplus and boxminus \boxminus operators, we refer the reader to [95], Appendix A.

Gauss-Newton Descent

Gauss-Newton steps update the current linearization point $\bar{x}^{\{k\}}$ to a new linearization point $\bar{x}^{\{k+1\}}$.

$$\bar{x}^{(k+1)} \leftarrow \bar{x}^{(k)} \boxplus \left(- (J^T J)^{-1} J^T C(\bar{x}^{(k)}) \right) \quad (2.23)$$

After Gauss-Newton steps have been taken, the linearization point \bar{x}^* is fixed, and all or part of the original optimization problem, reproduced below:

$$\min_{\bar{x}} c(\bar{x}) = \min_{\bar{x}} C(\bar{x})^\top C(\bar{x})$$

is replaced with the following linear least squares optimization problem:

$$\min_{\bar{x}} .(\bar{x} \boxminus \mu)^\top \Sigma^{-1} (\bar{x} \boxminus \mu)$$

where μ and Σ are given by:

$$\begin{aligned} \mu &\leftarrow \bar{x}^* \boxplus (J^\top J)^{-1} J^\top C(\bar{x}^*) \\ \Sigma &\leftarrow (J^\top J)^{-1}. \end{aligned}$$

Marginalization

Marginalization removes variables \overline{x}_M from the optimization problem by applying linear approximation to C —in particular, the optimization problem $\min_{\overline{x}} .c(\overline{x})$ is approximated by:

$$\min_{\overline{x}_K} .(\overline{x}_K \boxplus \mu_K)^\top \Sigma_K^{-1} (\overline{x}_K \boxplus \mu_K),$$

where μ and Σ are defined as:

$$\begin{aligned} \mu_K &\leftarrow \overline{x}_K^* \boxplus \left(-\Sigma_K J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] C_2(\overline{x}^*) \right) \\ \Sigma_K &\leftarrow \left(J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] J_K \right)^{-1}. \end{aligned}$$

Chapter 3

Equivalence of Filtering and Optimization

In this chapter, we build upon the notation and terminology established in Chapter 2 to illustrate the equivalence of filtering and batch optimization-based SLAM algorithms. In particular, we will focus on the Extended Kalman Filter (EKF, in Section 3.2) and Multi-State Constrained Kalman Filter (MSCKF, in Section 3.4), as examples of widely-used filtering-based SLAM algorithms, and establish analogous optimization-based algorithms using our unified framework. Although similar results exist in the optimization literature [6], they do not analyze algorithmic submodules unique to SLAM, e.g., feature incorporation, processing, and discarding.

Below, we begin by presenting the classical formulation of EKF SLAM, as portrayed in the literature.

3.1 Extended Kalman Filter (EKF) on Euclidean Spaces, Standard Formulation

The Extended Kalman Filter (EKF), whose standard formulation is presented in Algorithm 3, is an iterative algorithm for updating estimates of the current pose x_t (i.e. $n = 1$) and positions of all observed features at the current time, $f_t := (f_1, \dots, f_p) \in \mathbb{R}^{pd_f}$. This corresponds to the sliding window filter in our formulation, with $n = 1$ and $q = p$. Below, as an application of our optimization-based SLAM framework, we present the dynamics and measurement maps of the EKF algorithm in \mathbb{R}^2 , as well as the associated cost functions. Dimension-wise, in its standard formulation, the 2D EKF is an instantiation of Algorithm 3 with $d_x = 3$, $d_f = 2$, and $d_z = 2$. To unify our notation, we will suppose that d_x, d_f, d_z assume these values throughout the rest of this section.

Let $x_t := (x_t^1, x_t^2, \theta_t) \in \mathbb{R}^{d_x}$ denote the *robot pose*, comprising its position and angle in \mathbb{R}^{d_f} , let $f_k := (f_k^1, f_k^2) \in \mathbb{R}^{d_f}$ denote the position of each *feature* $f_k \in \{f_1, \dots, f_p\}$ visible at time t , and let $z_{t,k} := (z_{t,k}^1, z_{t,k}^2) \in \mathbb{R}^{d_z}$ denote the measurement of feature f_k at time t .

The dynamics map $g : \mathbb{R}^{d_x} \rightarrow \mathbb{R}^{d_x}$, with $\dot{x}_t = g(x_t)$ is obtained by performing numerical integration on the continuous-time dynamics:

$$\begin{aligned}\dot{x}_t^1 &= v \cos \theta + w_t^1, \\ \dot{x}_t^2 &= v \sin \theta + w_t^2, \\ \dot{\theta}_t &= \omega + w_t^3,\end{aligned}$$

where $w_t := (w_t^1, w_t^2, w_t^3) \in \mathbb{R}^{d_x}$ denotes additive zero-mean Gaussian noise on the (x, y, θ) coordinates of the state variable, respectively, with joint covariance $w_t \sim \mathcal{N}(0, \Sigma_w)$ for some covariance matrix $\Sigma_w \in \mathbb{R}^{d_x \times d_x}$, $\Sigma_w \succ 0$.

The measurement map $h : \mathbb{R}^{d_x} \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_z}$ is given by:

$$\begin{aligned}z_{t,k}^1 &= f_k^1 - x_t^1 + v_t^1, \\ z_{t,k}^2 &= f_k^2 - x_t^2 + v_t^2,\end{aligned}$$

where $v_t := (v_t^1, v_t^2) \in \mathbb{R}^{d_z}$ denotes additive zero-mean Gaussian noise on the measurements $z_{t,j}^1, z_{t,j}^2 \in \mathbb{R}$, respectively, with joint covariance $v_t \sim \mathcal{N}(0, \Sigma_v)$ for some covariance matrix $\Sigma_v \in \mathbb{R}^{d_z \times d_z}$, $\Sigma_v \succ 0$. The measurement vector $z_t \in \mathbb{R}^{pd_f}$ is then given by concatenating each of the q residual measurements obtained at time t , i.e. $z_t := (z_{t,1}, \dots, z_{t,p}) \in \mathbb{R}^{pd_z}$.

Algorithm 3: Extended Kalman Filter SLAM, Standard Formulation.

Data: Prior distribution on $x_0 \in \mathbb{R}^{d_x}$: $\mathcal{N}(\mu_0, \Sigma_0)$, dynamics and measurement noise covariances $\Sigma_w \in \mathbb{R}^{d_x \times d_x}$, $\Sigma_v \in \mathbb{R}^{d_z \times d_z}$, (discrete-time) dynamics map $g : \mathbb{R}^{d_x} \rightarrow \mathbb{R}^{d_x}$, measurement map $h : \mathbb{R}^{d_x} \times \mathbb{R}^{pd_f} \rightarrow \mathbb{R}^{d_z}$, time horizon $T \in \mathbb{N}$.

Result: Estimates \hat{x}_t for all desired timesteps $t \leq T$.

```

1 for  $t = 0, \dots, T$  do
2   if detect new feature measurements  $z_{t,p+1:p+p'} := (z_{t,p+1}, \dots, z_{t,p+p'}) \in \mathbb{R}^{p'd_z}$ 
   then
3      $\mu_t, \Sigma_t, p \leftarrow$  Alg. 5, EKF feature augmentation  $(\mu_t, \Sigma_t, p, z_{t,p+1:p+p'}, h(\cdot))$ 
4   end
5    $z_{t,1:p} := (z_{t,1}, \dots, z_{t,p}) \in \mathbb{R}^{pd_z} \leftarrow$  New measurements of existing features.
6    $\bar{\mu}_t, \bar{\Sigma}_t \leftarrow$  Alg. 6, EKF feature update  $(\bar{\mu}_t, \bar{\Sigma}_t, z_{t,1:p}, h(\cdot))$ .
7   if  $t < T$  then
8      $\mu_{t+1}, \Sigma_{t+1} \leftarrow$  Alg. 7, EKF state propagation  $(\mu_t, \Sigma_t, g(\cdot))$ 
9   end
10 end
11 return  $\hat{x}_0, \dots, \hat{x}_T \in \mathbb{R}^{d_x}$ .
```

3.2 Extended Kalman Filter (EKF) on Euclidean Spaces, in an Optimization Framework

At each time t , the EKF SLAM algorithm on Euclidean spaces maintains the full state vector $\tilde{x}_t := (x_t, f_1, \dots, f_p) \in \mathbb{R}^{d_x + pd_f}$, consisting of the most recent state $x_t \in \mathbb{R}^{d_x}$ and feature position estimates $f_1, \dots, f_p \in \mathbb{R}^{d_f}$. In other words, the EKF SLAM algorithm follows our optimization-based SLAM algorithm, with $n = 1$ and $q = p$. At initialization ($t = 0$), no feature has been detected ($p = 0$), and the EKF full state is simply the initial state $\tilde{x}_0 = x_0 \in \mathbb{R}^{d_x}$, with mean $\mu_0 \in \mathbb{R}^{d_x}$ and covariance $\Sigma_0 \in \mathbb{R}^{d_x \times d_x}$. The corresponding initial running cost, $c_{EKF,0,0} : \mathbb{R}^{d_x} \rightarrow \mathbb{R}$, is thus defined by:

$$c_{EKF,0,0} = \|\tilde{x}_0 - \mu_0\|_{\Sigma_0^{-1}}^2.$$

Suppose, at the current time t , the running cost $c_{EKF,t,0} : \mathbb{R}^{d_x + pd_f} \rightarrow \mathbb{R}^{d_x + pd_f}$ is:

$$c_{EKF,t,0} = \|\tilde{x}_t - \mu_t\|_{\Sigma_t^{-1}}^2,$$

where $\tilde{x}_t := (x_t, f_1, \dots, f_p) \in \mathbb{R}^{d_x + pd_f}$ denotes the EKF full state at time t , with mean $\mu_t \in \mathbb{R}^{d_x + pd_f}$ and covariance $\Sigma_t \in \mathbb{R}^{(d_x + pd_f) \times (d_x + pd_f)}$. Below, we introduce the *feature augmentation*, *feature update*, and *state propagation* submodules of the EKF SLAM Algorithm. Each of these steps refines the mean and covariance of the EKF full states computed by applying the Gauss-Newton, Linear Approximation, and Marginalization steps.

First, the *feature augmentation step* appends the position estimates of newly observed features $f_{p+1}, \dots, f_{p+p'} \in \mathbb{R}^{d_f}$ to the EKF full state \tilde{x}_t , and updates its mean and covariance. In particular, feature measurements $z_{t,p+1}, \dots, z_{t,p+p'} \in \mathbb{R}^{d_z}$ are assimilated by adding measurement residual terms, creating a new cost $c_{EKF,t,1} : \mathbb{R}^{d_x + (p+p')d_f} \rightarrow \mathbb{R}$:

$$c_{EKF,t,1}(\tilde{x}_t, f_{p+1}, \dots, f_{p+p'}) := \|\tilde{x}_t - \mu_t\|_{\Sigma_t^{-1}}^2 + \sum_{k=p+1}^{p+p'} \|z_{t,k} - h(x_t, f_k)\|_{\Sigma_v^{-1}}^2.$$

In effect, $c_{EKF,t,1}$ appends positions of new features to \tilde{x}_t , and constrains it using feature measurements residuals. We then replace p with $p + p'$.

Next, the *feature update step* uses measurements of features originally contained in \tilde{x}_t to update the mean and covariance of \tilde{x}_t . More precisely, feature measurements $z_{t,1}, \dots, z_{t,p} \in \mathbb{R}^{d_z}$, of the p features f_1, \dots, f_p included in \tilde{x}_t , are introduced by incorporating the corresponding measurement residuals $\{(z_{t,k} - h(x_t, f_k))^\top \Sigma_v^{-1} (z_{t,k} - h(x_t, f_k)) \mid k = 1, \dots, p\}$ to create a new cost $c_{EKF,t,2} : \mathbb{R}^{d_x + pd_f} \rightarrow \mathbb{R}$:

$$c_{EKF,t,2}(\tilde{x}_t) := \|\tilde{x}_t - \mu_t\|_{\Sigma_t^{-1}}^2 + \sum_{k=1}^p \|z_{t,k} - h(x_t, f_k)\|_{\Sigma_v^{-1}}^2.$$

A Gauss-Newton step then constructs an updated mean $\bar{\mu}_t \in \mathbb{R}^{d_x+pd_f}$ and covariance $\bar{\Sigma}_t \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$ for \tilde{x}_t , creating a new cost $c_{EKF,t,3} : \mathbb{R}^{d_x+pd_f} \rightarrow \mathbb{R}$:

$$c_{EKF,t,3}(\tilde{x}_t) := \|\tilde{x}_t - \bar{\mu}_t\|_{\bar{\Sigma}_t^{-1}}^2,$$

which returns the running cost to the form of $c_{EKF,t,0}$.

The *state propagation* step propagates the EKF full state forward by one time step, via the EKF state propagation map $g : \mathbb{R}^{d_x+pd_f} \rightarrow \mathbb{R}^{d_x+pd_f}$. To propagate \tilde{x}_t forward in time, we add the dynamics residual, creating a new cost $c_{EKF,t,4} : \mathbb{R}^{2d_x+pd_f} \rightarrow \mathbb{R}$:

$$c_{EKF,t,4}(\tilde{x}_t, x_{t+1}) := \|\tilde{x}_t - \bar{\mu}_t\|_{\bar{\Sigma}_t^{-1}}^2 + \|x_{t+1} - g(x_t)\|_{\Sigma_w^{-1}}^2.$$

In effect, $c_{EKF,t,4}$ appends the new state $x_{t+1} \in \mathbb{R}^{d_x}$ to \tilde{x}_t , while adding a new constraint posed by the dynamics residuals. A marginalization step, with $\tilde{x}_{t,K} := (x_{t+1}, f_1, \dots, f_p) \in \mathbb{R}^{d_x+pd_f}$ and $\tilde{x}_{t,M} := x_t \in \mathbb{R}^{d_x}$, then removes the previous state $x_t \in \mathbb{R}^{d_x}$ from the running cost. This step produces a mean $\mu_{t+1} \in \mathbb{R}^{d_x+pd_f}$ and a covariance $\Sigma_{t+1} \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$ for the new EKF full state, $\tilde{x}_{t+1} := \tilde{x}_{t,K}$. The running cost is updated to $c_{EKF,t+1,0} : \mathbb{R}^{d_x+pd_f} \rightarrow \mathbb{R}$:

$$c_{EKF,t+1,0}(\tilde{x}_{t+1}) := \|\tilde{x}_{t+1} - \mu_{t+1}\|_{\Sigma_{t+1}^{-1}}^2,$$

which returns the running cost to the form of $c_{EKF,t,0}$.

The theorems below establish that the feature augmentation, feature update, and state propagation steps of the EKF, presented above in our optimization framework, correspond precisely to those presented in the standard EKF SLAM algorithm (Alg. 3) [105, 98].

Theorem 3.2.1. *The feature augmentation step of standard EKF SLAM (Alg. 5) is equivalent to applying a Gauss-Newton step to $c_{EKF,t,1} : \mathbb{R}^{d_x+(p+p')d_f} \rightarrow \mathbb{R}$, with:*

$$c_{EKF,t,1}(\tilde{x}_t, f_{p+1}, \dots, f_{p+p'}) = \|\tilde{x}_t - \mu_t\|_{\Sigma_t^{-1}}^2 + \sum_{k=p+1}^{p+p'} \|z_{t,k} - h(x_t, f_k)\|_{\Sigma_v^{-1}}^2.$$

Proof. To simplify the analysis below, we assume all degrees of freedom of new features are observed. More specifically, we assume the existence of an *inverse observation map* $\ell : \mathbb{R}^{d_x} \times \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_f}$, satisfying $h(x_t, \ell(x_t, z_t)) = z_t$ for each $x_t \in \mathbb{R}^{d_x}$, $z_t \in \mathbb{R}^{d_z}$, which directly generates position estimates of new features from their feature measurements and the current pose, by effectively “inverting” the measurement map $h : \mathbb{R}^{d_x} \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_z}$ [98]. When full observations are unattainable, the missing degrees of freedom are introduced as a prior to the system [98]; in this case, similar results follow.

First, to simplify notation, define:

$$\begin{aligned} z_{t,p+1:p+p'} &= (z_{t,p+1}, \dots, z_{t,p+p'}) \in \mathbb{R}^{p'd_z}, \\ f_{p+1:p+p'} &= (f_{p+1}, \dots, f_{p+p'}) \in \mathbb{R}^{p'd_f}, \end{aligned}$$

$$\begin{aligned}\tilde{h}(x_t, f_{p+1:p+p'}) &:= (h(x_t, f_{p+1}), \dots, h(x_t, f_{p+p'})) \in \mathbb{R}^{p'd_z}, \\ \tilde{\Sigma}_v &= \text{diag}\{\Sigma_v, \dots, \Sigma_v\} \in \mathbb{R}^{p'd_z \times p'd_z}.\end{aligned}$$

We can now rewrite the cost $c_{EKF,t,1}$ as:

$$c_{EKF,t,1}(\tilde{x}_t, f_{p+1:p+p'}) = \|\tilde{x}_t - \mu_t\|_{\tilde{\Sigma}_t^{-1}}^2 + \|z_{t,p+1:p+p'} - \tilde{h}(x_t, f_{p+1:p+p'})\|_{\tilde{\Sigma}_v^{-1}}^2.$$

To apply a Gauss-Newton step, our first task is to find a vector $C_1(\tilde{x}_t, f_{p+1:p+p'})$ of an appropriate dimension such that $c_{EKF,t,1}(\tilde{x}_t, f_{p+1:p+p'}) = C_1(\tilde{x}_t, f_{p+1:p+p'})^\top C_1(\tilde{x}_t, f_{p+1:p+p'})$. A natural choice is furnished by $C_1(\tilde{x}_t, f_{p+1:p+p'}) \in \mathbb{R}^{d_x + pd_f + p'd_z}$, as defined below:

$$C_1(\tilde{x}_t, f_{p+1:p+p'}) := \begin{bmatrix} \Sigma_t^{-1/2}(\tilde{x}_t - \mu_t) \\ \Sigma_v^{-1/2}(z_{t,p+1:p+p'} - \tilde{h}(x_t, f_{p+1:p+p'})) \end{bmatrix}.$$

Thus, our parameters for the Gauss-Newton algorithm submodule are:

$$\begin{aligned}\tilde{x}_t^* &:= (x_t^*, f_{1:p}^*, f_{p+1:p+p'}^*) = (\bar{\mu}_t, \ell(x_t^*, z_{t,p+1}), \dots, \ell(x_t^*, z_{t,p+p'})) \in \mathbb{R}^{d_x + (p+p')d_f}, \\ &\text{where } x_t^* \in \mathbb{R}^{d_x}, f_{1:p}^* \in \mathbb{R}^{pd_f}, f_{p+1:p+p'}^* \in \mathbb{R}^{p'd_f}, \\ C_1(\tilde{x}_t^*) &= \begin{bmatrix} \Sigma_t^{-1/2}(\tilde{x}_t^* - \mu_t) \\ \tilde{\Sigma}_v^{-1/2}(z_{t,p+1:p+p'} - \tilde{h}(x_t^*, f_{p+1:p+p'}^*)) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{d_x + pd_f + p'd_z}, \\ J &= \begin{bmatrix} \Sigma_t^{-1/2} & O \\ -\tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,x} [I \ O] & -\tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f} \end{bmatrix} \in \mathbb{R}^{(d_x + pd_f + p'd_z) \times (d_x + (p+p')d_f)},\end{aligned}$$

where $\tilde{H}_t := [\tilde{H}_{t,x} \ \tilde{H}_{t,f}] \in \mathbb{R}^{p'd_z \times (d_x + p'd_f)}$ is defined as the Jacobian of $\tilde{h} : \mathbb{R}^{d_x} \times \mathbb{R}^{p'd_f} \rightarrow \mathbb{R}^{p'd_z}$ at $(x_t^*, f_{p+1:p+p'}^*) \in \mathbb{R}^{d_x + p'd_f}$, with $\tilde{H}_{t,x} \in \mathbb{R}^{p'd_z \times d_x}$ and $\tilde{H}_{t,f} \in \mathbb{R}^{p'd_z \times pd_f}$. By Algorithm 1, the Gauss-Newton update is thus:

$$\begin{aligned}\Sigma_t \leftarrow (J^\top J)^\dagger &= \left(\begin{bmatrix} \Sigma_t^{-1/2} & -\begin{bmatrix} I \\ O \end{bmatrix} \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1/2} \\ O & -\tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f} \end{bmatrix} \begin{bmatrix} \Sigma_t^{-1/2} & O \\ -\tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,x} [I \ O] & -\tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f} \end{bmatrix} \right)^\dagger \\ &= \begin{bmatrix} \Sigma_t^{-1} + \begin{bmatrix} I \\ O \end{bmatrix} \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,x} [I \ O] & \begin{bmatrix} I \\ O \end{bmatrix} \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f} \\ \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,x} [I \ O] & \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f} \end{bmatrix}^\dagger \\ &= \begin{bmatrix} \Omega_{t,xx} + \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,x} & \Omega_{t,xf} & \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f} \\ \Omega_{t,fx} & \Omega_{t,ff} & O \\ \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,x} & O & \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f} \end{bmatrix}^\dagger, \\ \bar{\mu}_t \leftarrow \tilde{x}_t^* - (J^\top J)^\dagger J^\top C_1(\tilde{x}_t^*) &= (\bar{\mu}_t, \ell(x_t^*, z_{t,p+1}), \dots, \ell(x_t^*, z_{t,p+p'})),\end{aligned} \tag{3.1}$$

where \dagger denotes the Moore-Penrose pseudoinverse.

Here, we have defined $\Omega_{t,xx} \in \mathbb{R}^{d_x \times d_x}$, $\Omega_{t,xf} = \Omega_{t,fx}^\top \in \mathbb{R}^{d_x \times pd_f}$ and $\Omega_{t,ff} \in \mathbb{R}^{pd_f \times pd_f}$ by:

$$\begin{bmatrix} \Omega_{t,xx} & \Omega_{t,xf} \\ \Omega_{t,fx} & \Omega_{t,ff} \end{bmatrix} := \begin{bmatrix} \Sigma_{t,xx} & \Sigma_{t,xf} \\ \Sigma_{t,fx} & \Sigma_{t,ff} \end{bmatrix}^{-1} \quad (3.2)$$

To conclude the proof, we must show that (3.1) is identical to the update equations for covariance matrix in the standard formulation of the Extended Kalman Filter algorithm, i.e., we must show that:

$$\begin{bmatrix} \Sigma_{t,xx} & \Sigma_{t,xf} & \Sigma_{t,xx} L_x^\top \\ \Sigma_{t,fx} & \Sigma_{t,ff} & \Sigma_{t,fx} L_x^\top \\ L_x \Sigma_{t,xx} & L_x \Sigma_{t,xf} & L_x \Sigma_{t,xx} L_x^\top + L_z \Sigma_v L_z^\top \end{bmatrix} = \begin{bmatrix} \Omega_{t,xx} + \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,x} & \Omega_{t,xf} & \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f} \\ \Omega_{t,fx} & \Omega_{t,ff} & O \\ \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,x} & O & \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f} \end{bmatrix}^\dagger$$

This follows by applying (3.2), as well as the matrix equalities resulting from taking the derivative of the equation $z_t := h(x_t, \ell(x_t, z_t))$ with respect to $x_t \in \mathbb{R}^{d_x}$ and $z_t \in \mathbb{R}^{d_z}$, respectively:

$$\begin{aligned} I &= \tilde{H}_{t,f} L_z, \\ O &= \tilde{H}_{t,x} + \tilde{H}_{t,f} L_x. \end{aligned}$$

□

Theorem 3.2.2. *The feature update step of standard EKF SLAM (Alg. 6) is equivalent to applying a Gauss-Newton step on $c_{EKF,t,2} : \mathbb{R}^{d_x + pd_f} \rightarrow \mathbb{R}$, with:*

$$c_{EKF,t,2}(\tilde{x}_t) := \|\tilde{x}_t - \mu_t\|_{\Sigma_t^{-1}}^2 + \sum_{k=1}^p \|z_{t,k} - h(x_t, f_k)\|_{\Sigma_v^{-1}}^2.$$

Proof. First, to simplify notation, define:

$$\begin{aligned} z_{t,1:p} &:= (z_{t,1}, \dots, z_{t,p}) \in \mathbb{R}^{pd_z}, \\ f_{1:p} &:= (f_1, \dots, f_p) \in \mathbb{R}^{pd_f}, \\ \tilde{h}(x_t, f_{1:p}) &:= (h(x_t, f_1), \dots, h(x_t, f_p)) \in \mathbb{R}^{pd_z}, \\ \tilde{\Sigma}_v &:= \text{diag}\{\Sigma_v, \dots, \Sigma_v\} \in \mathbb{R}^{pd_z \times pd_z}. \end{aligned}$$

We can then rewrite the cost as:

$$c_{EKF,t,2}(\tilde{x}_t) = \|\tilde{x}_t^* - \mu_t\|_{\Sigma_t^{-1}}^2 + \|z_{t,1:p} - \tilde{h}(\tilde{x}_t^*)\|_{\tilde{\Sigma}_v^{-1}}^2.$$

To apply a Gauss-Newton step, our first task is to find a vector $C_2(\tilde{x}_t)$ of an appropriate dimension such that $c_{EKF,t,2}(\tilde{x}_t) = C_2(\tilde{x}_t)^\top C_2(\tilde{x}_t)$. A natural choice is furnished by $C_2(\tilde{x}_t) \in \mathbb{R}^{d_x + pd_f + pd_z}$, as defined below:

$$C_2(\tilde{x}_t) := \begin{bmatrix} \Sigma_t^{-1/2}(\tilde{x}_t - \mu_t) \\ \tilde{\Sigma}_v^{-1/2}(z_{t,1:p} - \tilde{h}(\tilde{x}_t)) \end{bmatrix}.$$

Thus, our parameters for the Gauss-Newton algorithm submodule are:

$$\begin{aligned}\tilde{x}_t^* &= \mu_t \in \mathbb{R}^{d_x+pd_f}, \\ C_2(\tilde{x}_t^*) &= \begin{bmatrix} \Sigma_t^{-1/2}(\tilde{x}_t^* - \mu_t) \\ \tilde{\Sigma}_v^{-1/2}(z_{t,1:p} - \tilde{h}(\tilde{x}_t^*)) \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\Sigma}_v^{-1/2}(z_{t,1:p} - \tilde{h}(\mu_t)) \end{bmatrix} \in \mathbb{R}^{d_x+pd_f+pd_z}, \\ J &= \begin{bmatrix} \Sigma_t^{-1/2} \\ -\tilde{\Sigma}_v^{-1/2}H_t \end{bmatrix} \in \mathbb{R}^{(d_x+pd_f+pd_z) \times (d_x+pd_f)},\end{aligned}$$

where $\tilde{H}_t \in \mathbb{R}^{pd_z} \times \mathbb{R}^{d_x+pd_f}$ is defined as the Jacobian of $\tilde{h} : \mathbb{R}^{d_x} \times \mathbb{R}^{pd_f} \rightarrow \mathbb{R}^{pd_z}$ at $\tilde{x}_t^* \in \mathbb{R}^{d_x+pd_f}$. By Algorithm 1, the Gauss-Newton update is thus given by:

$$\begin{aligned}\bar{\Sigma}_t &\leftarrow (J^\top J)^{-1} \\ &= (\Sigma_t^{-1} + H_t^\top \tilde{\Sigma}_v^{-1} H_t)^{-1} \\ &= \Sigma_t - \Sigma_t H_t^\top (\tilde{\Sigma}_v + H_t \Sigma_t H_t^\top)^{-1} H_t \Sigma_t, \\ \bar{\mu}_t &\leftarrow \mu_t - (J^\top J)^{-1} J^\top C_2(\tilde{x}_t^*) \\ &= \mu_t - (\Sigma_t^{-1} + H_t^\top \tilde{\Sigma}_v^{-1} H_t)^{-1} \begin{bmatrix} \Sigma_t^{-1/2} & -H_t^\top \tilde{\Sigma}_v^{-1/2} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \tilde{\Sigma}_v^{-1/2}(z_{t,1:p} - \tilde{h}(\mu_t)) \end{bmatrix} \\ &= \mu_t + (\Sigma_t^{-1} + H_t^\top \tilde{\Sigma}_v^{-1} H_t)^{-1} H_t^\top \tilde{\Sigma}_v^{-1} (z_{t,1:p} - \tilde{h}(\mu_t)), \\ &= \mu_t + \tilde{\Sigma}_v^{-1} H_t^\top (H_t \Sigma_t H_t^\top + \tilde{\Sigma}_v)^{-1} (z_{t,1:p} - \tilde{h}(\mu_t)),\end{aligned}$$

which are identical to the feature update equations for the mean and covariance matrix in the Extended Kalman Filter algorithm, i.e. (4) and (5) respectively. Note that, in the final step, we have used a variant of the Woodbury Matrix Identity. \square

Theorem 3.2.3. *The state propagation step of standard EKF SLAM (Alg. 7) is equivalent to applying a Marginalization step to $c_{EKF,t,4} : \mathbb{R}^{2d_x+pd_f} \rightarrow \mathbb{R}$, with:*

$$c_{EKF,t,4}(\tilde{x}_t, x_{t+1}) := \|\tilde{x}_t - \bar{\mu}_t\|_{\bar{\Sigma}_t^{-1}}^2 + \|x_{t+1} - g(x_t)\|_{\Sigma_w^{-1}}^2.$$

where $\tilde{x}_{t,K} := (x_{t+1}, f_1, \dots, f_p) \in \mathbb{R}^{d_x+pd_f}$ and $\tilde{x}_{t,M} = x_t \in \mathbb{R}^{d_x}$.

Proof. Intuitively, the state propagation step marginalizes out $x_t \in \mathbb{R}^{d_x}$ and retain $x_{t+1} \in \mathbb{R}^{d_x}$. In other words, in the notation of our Marginalization algorithm submodule, we have:

$$\begin{aligned}\tilde{x}_{t,K} &= \tilde{x}_{t+1} \in \mathbb{R}^{d_x+pd_f}, \\ \tilde{x}_{t,M} &= x_t \in \mathbb{R}^{d_x+pd_f}.\end{aligned}$$

To apply a marginalization step, our first task is to find vectors $C_K(x_K) = C_K(\tilde{x}_t)$ and $C_M(x_K, x_M) = C_M(\tilde{x}_t, x_{t+1})$ of appropriate dimensions such that:

$$c_{EKF,t,4}(\tilde{x}_t, x_{t+1}) = C_K(x_{t+1})^\top C_K(x_{t+1}) + C_M(\tilde{x}_t, x_{t+1})^\top C_M(\tilde{x}_t, x_{t+1}).$$

A natural choice is furnished by $C_K(x_{t+1}) \in \mathbb{R}$ and $C_M(\tilde{x}_t, x_{t+1}) \in \mathbb{R}^{d_x}$, as defined below:

$$\begin{aligned} c_K(x_{t+1}) &= 0 \\ c_M(\tilde{x}_t, x_{t+1}) &= \|\tilde{x}_t - \bar{\mu}_t\|_{\bar{\Sigma}_t^{-1}}^2 + \|x_{t+1} - g(x_t)\|_{\Sigma_w^{-1}}^2. \end{aligned}$$

where we have identified the following parameters, in the language of a Marginalization step (Section 2.2):

$$\begin{aligned} C_K(\tilde{x}_{t,K}) &= 0 \in \mathbb{R} \\ C_M(\tilde{x}_{t,K}, \tilde{x}_{t,M}) &= \begin{bmatrix} \bar{\Sigma}_t^{-1/2}(\tilde{x}_t - \bar{\mu}_t) \\ \Sigma_w^{-1/2}(x_{t+1} - g(x_t)) \end{bmatrix} \in \mathbb{R}^{2d_x + pd_f}. \end{aligned}$$

For convenience, we will define the pose and feature track components of the mean $\mu_t \in \mathbb{R}^{d_x + pd_f}$ by $\mu_t := (\mu_{t,x}, \mu_{t,f}) \in \mathbb{R}^{d_x + pd_f}$, with $\mu_{t,x} \in \mathbb{R}^{d_x}$ and $\mu_{t,f} \in \mathbb{R}^{pd_f}$, respectively. This mirrors our definition of $x_t \in \mathbb{R}^{d_x}$ and $f_{1:p} \in \mathbb{R}^{pd_f}$ as the components of the full state $\tilde{x}_t := (x_t, f_{1:p}) \in \mathbb{R}^{d_x + pd_f}$. In addition, we will define the components of $\bar{\Sigma}_t^{-1/2} \in \mathbb{R}^{(d_x + pd_f) \times (d_x + pd_f)}$ and $\bar{\Sigma}_t^{-1} \in \mathbb{R}^{(d_x + pd_f) \times (d_x + pd_f)}$ by:

$$\begin{aligned} \begin{bmatrix} \Omega_{t,xx} & \Omega_{t,xf} \\ \Omega_{t,fx} & \Omega_{t,ff} \end{bmatrix} &:= \bar{\Sigma}_t^{-1} \in \mathbb{R}^{(d_x + pd_f) \times (d_x + pd_f)}, \\ \begin{bmatrix} \Lambda_{t,xx} & \Lambda_{t,xf} \\ \Lambda_{t,fx} & \Lambda_{t,ff} \end{bmatrix} &:= \bar{\Sigma}_t^{-1/2} \in \mathbb{R}^{(d_x + pd_f) \times (d_x + pd_f)}, \end{aligned}$$

where $\Sigma_{t,xx}, \Lambda_{t,xx} \in \mathbb{R}^{d_x \times d_x}$, $\Sigma_{t,xf}, \Lambda_{t,xf} \in \mathbb{R}^{d_x \times pd_f}$, $\Sigma_{t,fx}, \Lambda_{t,fx} \in \mathbb{R}^{pd_f \times d_x}$, and $\Sigma_{t,ff}, \Lambda_{t,ff} \in \mathbb{R}^{pd_f \times pd_f}$. Using the above definitions, we can reorder the residuals in $C_K \in \mathbb{R}$ and $C_M \in \mathbb{R}^{2d_x + pd_f}$, and thus redefine them by:

$$\begin{aligned} C_K(\tilde{x}_{t,K}) &= 0 \in \mathbb{R} \\ C_M(\tilde{x}_{t,K}, \tilde{x}_{t,M}) &= \begin{bmatrix} \Lambda_{t,xx}(x_t - \mu_{t,x}) + \Lambda_{t,xf}(f_{1:p} - \mu_{t,f}) \\ \Sigma_w^{-1/2}(x_{t+1} - g(x_t)) \\ \Lambda_{t,fx}(x_t - \mu_{t,x}) + \Lambda_{t,ff}(f_{1:p} - \mu_{t,f}) \end{bmatrix} \in \mathbb{R}^{2d_x + pd_f}. \end{aligned}$$

Our state variables and cost functions for the Gauss-Newton algorithm submodule are:

$$\begin{aligned} \bar{x}_M^* &= \tilde{x}_t^* = \bar{\mu}_t \in \mathbb{R}^{d_x + pd_f}, \\ \bar{x}_K^* &= g(\tilde{x}_t^*) = g(\bar{\mu}_t) \in \mathbb{R}^{d_x + pd_f}, \\ C_K(\tilde{x}_{t,K}^*) &= 0 \in \mathbb{R}, \\ C_M(\tilde{x}_{t,K}^*, \tilde{x}_{t,M}^*) &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{2d_x + pd_f}, \\ J_K &= \begin{bmatrix} O & \Lambda_{xf} \\ \Sigma_w^{-1/2} & O \\ O & \Lambda_{ff} \end{bmatrix} \in \mathbb{R}^{(2d_x + pd_f) \times (d_x + pd_f)} \end{aligned}$$

$$J_M = \begin{bmatrix} \Lambda_{xx} \\ -\Sigma_w^{-1/2} G_t \\ \Lambda_{xf} \end{bmatrix} \in \mathbb{R}^{(2d_x + pd_f) \times d_x},$$

where we have defined G_t to be the Jacobian of $g : \mathbb{R}^{d_x} \rightarrow \mathbb{R}^{d_x}$ at $\overline{\mu_{t,x}} \in \mathbb{R}^{d_x}$, i.e.:

$$G_t := \left. \frac{\partial g}{\partial x_t} \right|_{x_t = \overline{\mu_{t,x}}}$$

Applying the Marginalization equations, we thus have:

$$\begin{aligned} \mu_{t+1} &\leftarrow \tilde{x}_{t,K} - \Sigma_{t+1} J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] C_M (\overline{x_K^*}, \overline{x_M^*}) \\ &= g(\overline{\mu_t}), \\ \Sigma_{t+1} &\leftarrow (J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] J_K)^{-1}, \\ &= (J_K^\top J_K - J_K^\top J_M (J_M^\top J_M)^{-1} J_M^\top J_K)^{-1}, \\ &= \left(\begin{bmatrix} \Sigma_w^{-1} & O \\ O & \Lambda_{fx} \Lambda_{xf} + \Lambda_{ff}^2 \end{bmatrix} - \begin{bmatrix} -\Sigma_w^{-1} G_t \\ \Lambda_{fx} \Lambda_{xx} + \Lambda_{ff} \Lambda_{fx} \end{bmatrix} (\Lambda_{xx}^2 + \Lambda_{xf} \Lambda_{fx} + G_t^\top \Sigma_w^{-1} G_t)^{-1} \right. \\ &\quad \left. \cdot [-G_t^\top \Sigma_w^{-1} \quad \Lambda_{xx} \Lambda_{xf} + \Lambda_{fx} \Lambda_{ff}] \right)^{-1} \\ &= \left(\begin{bmatrix} \Sigma_w^{-1} & O \\ O & \Omega_{ff} \end{bmatrix} - \begin{bmatrix} -\Sigma_w^{-1} G_t \\ \Omega_{fx} \end{bmatrix} (\Omega_{xx} + G_t^\top \Sigma_w^{-1} G_t)^{-1} [-G_t^\top \Sigma_w^{-1} \quad \Omega_{xf}] \right)^{-1} \end{aligned}$$

To show that this is indeed identical to the propagation equation for the covariance matrix in the Extended Kalman Filter algorithm, i.e. Algorithm 3, Line 5, we must show that:

$$\begin{aligned} &\left(\begin{bmatrix} \Sigma_w^{-1} & O \\ O & \Omega_{ff} \end{bmatrix} - \begin{bmatrix} -\Sigma_w^{-1} G_t \\ \Omega_{fx} \end{bmatrix} (\Omega_{xx} + G_t^\top \Sigma_w^{-1} G_t)^{-1} [-G_t^\top \Sigma_w^{-1} \quad \Omega_{xf}] \right)^{-1} \\ &\quad \cdot \begin{bmatrix} G_t \overline{\Sigma}_{t,xx} G_t^\top + \Sigma_w & G_t \overline{\Sigma}_{t,xf} \\ \overline{\Sigma}_{t,xf} G_t^\top & \overline{\Sigma}_{t,ff} \end{bmatrix} \end{aligned}$$

This follows by brute-force expanding the above block matrix components, and applying Woodbury's Matrix Identity, along with the definitions of $\Sigma_{t,xx}$, $\Lambda_{t,xx}$, $\Sigma_{t,xf}$, $\Lambda_{t,xf}$, $\Sigma_{t,fx}$, $\Lambda_{t,fx}$, $\Sigma_{t,ff}$, and $\Lambda_{t,ff}$. \square

Remark 3.2.1. *In practice, Gauss-Newton steps for feature augmentation can be delayed and done with feature updates.*

Next, we present the original formulation of MSCKF SLAM, as portrayed in [79].

Algorithm 4: EKF SLAM, as an iterative optimization problem.

Data: Prior $\mathcal{N}(\mu_0, \Sigma_0)$ on $x_0 \in \mathbb{R}^{d_x}$, noise covariances Σ_w, Σ_v , dynamics map g , measurement map h , time horizon T .

Result: Estimates $\hat{x}_t \in \mathbb{R}^{d_x}, \forall t \in \{1, \dots, T\}$.

```

1  $f_0(x) \leftarrow \|x_0 - \mu_0\|_{\Sigma_0^{-1}}^2$ 
2  $p \leftarrow 0$ .
3 for  $t = 0, 1, \dots, T$  do
4    $(z_{t,1}, \dots, z_{t,p}) \leftarrow$  Measurements of existing features.
5    $\text{cost}_t \leftarrow \text{cost}_t + \sum_{k=1}^p \|z_{t,k} - h(x_t, f_k)\|_{\Sigma_v^{-1}}^2$ 
6    $\bar{\mu}_t, \bar{\Sigma}_t, \text{cost}_t \leftarrow$  1 Gauss-Newton step on  $\text{cost}_t$ , about  $\mu_t$ , (Alg. 1).
7    $\hat{x}_t \leftarrow \bar{\mu}_t \in \mathbb{R}^{d_x + p d_f}$ .
8    $(z_{t,p+1}, \dots, z_{t,p+p'}) \leftarrow$  Measurements of new features.
9    $\text{cost}_t \leftarrow \text{cost}_t + \sum_{k=p+1}^{p+p'} \|z_{t,k} - h(x_t, f_k)\|_{\Sigma_v^{-1}}^2$ 
10   $\bar{\mu}_t \leftarrow (\bar{\mu}_t, \ell(x_t, z_{t,p+1}), \dots, \ell(x_t, z_{t,p+p'})) \in \mathbb{R}^{d_x + (p+p')d_f}$ .
11   $\bar{\mu}_t, \bar{\Sigma}_t, \text{cost}_t \leftarrow$  1 Gauss-Newton step on  $\text{cost}_t$ , about  $\bar{\mu}_t$  (Alg. 1).
12   $p \leftarrow p + p'$ 
13  if  $t < T$  then
14     $\text{cost}_t \leftarrow \text{cost}_t + \|x_{t+1} - g(x_t)\|_{\Sigma_w^{-1}}^2$ 
15     $\mu_{t+1}, \Sigma_{t+1}, \text{cost}_t \leftarrow$  1 Marginalization step on  $\text{cost}_{t+1}$  with  $x_M = x_t$ , about
       $(\bar{\mu}_t, g(\bar{\mu}_t))$  (Alg. 2).
16     $\text{cost}_{t+1} \leftarrow \|x_{t+1} - \mu_{t+1}\|_{\Sigma_{t+1}^{-1}}^2$ 
17  end
18 end
19 return  $\hat{x}_0, \dots, \hat{x}_T$ 

```

3.3 Multi-State Constrained Kalman Filter (MSCKF), Standard Formulation

The Multi-State Constrained Kalman Filter (MSCKF) algorithm iteratively refines the mean and covariance of a MSCKF full state, consisting of the most recent IMU state and a sliding window of n poses (Algorithm 8). In particular, when a set of new IMU measurements is obtained, the MSCKF full state is propagated forward in time. When a new image measurement arrives, the current pose is appended to the MSCKF full state vector. Features not observed in the current pose are marginalized. If the number of poses maintained in the MSCKF full state, denoted n , exceeds a pre-specified upper bound N_{\max} , then features common to every third currently maintained pose, starting from the second oldest pose, are marginalized. Below, we discuss the key components of the MSCKF algorithm—the IMU state, poses, MSCKF full states, features, image measurements, dynamics, pose augmenta-

Algorithm 5: Extended Kalman Filter, Feature Augmentation Sub-block.

Data: Current EKF state $\tilde{x}_t \in \mathbb{R}^{d_x+pd_f}$, with mean $\mu_t \in \mathbb{R}^{d_x+pd_f}$ and covariance $\Sigma_t \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$, current number of features p , observations of new features at current pose $z_{t,p+1:p+p'} := (z_{t,p+1}, \dots, z_{t,p+p'}) \in \mathbb{R}^{p'd_z}$, measurement map $h : \mathbb{R}^{d_x} \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_z}$, inverse measurement map $\ell : \mathbb{R}^{d_x} \times \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_f}$.

Result: Updated number of features p , updated EKF state mean $\mu_t \in \mathbb{R}^{d_x+pd_f}$, covariance $\Sigma_t \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$ (with p already updated)

- 1 $(\mu_{t,x}, \mu_{t,f,1:p}) \leftarrow \mu_t \in \mathbb{R}^{d_x+pd_f}$, with $\mu_{t,x} \in \mathbb{R}^{d_x}$, $\mu_{t,f,1:p} \in \mathbb{R}^{pd_f}$.
- 2 $\ell : \mathbb{R}^{d_x} \times \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_f} \leftarrow$ Inverse measurement map, satisfying $z_{t,k} = h(x_t, \ell(x_t, z_{t,k}))$ for each $x_t \in \mathbb{R}^{d_x}$, $z_{t,k} \in \mathbb{R}^{d_z}$, $\forall k = p+1, \dots, p+p'$.
- 3 $\tilde{\ell}(\mu_{t,x}, z_{t,p+1}, \dots, z_{t,p+p'}) \leftarrow (\ell(\mu_{t,x}, z_{t,p+1}), \dots, \ell(\mu_{t,x}, z_{t,p+p'})) \in \mathbb{R}^{p'd_f \times (d_x+p'd_z)}$
- 4 $\mu_t \leftarrow (\mu_t, \tilde{\ell}(\mu_{t,x}, z_{t,p+1}, \dots, z_{t,p+p'})) \in \mathbb{R}^{d_x+(p+p')d_f}$
- 5 $\begin{bmatrix} \Sigma_{t,xx} & \Sigma_{t,xf} \\ \Sigma_{t,fx} & \Sigma_{t,ff} \end{bmatrix} \leftarrow \Sigma_t \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$, with $\Sigma_{t,xx} \in \mathbb{R}^{d_x \times d_x}$, $\Sigma_{t,xf} = \Sigma_{t,fx}^\top \in \mathbb{R}^{d_x \times pd_f}$, $\Sigma_{t,ff} \in \mathbb{R}^{pd_f \times pd_f}$.
- 6 $L_x \leftarrow \frac{\partial \tilde{\ell}}{\partial x} \Big|_{(\mu_t, z'_t)} \in \mathbb{R}^{p'd_f \times d_x}$
- 7 $L_z \leftarrow \frac{\partial \tilde{\ell}}{\partial z} \Big|_{(\mu_t, z'_t)} \in \mathbb{R}^{p'd_f \times p'd_z}$
- 8 $\tilde{\Sigma}_v \leftarrow \text{diag}\{\Sigma_v, \dots, \Sigma_v\} \in \mathbb{R}^{p'd_z \times p'd_z}$
- 9 $\Sigma_t \leftarrow \begin{bmatrix} \Sigma_{t,xx} & \Sigma_{t,xf} & \Sigma_{t,xx} L_x^\top \\ \Sigma_{t,fx} & \Sigma_{t,ff} & \Sigma_{t,fx} L_x^\top \\ L_x \Sigma_{t,xx} & L_x \Sigma_{t,xf} & L_x \Sigma_{t,xx} L_x^\top + L_z \tilde{\Sigma}_v L_z^\top \end{bmatrix} \in \mathbb{R}^{(d_x+(p+p')d_f) \times (d_x+(p+p')d_f)}$
- 10 $p \leftarrow p + p'$
- 11 **return** $\mu_t \in \mathbb{R}^{d_x+pd_f}$, $\Sigma_t \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$, $p \geq 0$

tion and measurement maps—in more detail.

The IMU state $x_{t,\text{IMU}}$ takes the form:

$$x_{t,\text{IMU}} := (q_{WS}, v_S, b_g, b_a, r_{WS})(t) \in \mathbb{R}^3 \times \mathbb{H}_u \times \mathbb{R}^9, \quad (3.3)$$

where we use S and W to represent the sensor frame and world frame, respectively. Here, $r_{WS} \in \mathbb{R}^3$ denotes the position of the IMU sensor frame represented in the world frame, $q_{WS} \in \mathbb{H}_u$ denotes the unit quaternion of axis rotation from world frame to IMU sensor frame, and $R(q_{WS}) \in SO(3)$ denotes the rotation matrix associated with q_{WS} . Moreover, $v_S \in \mathbb{R}^3$ denotes the linear velocity of the IMU sensor frame relative to the world frame, as represented in the world frame, while $b_g \in \mathbb{R}^3$ and $b_a \in \mathbb{R}^3$ denote the sensor biases of the gyroscope and accelerometer, respectively. Finally, $\tilde{\omega}_S \in \mathbb{R}^3$ and $\tilde{a}_S \in \mathbb{R}^3$ denote gyroscope and accelerometer measurements, respectively.

For convenience, define $\mathcal{X}_{\text{IMU}} := \mathbb{R}^3 \times \mathbb{H}_u \times \mathbb{R}^9$ and $\mathcal{X}'_{\text{IMU}} := \mathbb{R}^3 \times \mathbb{H} \times \mathbb{R}^9$. The continuous-

Algorithm 6: Extended Kalman Filter, Feature Update Sub-block.

Data: Current EKF state $\tilde{x}_t \in \mathbb{R}^{d_x+pd_f}$, with mean $\mu_t \in \mathbb{R}^{d_x+pd_f}$ and covariance $\Sigma_t \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$, new measurements of existing features

$z_{t,1:p} := (z_{t,1}, \dots, z_{t,p}) \in \mathbb{R}^{pd_z}$, measurement map $h : \mathbb{R}^{d_x} \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_z}$

Result: Updated EKF state mean $\mu_t \in \mathbb{R}^{d_x+pd_f}$ and covariance

$\Sigma_t \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$

- 1 $f_{1:p} \leftarrow (f_1, \dots, f_p) \in \mathbb{R}^{pd_f}$.
- 2 $\tilde{h}(x_t, f_{1:p}) \leftarrow (h(x_t, f_1), \dots, h(x_t, f_p)) \in \mathbb{R}^{pd_z}$
- 3 $H_t \leftarrow \frac{\partial \tilde{h}}{\partial (x_t, f_{1:p})} \Big|_{\mu_t}$ Jacobian of $\tilde{h} : \mathbb{R}^{d_x} \times \mathbb{R}^{pd_f} \rightarrow \mathbb{R}^{pd_z}$ evaluated at $\mu_t \in \mathbb{R}^{d_x+pd_f}$.
- 4 $\tilde{\Sigma}_v \leftarrow \text{diag}\{\Sigma_v, \dots, \Sigma_v\} \in \mathbb{R}^{pd_z \times pd_z}$.
- 5 $\bar{\mu}_t \leftarrow \mu_t + \Sigma_t H_t^T (H_t \Sigma_t H_t^T + \tilde{\Sigma}_v)^{-1} (z_{t,1:p} - \tilde{h}(\mu_t)) \in \mathbb{R}^{d_x+pd_f}$.
- 6 $\bar{\Sigma}_t \leftarrow \Sigma_t - \Sigma_t H_t^T (H_t \Sigma_t H_t^T + \tilde{\Sigma}_v)^{-1} H_t \Sigma_t \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$.
- 7 **return** $\bar{\mu}_t \in \mathbb{R}^{d_x+pd_f}$, $\bar{\Sigma}_t \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$.

Algorithm 7: Extended Kalman Filter, State Propagation Sub-block.

Data: Current EKF state $\tilde{x}_t \in \mathbb{R}^{d_x+pd_f}$, with mean $\bar{\mu}_t \in \mathbb{R}^{d_x+pd_f}$ and covariance $\bar{\Sigma}_t \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$, (discrete-time) dynamics map $g : \mathbb{R}^{d_x} \rightarrow \mathbb{R}^{d_x}$

Result: Propagated EKF state mean $\mu_{t+1} \in \mathbb{R}^{d_x+pd_f}$ and covariance

$\Sigma_{t+1} \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$

- 1 $(\bar{\mu}_{t,x}, \bar{\mu}_{t,f,1:p}) \leftarrow \bar{\mu}_t$, with $\bar{\mu}_{t,x} \in \mathbb{R}^{d_x}$, $\bar{\mu}_{t,f,1:p} \in \mathbb{R}^{pd_f}$.
- 2 $\begin{bmatrix} \bar{\Sigma}_{t,xx} & \bar{\Sigma}_{t,xf} \\ \bar{\Sigma}_{t,fx} & \bar{\Sigma}_{t,ff} \end{bmatrix} \leftarrow \bar{\Sigma}_t \in \mathbb{R}^{d_x \times d_x}$, with $\bar{\Sigma}_{t,xx} \in \mathbb{R}^{d_x \times d_x}$, $\bar{\Sigma}_{t,xf} = \bar{\Sigma}_{t,fx}^\top \in \mathbb{R}^{d_x \times pd_f}$, $\bar{\Sigma}_{t,ff} \in \mathbb{R}^{pd_f \times pd_f}$.
- 3 $G_t \leftarrow \frac{\partial g}{\partial x} \Big|_{\bar{\mu}_{t,x}}$.
- 4 $\mu_{t+1} \leftarrow (g(\bar{\mu}_{t,x}), \bar{\mu}_{t,f,1:p}) \in \mathbb{R}^{d_x+pd_f}$.
- 5 $\Sigma_{t+1} \leftarrow \begin{bmatrix} G_t \bar{\Sigma}_{t,xx} G_t^\top + \Sigma_w & G_t \bar{\Sigma}_{t,xf} \\ \bar{\Sigma}_{t,fx} G_t^\top & \bar{\Sigma}_{t,ff} \end{bmatrix} \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$.
- 6 **return** $\mu_{t+1} \in \mathbb{R}^{d_x+pd_f}$, $\Sigma_{t+1} \in \mathbb{R}^{(d_x+pd_f) \times (d_x+pd_f)}$.

Algorithm 8: Multi-State Constrained Kalman Filter, Standard Formulation.

Data: Prior distribution on $x_{\text{IMU},0} \in \mathcal{X}_p$: $\mathcal{N}(\mu_0, \Sigma_0)$, dynamics and measurement noise covariances $\Sigma_w \in \mathbb{R}^{d_x \times d_x}$, $\Sigma_v \in \mathbb{R}^{d_x \times d_z}$, discrete-time dynamics map $g_{\text{IMU}} : \mathbb{R}^{d_{\text{IMU}}} \times \mathbb{R}^{d_{\text{IMU}}}$, measurement map $h : \mathcal{X}_p \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_z}$, time horizon T , pose transformation $\psi : \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n \times \mathcal{X}_p \rightarrow \mathcal{X}_p$ (IMU \rightarrow global).

Result: Estimates $\hat{x}_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ for all desired timesteps $t \leq T$, where $n :=$ number of poses in \hat{x}_t at time t .

```

1  $S_z, S_x, S_{z,1}, S_{z,2} \leftarrow \phi$ 
2  $(n, p) \leftarrow (0, 0)$ 
3 for  $t = 0, \dots, T$  do
4   while new image with new pose  $x_{n+1} \in \mathcal{X}_p$  recorded, new IMU measurement not
   yet received do
5      $\mu_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n, \Sigma_t \in \mathbb{R}^{(d_{\text{IMU}}+nd_x)^2} \leftarrow$  Alg. 9 ( $\tilde{x}_t, \mu_t, \Sigma_t, x_{n+1}, x_{n+1}^{\text{IMU}}, \psi(\cdot)$ )
6      $\{z_{n+1,j} \mid \text{feature } j \text{ is observed at } x_{n+1}\} \leftarrow$  Feature measurements at  $x_{n+1}$ 
7      $\{f_j^* \mid \text{Feature } j \text{ is observed at } x_{n+1}\} \leftarrow$  Feature position estimates at  $x_{n+1}$ .
8      $x_{n+1} \in \mathcal{X}_p \leftarrow$  Record new estimates of existing features, first estimate of new
     features.
9      $S_z \leftarrow S_z \cup \{(x_{n+1}, f_j) \mid \text{Feature } j \text{ observed at } n+1\}$ 
10     $n \leftarrow n+1$ 
11    if  $n \geq N_{\text{max}} - 1$  then
12       $S_x \leftarrow \{x_i \mid i \bmod 3 = 2, \text{ and } 1 \leq i \leq n.\}$ 
13       $S_{z,1} \leftarrow \{(x_i, f_j) \in S_z \mid x_i \in S_x, \text{ feature } j \text{ observed at each pose in } S_x\}$ 
14    end
15     $S_{z,2} \leftarrow \{(x_i, f_j) \in S_z \mid x_i \in x_{1:n}, \text{ feature } j \text{ observed at } x_i \text{ but not at } x_n.\}$ 
16     $S_f \leftarrow \{f_j \mid \exists x_i \in x_{1:n} \text{ s.t. } (x_i, f_j) \in S_{z,1} \cup S_{z,2}\}$ 
17    if  $S_f \neq \phi$  then
18       $\bar{\mu}_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n, \bar{\Sigma}_t \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)} \leftarrow$  Alg. 10 ( $\tilde{x}_t, \mu_t, \Sigma_t,$ 
       $x_{n+1}, S_{z,1} \cup S_{z,2}, S_f, h(\cdot)$ )
19       $\hat{x}_t \leftarrow \bar{\mu}_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ .
20    end
21     $S_z \leftarrow S_z \setminus (S_{z,1} \cup \{(x_i, f_j) \mid x_i \in S_x\})$ 
22    Reindex poses and features, i.e.,  $\{x_1, \dots, x_{n-|S_x|}\}$  and  $\{f_1, \dots, f_{p-|S_f|}\}$ .
23     $(p, n) \leftarrow (p - |S_f|, n - |S_x|)$ 
24  end
25  if  $t < T$  then
26     $\mu_{t+1} \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n, \Sigma_{t+1} \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)} \leftarrow$  Alg. 11, MSCKF State
    Propagation ( $\tilde{x}_t, \bar{\mu}_t, \bar{\Sigma}_t$ )
27  end
28 end
29 return  $\hat{x}_0, \dots, \hat{x}_T \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ 

```

time IMU dynamics map $g_{\text{IMU},ct} : \mathcal{X}_{\text{IMU}} \rightarrow \mathcal{X}'_{\text{IMU}}$ is given by:

$$\begin{aligned}\dot{q}_{WS} &= q_{WS} \star \frac{1}{2} \begin{bmatrix} 0 \\ \tilde{\omega}_S - b_g - w_g \end{bmatrix}, \\ \dot{b}_g &= w_{b_g}, \\ \dot{v}_S &= R(q_{WS})^\top (\tilde{a}_S - b_a + w_a) + g_W, \\ \dot{b}_a &= w_{b_a}, \\ \dot{r}_{WS} &= v_S.\end{aligned}$$

where \star denotes quaternion multiplication, and $w_g, w_a, w_{b_g}, w_{b_a} \in \mathbb{R}^3$ denote zero-mean standard Gaussian noise.

Each pose $x_k \in \{x_1, \dots, x_n\}$ currently maintained in the sliding window of poses takes the form $x_k := (q_{WC_k}, r_{WC_k}) \in \mathbb{H}_u \times \mathbb{R}^3$, where $r_{WC_k} \in \mathbb{R}^3$ denotes the position of the camera at pose x_k in the world frame, while $q_{WC_k} \in \mathbb{H}_u$ denotes the quaternion associated with the axis rotation from the world frame to the camera frame at pose $x_k \in \mathbb{H} \times \mathbb{R}^3$. For convenience, we define $\mathcal{X}_p := \mathbb{H}_u \times \mathbb{R}^3$.

The MSCKF full state maintained throughout the operation of the MSCKF algorithm contains the IMU state at the current time, as well as a collection of n poses, where n is constrained to remain below a pre-specified, fixed upper bound N_{\max} :

$$\tilde{x}_t := (x_{t,\text{IMU}}, x_1, \dots, x_n) \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n. \quad (3.4)$$

The state space is thus $\mathcal{X} := \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$.

When a new image measurement arrives, the estimate of the current camera pose in the IMU frame, denoted $x_{n+1}^{\text{IMU}} \in \mathcal{X}_{\text{IMU}}$, is transformed to the global frame, and appended to the MSCKF full state \tilde{x}_t . This coordinate transformation is realized by the map $\psi : \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n \rightarrow \mathcal{X}_p$, defined by:

$$\begin{aligned}& \psi(q_{WS}, v_S, b_g, b_a, r_{WS}, q_{WC_1}, r_{WC_1}, \dots, q_{WC_n}, r_{WC_n}, q_{WI_{n+1}}, r_{WI_{n+1}}) \\ &= (q_{IC} \star q_{WI_{n+1}}, r_{WI_{n+1}} + C(q_{WI_{n+1}})r_{IC}) \\ &:= (q_{WC_{n+1}}, r_{WC_{n+1}}),\end{aligned}$$

where q_{IC} denotes the quaternion encoding the (fixed) transformation from the IMU frame to the camera frame. In summary, the MSCKF algorithm defines the new pose x_{n+1} and updates the MSCKF full state \tilde{x}_t as follows:

$$\begin{aligned}x_{n+1} &\leftarrow \psi(\tilde{x}_t, x_{n+1}^{\text{IMU}}) \in \mathcal{X}_p, \\ \tilde{x}_t &\leftarrow (\tilde{x}_t, x_{n+1}) = (\tilde{x}_t, \psi(\tilde{x}_t, x_{n+1}^{\text{IMU}})) \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^{n+1},\end{aligned}$$

with the map ψ as defined above.

When new feature position estimates are detected from a new image measurement, the new camera pose corresponding to this image measurement is appended to \tilde{x}_t , and n is

incremented by 1. If $n = N_{\max}$ the upper limit N_{\max} , a third of all old poses in \tilde{x}_t is discarded, starting from the second oldest pose. Then, feature measurements, corresponding to features unobserved at the current pose, are marginalized and used to update the mean and covariance of the new MSCKF full state \tilde{x}_t .

As is the case with the EKF algorithm, we assume that the image measurement space and feature space are given by \mathbb{R}^{d_z} and \mathbb{R}^{d_f} , respectively, with $d_z = 2$ and $d_f = 3$. Throughout the duration of the MSCKF algorithm, poses and features are added into, dropped from, and marginalized from the MSCKF full state. Suppose at a given time, the MSCKF maintains n poses in the *MSCKF full state* \tilde{x}_t , and retains measurements of p features. For each pose $i \in \{1, \dots, n\}$ and feature $j \in \{1, \dots, p\}$ currently maintained in the SLAM algorithm, if feature j were detected at pose i , let $z_{i,j} \in \mathbb{R}^{d_z}$ denote the associated feature measurement. For the MSCKF, the measurement map $h : \mathcal{X}_{\text{IMU}} \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_z}$ is given by:

$$z_{i,j} = h(x_i, f_j) := \frac{1}{(R(q_{WC_k})f_j - r_{WC_k})_z} \begin{bmatrix} (R(q_{WC_k})f_j - r_{WC_k})_x \\ (R(q_{WC_k})f_j - r_{WC_k})_y \end{bmatrix} (t) + v_{i,j}.$$

where $R(q_{WC_k}) \in SO(3)$ denotes the rotation matrix associated with the quaternion q_{WC_k} , $f_j \in \mathbb{R}^3$ denotes the position of feature j in the world frame, while the subscript indices “ x, y, z ” refer to the respective coordinates of the vector $R(q_{WS})f_j - r_{WS} \in \mathcal{X}_p$. Meanwhile, $v_{i,j} \in \mathbb{R}^{d_z}$ denotes zero-mean standard Gaussian noise in the measurement at time t , with covariance matrix $\Sigma_v \in \mathbb{R}^{d_z \times d_z}$, $\Sigma_v \succ 0$.

When a new image measurement is received, the MSCKF algorithm performs marginalization, described in Section 2.6, using two sets of feature measurements—the set of all feature measurements common to old poses x_i to be dropped, denoted $S_{z,1}$, as well as the set of all feature measurements of features f_j not seen in the current pose, denoted $S_{z,2}$. These are more precisely defined in Section 3.4. The measurement vector used for marginalization, denoted $\tilde{z} \in \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z}$, is then given by concatenating the q residual measurements obtained at times $t - n + 1, \dots, t$, i.e.:

$$\tilde{z} := \{z_{i,j} | (x_i, f_j) \in S_{z,1} \cup S_{z,2}\} \in \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z}.$$

3.4 Multi-State Constrained Kalman Filter (MSCKF), on Manifolds

The MSCKF algorithm maintains a full state, $\tilde{x}_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$, containing the most recent IMU state, $x_{\text{IMU}} \in \mathcal{X}_{\text{IMU}}$ and n recent poses, $(x_1, \dots, x_n) \in (\mathcal{X}_p)^n$:

$$\tilde{x}_t := (x_{t,\text{IMU}}, x_1, \dots, x_n) \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n,$$

with mean $\mu_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ and covariance $\Sigma_t \in \mathbb{R}^{(d_{\text{IMU}} + nd_x) \times (d_{\text{IMU}} + nd_x)}$. As new poses are introduced, old poses are discarded, and features are processed and discarded to update \tilde{x}_t , the mean μ_t , covariance Σ_t , and $n \in \mathbb{N}$ accordingly.

Algorithm 9: Multi-State Constrained Kalman Filter, Pose Augmentation Sub-block.

Data: MSCKF state $\tilde{x}_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$, with mean $\mu_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ and covariance $\Sigma_t \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)}$, New pose $x_{n+1} \in \mathcal{X}_p$, measurement of new pose in IMU frame $x_{n+1}^{\text{IMU}} \in \mathcal{X}_p$, Transformation of poses from IMU frame to global frame $\psi : \mathbb{R}^{(d_{\text{IMU}}+nd_x)} \times \mathcal{X}_p \rightarrow \mathcal{X}_p$

Result: Updated MSCKF state mean $\mu_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ and covariance $\Sigma_t \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)}$, updated number of poses n .

- 1 $\tilde{x}_t \leftarrow (\tilde{x}_t, x_{n+1}) \in \mathbb{R}^{d_{\text{IMU}}+(n+1)d_x}$, where $x_{n+1} \in \mathcal{X}_p$ is the new pose vector.
 - 2 $\{z_{n+1,j} \mid \text{Feature } j \text{ is observed at pose } n+1\} \leftarrow$ Feature measurements at pose x_{n+1}
 - 3 $\{f_j^* \mid \text{Feature } j \text{ is observed at pose } x_{n+1}\} \leftarrow$ Feature position estimates at pose x_{n+1} .
 - 4 $\mu_t \leftarrow (\mu_t, \psi(\mu_t, x_{n+1}^{\text{IMU}})) \in \mathbb{R}^{d_{\text{IMU}}+(n+1)d_x}$, where $\mu_{t,\text{IMU}} \in \mathbb{R}^{d_{\text{IMU}}} :=$ IMU component of μ_t , $x_{n+1}^{\text{IMU}} \in \mathcal{X}_p :=$ pose estimate of x_{n+1} from the IMU frame.
 - 5 $\Sigma_t \leftarrow \begin{bmatrix} I_{d_{\text{IMU}}+(n+1)d_x} \\ \frac{\partial \psi}{\partial (\tilde{x}_t, x_{n+1}^{\text{IMU}})} \end{bmatrix} \Sigma_t \begin{bmatrix} I_{d_{\text{IMU}}+(n+1)d_x} \\ \frac{\partial \psi}{\partial (\tilde{x}_t, x_{n+1}^{\text{IMU}})} \end{bmatrix}^\top$
 - 6 **return** $\mu_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$, $\Sigma_t \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)}$, $n \geq 0$
-

At initialization ($t = 0$), no pose has yet been recorded ($n = 0$), and the full state \tilde{x}_0 is the initial IMU state $\tilde{x}_{0,\text{IMU}} \in \mathcal{X}_{\text{IMU}}$, with mean $\mu_0 \in \mathcal{X}_{\text{IMU}}$ and covariance $\Sigma_0 \in \mathbb{R}^{d_{\text{IMU}} \times d_{\text{IMU}}}$. Thus, $\tilde{x}_0 = \mu_0$ optimizes the initial running cost $c_{\text{MSCKF},0} : \mathcal{X}_{\text{IMU}} \rightarrow \mathbb{R}$ in our algorithm:

$$c_{\text{MSCKF},0}(\tilde{x}_0) = \|\tilde{x}_0 \boxminus \mu_0\|_{\Sigma_0^{-1}}^2.$$

Suppose that, at the current time t , the running cost $c_{\text{MSCKF},t,0} : \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n \rightarrow \mathbb{R}$ is:

$$c_{\text{MSCKF},t,0}(\tilde{x}_t) = \|\tilde{x}_t \boxminus \mu_t\|_{\Sigma_t^{-1}}^2,$$

where $\mu_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ and $\Sigma_t \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)}$ denote the mean and covariance of the full state $\tilde{x}_t := (x_{t,\text{IMU}}, x_1, \dots, x_n) \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ at time t , consisting of the current IMU state and n poses. When a new image is received, the *pose augmentation step* adds a new pose $x_{n+1} \in \mathcal{X}_p$ (global frame) to \tilde{x}_t , derived from $x_{n+1}^{\text{IMU}} \in \mathcal{X}_{\text{IMU}}$, the IMU position estimate in the global frame, via the map $\psi : \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n \times \mathcal{X}_{\text{IMU}} \rightarrow \mathcal{X}_p$, i.e.,

$$x_{n+1} := \psi(\tilde{x}_t, x_{n+1}^{\text{IMU}}) \in \mathcal{X}_p.$$

The *feature update step* uses features measurements to update the mean and covariance of \tilde{x}_t . In MSCKF, (A) if a feature becomes unobserved in the current pose, it is discarded, and (B) when $n \geq N_{\text{max}}$, a specified upper bound, features common to $\lfloor N_{\text{max}} \rfloor / 3$ of the n poses, evenly spaced in time, are processed via a feature update step, then dropped alongside the corresponding poses. Let $S_{z,1}$ and $S_{z,2}$ denote sets of pose-feature pairs (x_i, f_j) from cases (A)

Algorithm 10: Multi-State Constrained Kalman Filter, Feature Update Sub-block.

Data: MSCKF state $\tilde{x}_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$, with mean $\mu_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ and covariance $\Sigma_t \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)}$, Set of image measurements for marginalization $S_{z,1} \cup S_{z,2}$, Set of features to marginalize S_f , measurement map $h : \mathcal{X}_p \times \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_z}$.

Result: Updated MSCKF state mean $\bar{\mu}_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ and covariance $\bar{\Sigma}_t \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)}$.

- 1 $f_{S_f} \in \mathbb{R}^{|S_f|d_f} \leftarrow$ Concatenation of all features in S_f
- 2 $f_{S_f}^* \in \mathbb{R}^{|S_f|d_f} \leftarrow$ Concatenation of position estimate of all features in S_f
- 3 $\tilde{h}(\tilde{x}_t, f_{S_f}) \in \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z} \leftarrow$ Concatenation of measurement map outputs $\{h(x_i, f_j) | (x_i, f_j) \in S_{z,1} \cup S_{z,2}\}$.
- 4 $\tilde{z} \in \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z} \leftarrow$ Concatenation of feature measurements $\{z_{ij} | (x_i, f_j) \in S_{z,1} \cup S_{z,2}\}$.
- 5 $\tilde{H}_{t,x} \leftarrow \frac{\partial \tilde{h}}{\partial \tilde{x}_t}(\mu_t, f_{S_f}^*) \in \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z \times (d_{\text{IMU}}+nd_x)}$.
- 6 $\tilde{H}_{t,f} \leftarrow \frac{\partial \tilde{h}}{\partial f_{S_f}}(\mu_t, f_{S_f}^*) \in \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z \times |S_f|d_f}$.
- 7 $\{a_1, \dots, a_{|S_{z,1} \cup S_{z,2}|d_z - |S_f|d_f}\} \subset \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z} \leftarrow$ Orthonormal basis for $N(\tilde{H}_{t,f}^\top)$.
- 8 $A \leftarrow [a_1 \ \dots \ a_{|S_{z,1} \cup S_{z,2}|d_z - |S_f|d_f}] \in \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z \times (|S_{z,1} \cup S_{z,2}|d_z - |S_f|d_f)}$.
- 9 $QT \leftarrow$ QR Decomposition of $A^\top \tilde{H}_{t,x}$, with $Q \in \mathbb{R}^{(|S_{z,1} \cup S_{z,2}|d_z - |S_f|d_f) \times (|S_{z,1} \cup S_{z,2}|d_z - |S_f|d_f)}$, $T \in \mathbb{R}^{(|S_{z,1} \cup S_{z,2}|d_z - |S_f|d_f) \times (d_{\text{IMU}}+nd_x)}$.
- 10 $\bar{\Sigma}_t^{-1} \leftarrow \Sigma_t^{-1} + T^\top (Q^\top A^\top R A Q)^{-1} T \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)}$.
- 11 $\bar{\mu}_t \leftarrow \mu_t \boxplus (\Sigma_t^{-1} + T^\top (Q^\top A^\top R A Q)^{-1} T)^{-1} T^\top (Q^\top A^\top R A Q)^{-1} (\tilde{z} \boxminus \tilde{h}(\tilde{x}_t)) \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$.
- 12 $\hat{x}_t \leftarrow \bar{\mu}_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$.
- 13 **return** $\bar{\mu}_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$, $\bar{\Sigma}_t \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)}$

Algorithm 11: Multi-State Constrained Kalman Filter, State Propagation Sub-block.

Data: MSCKF state $\tilde{x}_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$, with mean $\mu_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ and covariance $\Sigma_t \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)}$, (discrete-time) dynamics map $g : \mathbb{R}^{d_{\text{IMU}}} \rightarrow \mathbb{R}^{d_{\text{IMU}}}$.

Result: Updated MSCKF state mean $\mu_{t+1} \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ and covariance $\Sigma_{t+1} \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)}$.

- 1 $(\bar{\mu}_{t,\text{IMU}}, \bar{\mu}_{t,x,1:n}) \leftarrow \bar{\mu}_t$, with $\bar{\mu}_{t,\text{IMU}} \in \mathbb{R}^{d_{\text{IMU}}}$, $\bar{\mu}_{t,x,1:n} \in \mathbb{R}^{nd_x}$.
- 2 $G_t \leftarrow$ Jacobian of $g_{\text{IMU}} : \mathbb{R}^{d_{\text{IMU}}} \rightarrow \mathbb{R}^{d_{\text{IMU}}}$ evaluated at $\bar{\mu}_{t,\text{IMU}} \in \mathbb{R}^{d_{\text{IMU}}}$.
- 3 $\mu_{t+1} \leftarrow (g_{\text{IMU}}(\bar{\mu}_{t,\text{IMU}}), \bar{\mu}_{t,x,1:n}) \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$.
- 4 $\Sigma_{t+1} \leftarrow \begin{bmatrix} G_t & O \\ O & I_{nd_x} \end{bmatrix} \bar{\Sigma}_t \begin{bmatrix} G_t^\top & O \\ O & I_{nd_x} \end{bmatrix} + \begin{bmatrix} \Sigma_w & O \\ O & O \end{bmatrix} \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)}$.
- 5 **return** $\mu_{t+1} \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$, $\Sigma_{t+1} \in \mathbb{R}^{(d_{\text{IMU}}+nd_x) \times (d_{\text{IMU}}+nd_x)}$.

and (B) above, respectively, and let S_f denote the set of features to be processed and (Alg. 8). These constraints are then incorporated, creating a new cost $c_{MSCKF,t,2} : \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n \rightarrow \mathbb{R}$:

$$c_{MSCKF,t,2}(\tilde{x}_t) := \|\tilde{x}_t \boxminus \mu_t\|_{\Sigma_t^{-1}}^2 + \sum_{(x_i, f_j) \in S_{z,1} \cup S_{z,2}} \|z_{i,j} \boxminus h(x_i, f_j)\|_{\Sigma_v^{-1}}^2,$$

where $z_{i,j} \in \mathbb{R}^{d_z}$ is the feature measurement of feature j observed from pose $x_i \in \mathcal{X}_p$. Using Gauss-Newton steps, we use constraints posed by measurement residuals to create an updated mean for \tilde{x}_t , denoted $\bar{\mu}_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$, and an updated covariance for \tilde{x}_t , denoted $\bar{\Sigma}_t \in \mathbb{R}^{(d_{\text{IMU}} + nd_x) \times (d_{\text{IMU}} + nd_x)}$. Then, we update our cost to $c_{MSCKF,t,3} : \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n \rightarrow \mathbb{R}$:

$$c_{MSCKF,t,3}(\tilde{x}_t) := \|\tilde{x}_t \boxminus \bar{\mu}_t\|_{\bar{\Sigma}_t^{-1}}^2,$$

which assumes the form of $c_{MSCKF,t,0}$.

The *state propagation* step propagates the full state by assimilating dynamics residuals, creating a new cost $c_{MSCKF,t,4} : (\mathcal{X}_{\text{IMU}})^2 \times (\mathcal{X}_p)^n \rightarrow \mathbb{R}$:

$$c_{MSCKF,t,4}(\tilde{x}_t, x_{t+1, \text{IMU}}) := \|\tilde{x}_t \boxminus \bar{\mu}_t\|_{\bar{\Sigma}_t^{-1}}^2 + \|x_{t+1, \text{IMU}} \boxminus g_{\text{IMU}}(x_t, \text{IMU})\|_{\Sigma_t^{-1}}^2.$$

In effect, $c_{MSCKF,t,4}$ appends the new IMU variable $x_{t+1, \text{IMU}} \in \mathcal{X}_{\text{IMU}}$ to the current full state $\tilde{x}_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$, and constrains this new full state via the dynamics residuals. A marginalization step, with $\tilde{x}_{t,K} := (x_{t+1, \text{IMU}}, x_1, \dots, x_n) \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ and $\tilde{x}_{t,M} := x_{t, \text{IMU}} \in \mathcal{X}_{\text{IMU}}$, then removes the previous IMU state, $x_{t, \text{IMU}}$, from the running cost. This produces a mean $\mu_{t+1} \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ and a covariance $\Sigma_{t+1} \in \mathbb{R}^{(d_{\text{IMU}} + nd_x) \times (d_{\text{IMU}} + nd_x)}$ for the new MSCKF full state, $\tilde{x}_{t+1} := \tilde{x}_{t,K} = (x_{t+1, \text{IMU}}, x_1, \dots, x_n) \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$. The running cost is updated to $c_{MSCKF,t+1,0} : \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n \rightarrow \mathbb{R}$:

$$c_{MSCKF,t+1,0}(\tilde{x}_{t+1}) := \|\tilde{x}_{t+1} \boxminus \mu_{t+1}\|_{\Sigma_{t+1}^{-1}}^2,$$

which returns the running cost to the form of $c_{MSCKF,t,0}$.

The theorems below establish that the feature augmentation, feature update, and state propagation steps of the MSCKF, presented above in our optimization framework, correspond precisely to those presented in the standard MSCKF (Alg. 8) [79].

Theorem 3.4.1. *The pose augmentation step of the standard MSCKF (Alg. 9) is equivalent to applying a Gauss-Newton step to $c_{MSCKF,t,1} : \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n \times \mathcal{X}_{\text{IMU}} \rightarrow \mathbb{R}$, with:*

$$c_{MSCKF,t,1}(\tilde{x}_t, x_{n+1}) = \|\tilde{x}_t \boxminus \mu_t\|_{\Sigma_t^{-1}}^2 + \epsilon^{-1} \|x_{n+1} \boxminus \psi(\tilde{x}_t, x_{n+1}^{\text{IMU}})\|_2^2,$$

and taking $\epsilon \rightarrow 0$ in the resulting (augmented) mean μ_t and covariance Σ_t .

Proof. We claim that from an optimization perspective, the pose augmentation step is equivalent to applying one Gauss-Newton step to the cost function $c_{MSCKF,t,1}(\tilde{x}_t, x_{n+1})$, specified above, and then taking the limit $\epsilon \rightarrow 0$ in the resulting augmented mean $\mu_t(\epsilon) \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^{(n+1)}$ and augmented covariance $\mu_t(\epsilon) \in \mathbb{R}^{(d_{\text{IMU}} + (n+1)d_x) \times (d_{\text{IMU}} + (n+1)d_x)}$.

Algorithm 12: Multi-State Constrained Kalman Filter, as iterative optimization.

Data: Prior $\mathcal{N}(\mu_0, \Sigma_0)$ on $x_{\text{IMU},0} \in \mathcal{X}_{\text{IMU}}$, noise covariances Σ_w, Σ_v , dynamics g_{IMU} , measurement map h , time horizon T , Pose transform ψ (IMU \rightarrow global), $\epsilon > 0$.

Result: Estimates \hat{x}_t for all desired timesteps $t \in \{1, \dots, T\}$.

```

1 costt ← ||x0 ⊖ μ0||Σ02. (Initialize objective function).
2 Sz, Sx, Sz,1, Sz,2 ← ϕ
3 (n, p) ← (0, 0)
4 for t = 0, ⋯, T do
5   while new pose xn+1 ∈ Xp recorded, new IMU measurement not received do
6     costt ← costt + ε-1||xn+1 ⊖ ψ(x̃t, xn+1IMU)||22.
7     μt, Σt, costt ← 1 Gauss-Newton costt (Alg. 1), about (μt, ψ(μt, xn+1IMU)) with
        ε → 0.
8     {zn+1,j} ← Feature measurements at xn+1
9     Sz ← Sz ∪ {(xn+1, fj) | fj observed at n + 1}
10    n ← n + 1
11    if n ≥ Nmax - 1 then
12      Sx ← {xi | i mod 3 = 2, and 1 ≤ i ≤ n.}
13      Sz,1 ← {(xi, fj) ∈ Sz | xi ∈ Sx, feature j observed at each pose in Sx}
14    end
15    Sz,2 ← {(xi, fj) ∈ Sz | fj not observed at xn}.
16    costt ← costt + ∑(xi, fj) ∈ Sz,1 ∪ Sz,2 ||zi,j ⊖ h(xi, ft,j)||Σv-1
17    μ̄t, Σ̄t, costt ← 1 Gauss-Newton step on costt, about μt (Alg. 1)
18    x̂t ← μ̄t ∈ XIMU × (Xp)n.
19    Sz ← Sz \ (Sz,1 ∪ {(xi, fj) | xi ∈ Sx})
20    Reindex poses and features in ascending order.
21    (p, n) ← (p - |Sf|, n - |Sx|)
22  end
23  if t < T then
24    costt ← costt + ||xt+1,IMU ⊖ gIMU(xt,IMU)||Σw-12.
25    μt+1, Σt+1, costt ← 1 Marginalization step on costt, about (μ̄t, g(μt,IMU)) (Alg.
        2)
26  end
27 end
28 return x̂0, ⋯, x̂T ∈ XIMU × (Xp)n

```

To apply a Gauss-Newton step, our first task is to find a vector $C(\tilde{x}_t, x_{n+1})$ of an appropriate dimension such that $c_{MSCKF,t,1}(\tilde{x}_t, x_{n+1}) = C_1(\tilde{x}_t, x_{n+1})^\top C_1(\tilde{x}_t, x_{n+1})$. A natural choice is furnished by $C_1(\tilde{x}_t, x_{n+1}) \in \mathbb{R}^{d_{\text{IMU}}+(n+1)d_x}$, as defined below:

$$C_1(\tilde{x}_t, x_{n+1}) := \begin{bmatrix} \Sigma_t^{-1/2}(\tilde{x}_t \boxminus \mu_t) \\ \epsilon^{-1/2}(x_{n+1} \boxminus \psi(\tilde{x}_t, x_{n+1}^{\text{IMU}})) \end{bmatrix}.$$

Thus, our parameters for the Gauss-Newton algorithm submodule are:

$$\begin{aligned} (\tilde{x}_t^*, x_{n+1}^*) &:= (\mu_t, \psi(\mu_t, x_{n+1}^{\text{IMU}})) \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n, \\ C_1(\tilde{x}_t^*, x_{n+1}^*) &= \begin{bmatrix} \Sigma_t^{-1/2}(\tilde{x}_t^* - \mu_t) \\ \epsilon^{-1/2}(x_{n+1}^* - \psi(\tilde{x}_t^*, x_{n+1}^{\text{IMU}})) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{d_{\text{IMU}}+(n+1)d_x}, \\ J &= \begin{bmatrix} \Sigma_t^{-1/2} & O \\ -\epsilon^{-1/2}\Psi & \epsilon^{-1/2}I_{d_x} \end{bmatrix} \in \mathbb{R}^{(d_{\text{IMU}}+(n+1)d_x) \times (d_{\text{IMU}}+(n+1)d_x)}, \end{aligned}$$

where $\Psi \in \mathbb{R}^{d_x \times (d_{\text{IMU}}+nd_x)}$ is defined as the Jacobian of $\psi : \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n \rightarrow \mathcal{X}_p$ with respect to \tilde{x}_t at $(\tilde{x}_t^*, x_{n+1}^{\text{IMU}}) \in \mathbb{R}^{d_{\text{IMU}}+(n+1)d_x}$. By Algorithm 1, the Gauss-Newton update is thus given by:

$$\begin{aligned} \Sigma_t(\epsilon) &\leftarrow (J^\top J)^{-1} = \begin{bmatrix} \Sigma_t^{1/2} & O \\ \Psi \Sigma_t^{1/2} & \epsilon^{1/2}I_{d_x} \end{bmatrix} \begin{bmatrix} \Sigma_t^{1/2} & \Sigma_t^{1/2}\Psi^\top \\ O & \epsilon^{1/2}I_{d_x} \end{bmatrix} = \begin{bmatrix} \Sigma_t & \Sigma_t\Phi^\top \\ \Psi\Sigma_t & \Psi\Sigma_t\Psi^\top + \epsilon I_{d_x} \end{bmatrix}, \\ \mu_t(\epsilon) &\leftarrow \tilde{x}_t^* - (J^\top J)^{-1} J^\top C_1(\tilde{x}_t^*, x_{n+1}^*) = 0. \end{aligned}$$

Taking $\epsilon \rightarrow 0$ concludes the proof. \square

Theorem 3.4.2. *The feature update step of the standard MSCKF (Alg. 10) is equivalent to applying a Marginalization step to $c_{MSCKF,t,2} : \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n \times \mathbb{R}^{|S_f|d_f} \rightarrow \mathbb{R}$, with:*

$$c_{MSCKF,t,2}(\tilde{x}_t, f_{S_f}) := \|\tilde{x}_t \boxminus \mu_t\|_{\Sigma_t^{-1}}^2 + \sum_{(x_i, f_j) \in S_{z,1} \cup S_{z,2}} \|z_{i,j} \boxminus h(x_i, f_j)\|_{\Sigma_v^{-1}}^2,$$

where $f_{S_f} \in \mathbb{R}^{|S_f|d_f}$ denotes the stacked vector of all feature positions in S_f (see Alg. 8).

Proof. First, we rewrite $c_{MSCKF,t,2}$ as:

$$c_{MSCKF,t,2}(\tilde{x}_t, f_{S_f}) := \|\tilde{x}_t \boxminus \mu_t\|_{\Sigma_t^{-1}}^2 + \|\tilde{z} \boxminus \tilde{h}(\tilde{x}_t, f_{S_f})\|_{\tilde{\Sigma}_v^{-1}}^2,$$

where $\tilde{z} \in \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z}$, $\tilde{h} : \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n \times \mathbb{R}^{|S_f|d_f} \rightarrow \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z}$: are defined as follows— \tilde{z} denotes the stacked measurement vectors in $\{z_{i,j} | (x_i, f_j) \in S_{z,1} \cup S_{z,2}\} \in \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z}$, $\tilde{h}(\tilde{x}_t, f_{S_f})$ denotes the stacked outputs of the measurement map in $\{h(x_i, f_j) | (x_i, f_j) \in S_{z,1} \cup S_{z,2}\} \in \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z}$, and $\tilde{\Sigma}_v := \text{diag}\{\Sigma_v, \dots, \Sigma_v\} \in \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z \times |S_{z,1} \cup S_{z,2}|d_z}$.

Essentially, by marginalizing the feature position estimates, this step utilizes information from feature measurements to constrain our state estimates. To accomplish this, we choose our algorithm variables as follows:

$$\begin{aligned}\tilde{x}_{t,K} &:= \tilde{x}_t = (x_{t,\text{IMU}}, x_1, \dots, x_n) \in \mathbb{R}^{d_{\text{IMU}} + nd_x + |S_f|d_f}, \\ \tilde{x}_{t,M} &:= f_{S_f} \in \mathbb{R}^{|S_f|d_f}, \\ \bar{x} &:= (\tilde{x}_{t,K}, \tilde{x}_{t,M}) \in \mathbb{R}^{d_{\text{IMU}} + nd_x + |S_f|d_f}, \\ C_M(\tilde{x}_{t,K}, \tilde{x}_{t,M}) &:= \begin{bmatrix} \Sigma_t^{-1/2}(\tilde{x}_t \boxminus \mu_t) \\ \tilde{\Sigma}_v^{-1/2}(\tilde{z} \boxminus \tilde{h}(\tilde{x}_t, f_{S_f})) \end{bmatrix} \in \mathbb{R}^{d_{\text{IMU}} + nd_x + |S_{z,1} \cup S_{z,2}|d_z}.\end{aligned}$$

The Marginalization algorithm block then implies that:

$$\begin{aligned}J_K &:= \frac{\partial C_M}{\partial \tilde{x}_t}(\bar{\mu}_t, f_{S_f}^*) = \begin{bmatrix} \Sigma_t^{-1/2} \\ -\tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,x} \end{bmatrix} \in \mathbb{R}^{(d_{\text{IMU}} + nd_x + |S_{z,1} \cup S_{z,2}|d_z) \times (d_{\text{IMU}} + nd_x)}, \\ J_M &:= \frac{\partial C_M}{\partial f_{S_f}}(\bar{\mu}_t, f_{S_f}^*) = \begin{bmatrix} O \\ -\tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f} \end{bmatrix} \in \mathbb{R}^{(d_{\text{IMU}} + nd_x + |S_{z,1} \cup S_{z,2}|d_z) \times |S_f|d_f},\end{aligned}$$

where we have defined:

$$\begin{aligned}f_{S_f}^* &\in \mathbb{R}^{|S_f|d_f} \leftarrow \text{Stacked position estimates of features in } S_f, \\ \tilde{H}_{t,x} &:= \frac{\partial \tilde{h}}{\partial \tilde{x}_t}(\bar{\mu}_t, f_{S_f}^*) \in \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z \times (d_{\text{IMU}} + nd_x)}, \\ \tilde{H}_{t,f} &:= \frac{\partial \tilde{h}}{\partial f_{S_f}}(\bar{\mu}_t, f_{S_f}^*) \in \mathbb{R}^{|S_{z,1} \cup S_{z,2}|d_z \times |S_f|d_f}.\end{aligned}$$

Recall that the marginalization equations (2.14) and (2.15) in our formulation read:

$$\begin{aligned}\mu_K &\leftarrow \mu_K - \Sigma_K J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] C_M(\tilde{x}_{t,K}, \tilde{x}_{t,M}), \\ \Sigma_K &\leftarrow (J_K^\top (I - J_M (J_M^\top J_M)^{-1} J_M^\top) J_K)^{-1}.\end{aligned}$$

Substituting in the above expressions for J_K , J_M , and $C_M(\bar{\mu}_t, f_{S_f}^*)$, we have:

$$\begin{aligned}\bar{\Sigma}_t &\leftarrow (J_K^\top (I - J_M (J_M^\top J_M)^{-1} J_M^\top) J_K)^{-1}, \\ &= \left(\begin{bmatrix} \Sigma_t^{-1/2} & -\tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1/2} \end{bmatrix} \begin{bmatrix} I & O \\ O & I - \tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f} (\tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f})^{-1} \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1/2} \end{bmatrix} \right. \\ &\quad \left. \begin{bmatrix} \Sigma_t^{1/2} \\ -\tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,x} \end{bmatrix} \right)^{-1} \\ &= (\Sigma_t^{-1} + \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1/2} [I - \tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f} (\tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f})^{-1} \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1/2}] \tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,x})^{-1}\end{aligned}$$

$$\begin{aligned}
\bar{\mu}_t &\leftarrow \mu_K - \Sigma_K J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] C_M (\bar{\mu}_t, f_{S_f}^*) \\
&= \mu_t + (\Sigma_t^{-1} + \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1/2} [I - \tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f} (\tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f})^{-1} \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1/2}] \cdot \tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,x})^{-1} \\
&\quad \cdot \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1/2} [I - \tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f} (\tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f})^{-1} \cdot \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1/2}] \tilde{\Sigma}_v^{-1/2} (\tilde{z} - \tilde{h}(\tilde{x}_t, f_{S_f})).
\end{aligned}$$

Comparing with the update step in the MSCKF algorithm, i.e., (10) and (9), it suffices to show:

$$\begin{aligned}
T^\top (Q^\top A^\top \tilde{\Sigma}_v A Q)^{-1} &= \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1/2} [I - \tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f} (\tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f})^{-1} \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1/2}] \cdot \tilde{\Sigma}_v^{-1/2} \\
&= \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1} - \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f} (\tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f})^{-1} \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1}.
\end{aligned}$$

To see this, recall that A is defined as a full-rank matrix whose columns span $N(\tilde{H}_{t,f}^\top)$. Thus:

$$(\tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f})^\top \cdot \tilde{\Sigma}_v^{1/2} A Q = \tilde{H}_{t,f}^\top A Q = O.$$

In other words, the columns of $\tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f}$ and of $\tilde{\Sigma}_v^{1/2} A Q$ form bases of orthogonal subspaces whose direct sum equals $\mathbb{R}^{n_{qdz}}$. We thus have:

$$\tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f} (\tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f})^{-1} \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1/2} + \tilde{\Sigma}_v^{1/2} A Q (Q^\top A^\top \tilde{\Sigma}_v A Q)^{-1} Q^\top A^\top \tilde{\Sigma}_v^{1/2} = I,$$

which in turn implies that:

$$\begin{aligned}
T^\top (Q^\top A^\top \tilde{\Sigma}_v A Q)^{-1} &= \tilde{H}_{t,x}^\top A Q (Q^\top A^\top \tilde{\Sigma}_v A Q)^{-1} Q^\top A^\top \\
&= \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1/2} (\tilde{\Sigma}_v^{1/2} A Q) (Q^\top A^\top \tilde{\Sigma}_v^{1/2} \cdot \tilde{\Sigma}_v^{1/2} A Q)^{-1} (Q^\top A^\top \tilde{\Sigma}_v^{1/2}) \tilde{\Sigma}_v^{-1/2} \\
&= \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1/2} (I - \tilde{\Sigma}_v^{-1/2} \tilde{H}_{t,f} (\tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f})^{-1} \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1/2}) \tilde{\Sigma}_v^{-1/2} \\
&= \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1} - \tilde{H}_{t,x}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f} (\tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1} \tilde{H}_{t,f})^{-1} \tilde{H}_{t,f}^\top \tilde{\Sigma}_v^{-1},
\end{aligned}$$

as claimed. \square

Theorem 3.4.3. *The state propagation step of the standard MSCKF (Alg. 11) is equivalent to applying a Marginalization step to $c_{MSCKF,t,4} : (\mathcal{X}_{IMU})^2 \times (\mathcal{X}_p)^n \rightarrow \mathbb{R}$, with:*

$$c_{MSCKF,t,4}(\tilde{x}_t, x_{t+1,IMU}) := \|\tilde{x}_t \boxminus \bar{\mu}_t\|_{\tilde{\Sigma}_t^{-1}}^2 + \|x_{t+1,IMU} \boxminus g_{IMU}(x_t, IMU)\|_{\tilde{\Sigma}_t^{-1}}^2.$$

with $\tilde{x}_{t,K} := (x_{t+1,IMU}, x_1, \dots, x_n) \in \mathcal{X}_{IMU} \times (\mathcal{X}_p)^n$ and $\tilde{x}_{t,M} = x_{t,IMU} \in \mathcal{X}_{IMU}$.

Proof. We claim that from an optimization perspective, the update step is equivalent to applying one marginalization step to the cost function $c_{MSCKF,t,4}(\tilde{x}_t, x_{t+1,IMU})$ specified above. In particular, we wish to marginalize out $x_{t,IMU} \in \mathcal{X}_{IMU}$ and retain $x_{t+1,IMU} \in \mathcal{X}_{IMU}$; in other words, in the notation of our Marginalization algorithm submodule, we have:

$$\begin{aligned}
\tilde{x}_{t,K} &:= (x_{t+1,IMU}, x_1, \dots, x_n) \in \mathcal{X}_{IMU} \times (\mathcal{X}_p)^n, \\
\tilde{x}_{t,M} &:= x_{t,IMU} \in \mathcal{X}_{IMU}.
\end{aligned}$$

To apply a marginalization step, our first task is to find vectors $C_K(x_K) = C_K(\tilde{x}_t)$ and $C_M(x_K, x_M) = C_M(\tilde{x}_t, x_{t+1, \text{IMU}})$ of appropriate dimensions such that $c_{\text{MSCKF}, t, 4}(\tilde{x}_t, x_{t+1, \text{IMU}}) = C_K(x_{t+1, \text{IMU}})^\top C_K(x_{t+1, \text{IMU}}) + C_M(\tilde{x}_t, x_{t+1, \text{IMU}})^\top C_M(\tilde{x}_t, x_{t+1, \text{IMU}})$. A natural choice is furnished by $C_K(x_{t+1, \text{IMU}}) \in \mathbb{R}$ and $C_M(\tilde{x}_t, x_{t+1, \text{IMU}}) \in \mathcal{X}_p$, as defined below:

$$C_K(\tilde{x}_{t,K}) = 0 \in \mathbb{R}$$

$$C_M(\tilde{x}_{t,K}, \tilde{x}_{t,M}) = \begin{bmatrix} \bar{\Sigma}_t^{-1/2}(\tilde{x}_t - \bar{\mu}_t) \\ \Sigma_w^{-1/2}(x_{t+1, \text{IMU}} - g_{\text{IMU}}(x_{t, \text{IMU}})) \end{bmatrix} \in \mathbb{R}^{2d_{\text{IMU}} + nd_x}.$$

For convenience, we will define the IMU state and pose components of the mean $\mu_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$ by $\mu_t := (\mu_{t, \text{IMU}}, \mu_{t, \text{IMU}}) \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$, with $\mu_{t, \text{IMU}} \in \mathcal{X}_p$ and $\mu_{t, x} \in (\mathcal{X}_p)^n$, respectively. This mirrors our definition of $x_t \in \mathcal{X}_p$ and $x_{n+1} \in (\mathcal{X}_p)^n$ as the components of the full state $\tilde{x}_t := (x_t, x_{n+1}) \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n$. In addition, we will define the components of $\bar{\Sigma}_t^{-1/2} \in \mathbb{R}^{(d_{\text{IMU}} + nd_x) \times (d_{\text{IMU}} + nd_x)}$ and $\bar{\Sigma}_t^{-1} \in \mathbb{R}^{(d_{\text{IMU}} + nd_x) \times (d_{\text{IMU}} + nd_x)}$ by:

$$\begin{bmatrix} \Omega_{t, \text{IMU}, \text{IMU}} & \Omega_{t, \text{IMU}, x} \\ \Omega_{t, x, \text{IMU}} & \Omega_{t, x, x} \end{bmatrix} := \bar{\Sigma}_t^{-1} \in \mathbb{R}^{(d_{\text{IMU}} + nd_x) \times (d_{\text{IMU}} + nd_x)},$$

$$\begin{bmatrix} \Lambda_{t, \text{IMU}, \text{IMU}} & \Lambda_{t, \text{IMU}, x} \\ \Lambda_{t, x, \text{IMU}} & \Lambda_{t, x, x} \end{bmatrix} := \bar{\Sigma}_t^{-1/2} \in \mathbb{R}^{(d_{\text{IMU}} + nd_x) \times (d_{\text{IMU}} + nd_x)},$$

with the dimensions of the above block matrices given by $\Sigma_{t, \text{IMU}, \text{IMU}}, \Lambda_{t, \text{IMU}, \text{IMU}} \in \mathbb{R}^{d_{\text{IMU}} \times d_{\text{IMU}}}$, $\Sigma_{t, \text{IMU}, x}, \Lambda_{t, \text{IMU}, x} \in \mathbb{R}^{d_{\text{IMU}} \times nd_x}$, $\Sigma_{t, x, \text{IMU}}, \Lambda_{t, x, \text{IMU}} \in \mathbb{R}^{nd_x \times d_{\text{IMU}}}$, and $\Sigma_{t, x, x}, \Lambda_{t, x, x} \in \mathbb{R}^{nd_x \times nd_x}$. Using the above definitions, we can reorder the residuals in $C_K \in \mathbb{R}$ and $C_M \in \mathbb{R}^{2d_{\text{IMU}} + nd_x}$, and thus redefine them by:

$$C_K(\tilde{x}_{t,K}) = 0 \in \mathbb{R}$$

$$C_M(\tilde{x}_{t,K}, \tilde{x}_{t,M}) = \begin{bmatrix} \Lambda_{t, \text{IMU}, \text{IMU}}(x_{t, \text{IMU}} - \mu_{t, \text{IMU}}) + \Lambda_{t, \text{IMU}, x}(x_{1:n} - \mu_{t, x}) \\ \Sigma_w^{-1/2}(x_{t+1, \text{IMU}} - g_{\text{IMU}}(x_{t, \text{IMU}})) \\ \Lambda_{t, x, \text{IMU}}(x_{t, \text{IMU}} - \mu_{t, \text{IMU}}) + \Lambda_{t, x, x}(x_{1:n} - \mu_{t, x}) \end{bmatrix} \in \mathbb{R}^{2d_{\text{IMU}} + nd_x},$$

where $x_{1:n} := (x_1, \dots, x_n) \in (\mathcal{X}_p)^n$.

Our state variables and cost functions for the Gauss-Newton algorithm submodule are:

$$\bar{x}_M^* = \tilde{x}_t^* = \bar{\mu}_t \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n,$$

$$\bar{x}_K^* = g(\tilde{x}_t^*) = g(\bar{\mu}_t) \in \mathcal{X}_{\text{IMU}} \times (\mathcal{X}_p)^n,$$

$$C_K(\tilde{x}_{t,K}^*) = 0 \in \mathbb{R},$$

$$C_M(\tilde{x}_{t,K}^*, \tilde{x}_{t,M}^*) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{2d_{\text{IMU}} + nd_x},$$

$$J_K = \begin{bmatrix} O & \Lambda_{\text{IMU}, x} \\ \Sigma_w^{-1/2} & O \\ O & \Lambda_{xx} \end{bmatrix} \in \mathbb{R}^{(2d_{\text{IMU}} + nd_x) \times (d_{\text{IMU}} + nd_x)},$$

$$J_M = \begin{bmatrix} \Lambda_{\text{IMU,IMU}} \\ -\Sigma_w^{-1/2} G_t \\ \Lambda_{x,\text{IMU}} \end{bmatrix} \in \mathbb{R}^{(2d_{\text{IMU}}+nd_x) \times d_x},$$

where we have defined G_t to be the Jacobian of $g_{\text{IMU}} : \mathcal{X}_{\text{IMU}} \rightarrow \mathcal{X}_{\text{IMU}}$ at $\overline{\mu_{t,\text{IMU}}} \in \mathcal{X}_{\text{IMU}}$, i.e.:

$$G_t := \left. \frac{\partial g}{\partial x_{t,\text{IMU}}} \right|_{x_{t,\text{IMU}} = \overline{\mu_{t,\text{IMU}}}}$$

Applying the Marginalization update equations, we thus have:

$$\begin{aligned} \mu_{t+1} &\leftarrow \tilde{x}_{t,K} - \Sigma_{t+1} J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] C_M (\overline{x_K^*}, \overline{x_M^*}) \\ &= g(\overline{\mu_t}), \\ \Sigma_{t+1} &\leftarrow (J_K^\top [I - J_M (J_M^\top J_M)^{-1} J_M^\top] J_K)^{-1}, \\ &= (J_K^\top J_K - J_K^\top J_M (J_M^\top J_M)^{-1} J_M^\top J_K)^{-1}, \\ &= \left(\begin{bmatrix} \Sigma_w^{-1} & O \\ O & \Lambda_{x,\text{IMU}} \Lambda_{\text{IMU},x} + \Lambda_{xx}^2 \end{bmatrix} - \begin{bmatrix} -\Sigma_w^{-1} G_t \\ \Lambda_{x,\text{IMU}} \Lambda_{\text{IMU,IMU}} + \Lambda_{xx} \Lambda_{x,\text{IMU}} \end{bmatrix} \right) \\ &\quad \cdot (\Lambda_{\text{IMU,IMU}}^2 + \Lambda_{\text{IMU},x} \Lambda_{x,\text{IMU}} + G_t^\top \Sigma_w^{-1} G_t)^{-1} \\ &\quad \cdot \left[-G_t^\top \Sigma_w^{-1} \quad \Lambda_{\text{IMU,IMU}} \Lambda_{\text{IMU},x} + \Lambda_{x,\text{IMU}} \Lambda_{xx} \right]^{-1} \\ &= \left(\begin{bmatrix} \Sigma_w^{-1} & O \\ O & \Omega_{xx} \end{bmatrix} - \begin{bmatrix} -\Sigma_w^{-1} G_t \\ \Omega_{x,\text{IMU}} \end{bmatrix} (\Omega_{\text{IMU,IMU}} + G_t^\top \Sigma_w^{-1} G_t)^{-1} \begin{bmatrix} -G_t^\top \Sigma_w^{-1} & \Omega_{\text{IMU},x} \end{bmatrix} \right)^{-1} \end{aligned}$$

To show that this is indeed identical to the propagation equation for the covariance matrix in the Extended Kalman Filter algorithm, i.e. Algorithm 3, Line 5, we must show that:

$$\begin{aligned} &\left(\begin{bmatrix} \Sigma_w^{-1} & O \\ O & \Omega_{xx} \end{bmatrix} - \begin{bmatrix} -\Sigma_w^{-1} G_t \\ \Omega_{x,\text{IMU}} \end{bmatrix} (\Omega_{\text{IMU,IMU}} + G_t^\top \Sigma_w^{-1} G_t)^{-1} \begin{bmatrix} -G_t^\top \Sigma_w^{-1} & \Omega_{\text{IMU},x} \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} G_t \overline{\Sigma}_{t,\text{IMU,IMU}} G_t^\top + \Sigma_w & G_t \overline{\Sigma}_{t,\text{IMU},x} \\ \overline{\Sigma}_{t,\text{IMU},x} G_t^\top & \overline{\Sigma}_{t,x,x} \end{bmatrix} \end{aligned}$$

This follows by brute-force expanding the above block matrix components, and applying Woodbury's Matrix Identity, along with the definitions of $\Sigma_{t,\text{IMU,IMU}}$, $\Lambda_{t,\text{IMU,IMU}}$, $\Sigma_{t,\text{IMU},x}$, $\Lambda_{t,\text{IMU},x}$, $\Sigma_{t,x,\text{IMU}}$, $\Lambda_{t,x,\text{IMU}}$, $\Sigma_{t,x,x}$, and $\Lambda_{t,x,x}$. □

3.5 State-of-the-Art SLAM Algorithms

Our framework balances the need for computational efficiency, estimation accuracy, and map precision, tradeoffs observed in design choices for existing SLAM algorithms.

- **Extended Kalman Filter (EKF)** [105, 98, 117] –The EKF iteratively updates position estimates of the current pose and all observed features; all past poses are marginalized. This design favors computational speed over localization and mapping accuracy. A variant, the *iterated Extended Kalman Filter* (iEKF), takes multiple Gauss-Newton steps before marginalization to tune the linearization point. This improves mapping and localization accuracy but increases computation time slightly.
- **Iterated Extended Kalman Filter (iEKF)** [117] [106]–The iterated EKF algorithm extends the EKF by performing multiple steps of Gauss-Newton descent before the marginalization step, to ensure that marginalization occurs about a sufficiently accurate linearization point. Compared to standard EKF, this design improves mapping and localization accuracy at the cost of a heavier computational burden.
- **Multi-State Constrained Kalman Filter** [79, 69, 70]—The MSCKF iteratively updates a full state, with the current IMU state and n past poses, while processing features observed at these poses; here $n \leq N_{\max}$, a specified upper bound that trades off accuracy and computational speed.
- **Sliding Window Smoother, Fixed-Lag Smoother** [76, 97, 30]—The fixed-lag smoother resembles the MSCKF, but performs multiple steps of Gauss-Newton descent before the marginalization step, to adjust the linearization point. This improves localization and mapping accuracy, but increases the computation time.
- **Open Keyframe Visual-Inertial SLAM (OKVIS)** [67]—OKVIS updates a sliding window of “keyframes”, poses deemed most informative, which may be arbitrarily spaced in time. Keyframe poses leaving the sliding window, and associated landmarks, are marginalized. This design aims to improve estimation accuracy by maximizing information encoded by the stored poses, without increasing computation time.
- **GraphSLAM, Bundle Adjustment** [105, 50] –These algorithms solve the full SLAM problem (no marginalization), often with high accuracy, but can be very slow.

3.6 Experiments

This section describes the empirical performance of different marginalization schemes on pose tracking of real-world data. We examine the MSCKF [79], a standard SWF, and the keyframe-based OKVIS algorithm [67], each implemented using our unifying framework.

Simulation Settings

Experiments are performed on the EuRoC MAV dataset of stereo images and IMU data [13]. We standardize the front-end across all experiments, using BRISK keypoint features

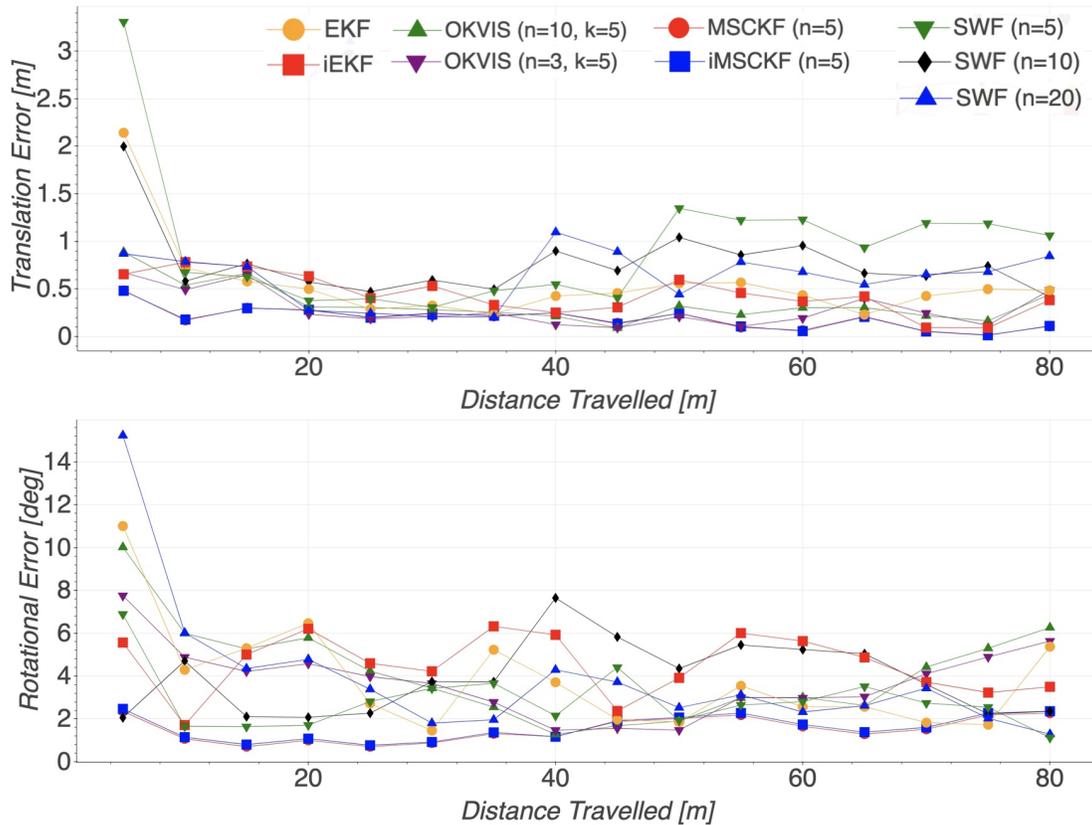


Figure 3.1: Localization on Vicon Room 2 (medium): Drift from, vs. distance traveled along, the ground-truth trajectory, at 5 meter intervals. We apply trajectory alignment as in [107].

with brute-force matching. Outlier rejection between stereo cameras is performed via epipolar constraint tests, and outlier rejection between stereo frames at subsequent timesteps is performed via the reprojection distance test, using the latest estimate of the camera pose and feature positions. We use the GTSAM back-end in C++ to construct and update costs, compute Jacobians, and implement Gauss-Newton and marginalization steps [28, 29].

To construct dynamics and measurement maps, we collect on-board IMU odometry measurements, and apply IMU pre-integration scheme as in [43] and trajectory alignment as in [107]. Please see [43, 107, 95] for more details.

Results and Discussion

Localization root-mean-squared error on Vicon Room and Machine Hall sequences from the Euroc MAV dataset are presented in Figure 3.2. Due to space constraints, only the estimator drift on the V2_02 sequence is plotted (Fig 3.1). First, we analyze standard SWFs of window

Data	MSCKF(n=5)	iMSCKF(n=5)	OKVIS(n=10,k=5)	OKVIS(n=3,k=5)	SWF(n=10)	SWF(n=20)	SWF(n=5)	EKF	iEKF
V1_01	0.09m, 2.87°	0.09m, 2.87°	0.20m, 3.62°	0.23m, 3.68°	0.36m, 3.12°	0.36m, 4.72°	1.00m, 8.22°	1.10m, 8.79°	0.04m, 59.10°
V1_02	0.15m, 1.52°	0.16m, 1.55°	-	-	0.33m, 3.90°	1.16m, 4.69°	0.30m, 3.68°	0.75m, 8.13°	0.42m, 5.48°
V1_03	1.00m, 4.84°	1.12m, 4.90°	0.42m, 6.49°	0.46m, 6.99°	-	6.36m, 25.85°	0.79m, 6.47°	14.84m, 24.58°	1.83m, 8.69°
V2_01	0.14m, 0.83°	0.14m, 0.80°	0.45m, 2.88°	0.39m, 2.58°	0.75m, 5.23°	0.23m, 1.71°	0.26m, 1.96°	0.79m, 3.67°	0.48m, 3.58°
V2_02	0.24m, 1.67°	0.24m, 1.74°	0.44m, 4.69°	0.38m, 4.13°	0.96m, 3.71°	0.75m, 4.49°	1.42m, 2.99°	0.83m, 4.32°	0.47m, 4.14°
MH01	0.07m, 1.03°	0.07m, 1.03°	0.63m, 8.44°	0.72m, 11.61°	0.19m, 1.31°	0.11m, 1.14°	0.43m, 4.02°	0.46m, 3.09°	0.42m, 3.11°
MH02	0.14m, 1.05°	0.19m, 1.86°	0.76m, 8.79°	0.93m, 9.99°	0.23m, 1.89°	0.23m, 1.58°	0.28m, 2.80°	0.34m, 2.21°	0.50m, 2.68°
MH03	0.26m, 1.36°	0.25m, 1.35°	0.64m, 4.29°	0.86m, 5.79°	0.32m, 1.77°	0.54m, 1.75°	0.25m, 1.60°	1.19m, 4.62°	1.33m, 5.23°
MH04	1.11m, 1.62°	1.06m, 1.52°	-	-	0.79m, 1.75°	0.86m, 1.79°	0.59m, 1.28°	4.10m, 4.58°	5.21m, 6.32°

Figure 3.2: Root-mean-squared error in translation and rotation on Vicon Room and Machine Hall sequences from the Euroc MAV dataset. We apply trajectory alignment as in [107].

size $n = 5, 10, 20$ frames. Features are marginalized when they are only visible in the oldest frame in the optimization window. EKF and iEKF are also included, and are implemented as SWFs with window size 1. For the former, only 1 Gauss-Newton step is taken, and for the latter, steps are taken until convergence. Next, we implement MSCKF via incremental optimization (Section 3.4), with window size $n = 5$, and with two optimization schemes: (1) standard, with one Gauss-Newton step after marginalization, and (2) a version that takes steps until convergence (“Iterated MSCKF,” or iMSCKF). Finally, we implement OKVIS with IMU window size $n = 3, 10$, keyframe window size $k = 5$, and marginalization and keyframe selection schemes as in Leutenegger et al. [67].

Our experiments show that, overall, OKVIS outperforms baseline SWFs, even when the latter has a larger window size. Moreover, our MSCKF implementation outperforms SWFs and OKVIS, even under challenging camera motions, despite the latter maintaining nonlinear constraints between camera poses and landmarks, and taking multiple Gauss-Newton steps per iteration. This persists even for SWFs with larger window sizes. Taking multiple Gauss-Newton steps in the iMSCKF estimator did not noticeably improve performance over the standard MSCKF (Figure 3.1).

In contrast with SWF and OKVis implementations of comparable sizes, the MSCKF recovers better from localization errors, by employing a marginalization scheme that always maintains poses arbitrarily far in the past. This is because older poses represent higher baselines and thus supply better localization information [7]. For instance, the MSCKF maintains the first pose in the estimator for a long time, thus enforcing consistency with subsequent pose estimates and minimizing drift. In contrast, although OKVIS allows older keyframes to be maintained, in practice keyframes are usually roughly evenly spaced and form a sliding temporal window of camera motion. Thus, earlier poses are quickly marginalized, causing estimates to drift more at the start of the trajectory. Furthermore, the MSCKF processes features in the optimization window only after they have matured. Thus, it maximally utilizes localization information with fewer updates, and ensures that each feature is always initialized through multiple-view, instead of merely stereo, triangulation. This minimizes the linearization error when features are processed and dropped.

3.7 Discussion

This chapter presented a framework for formulating and analyzing optimization and filtering-based SLAM approaches as the iterative application of key algorithm submodules, and proves that it encompasses state-of-the-art filtering algorithms as special cases. Experimental analysis indicate our formulation is useful for analyzing various design choices inherent in these existing SLAM algorithms, and implementing them in a modular fashion for a wide range of robotics applications, which we are eager to test on hardware.

Part II

Game-Theoretic Motion Planning for Autonomous Vehicles

Chapter 4

Defensive Driving

When deployed in real-world applications, autonomous vehicles and robots should operate safely in a robust manner. Violations of safety constraints can lead to catastrophic crashes in real-life traffic scenarios, particularly when the self-driving system on a semi-autonomous car is directed by a distracted individual (Figure 4.1), or when an autonomous car is in the vicinity of human-operated vehicles piloted by distracted drivers. To avoid such disastrous outcomes, autonomous navigation pipelines must be capable of endowing self-driving vehicles with the ability to reason about how their behavior and the behavior of other vehicles in their vicinity affect each other.

In scenarios in which an “ego” agent navigates in an environment with multiple other “non-ego” agents, existing methods formulate safety for the ego agent in the following two ways. Adversarial robustness methods, such as Hamilton-Jacobi-Isaacs (HJI) equation-based reachability theory, aim to generate trajectories that would ensure the safety of the ego agent despite worst-case behavior of all other agents [4, 78, 74]. Another commonly proposed methodology involves probabilistic constraint satisfaction [36, 81]. Here, algorithms attempt to bound the probability that the ego agent’s trajectory becomes unsafe. Unfortunately, each of these approaches carries significant drawbacks. HJI methods only apply to zero-sum game settings, and exact solution methods suffer from the so-called “curse of dimensionality,” with computational cost increasing exponentially in the dimension of the state [4]. Moreover, decision-making based on reachability-based methods may be overly conservative, resulting in excessive congestion and gridlock. Meanwhile, probabilistic constraint satisfaction encodes safety via distributional assumptions, but does not allow the ego player to anticipate more specific patterns of adversarial interactions with non-ego agents.

To address these shortcomings, we present a novel formulation of robustness within the framework of general-sum dynamic game theory, modeled on defensive driving. Here, the ego player pre-supposes that: (1) For an initial fixed window of time T_{adv} , other agents are temporarily distracted, and thus unintentionally behave in an erratic or even adversarial manner, and (2) After this time interval $[0, T_{adv}]$ has expired, the agent returns to normal operation, and is willing to cooperate with the ego agent to ensure safe operation, e.g., to avoid collisions. To encode this, we prepend an *adversarial phase* to the ego agent’s cost



Figure 4.1: As reported by Neal E. Boudette and Bill Vlasic of the New York Times [12] (“Tesla Self-Driving System Faulted by Safety Agency in Crash,” September 12, 2017), the self-driving Tesla vehicle shown above operated on a system for autonomous steering and control. Unfortunately, it was used by a driver who was distracted for an extended period of time, and guided the vehicle down a road on which it was not designed to be driven. In general, the failure of autonomous navigation pipelines to ensure safe operation raises serious concerns about the mass deployment of self-driving vehicles in real-life traffic.

function. In other words, we prepend a time interval during which other agents are assumed to be temporarily distracted, in order to render the ego agent’s equilibrium trajectory robust against other agents’ potentially dangerous behavior during this time. This is followed by a *cooperative phase*, during which the non-ego agents are expected to resume “normal” cooperative behavior. This paradigm is modeled on the concept of defensive driving, wherein a driver on a busy road guards themselves against other drivers who, while momentarily distracted, may temporarily behave dangerously. We demonstrate the effectiveness of our new formulation in encoding safety via multiple traffic scenarios.

The rest of this chapter proceeds as follows. Section 4.1 presents related work on the use of HJI reachability theory and chance-constrained optimal control to encode adversarial and probabilistic robustness, respectively, as well as recent literature on iterative algorithms for solving dynamic games. Section 4.2 presents the mathematical formulation for the dynamic game that models the multi-agent interactions studied in our work. Section 4.3 presents the main methodology of our algorithm, and formally introduces the concept of an adversarial

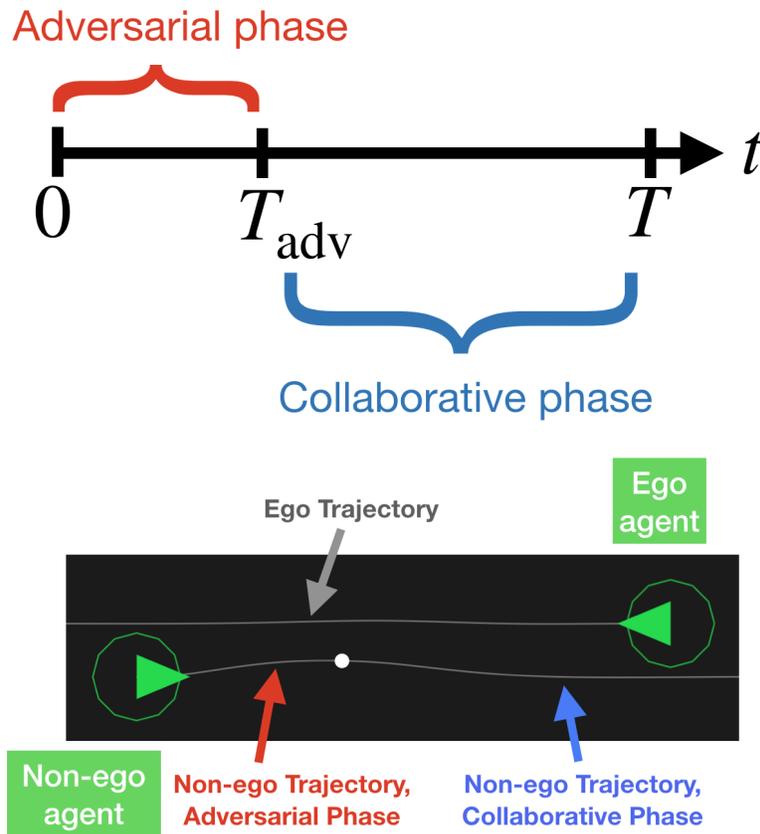


Figure 4.2: (*Top*) To encode robustness into its safety guarantees, the ego agent imagines that all other agents behave adversarially during an initial time frame before resuming “normal” cooperative behavior. (*Bottom*) In this example, an ego vehicle assumes that the non-ego, oncoming vehicle behaves adversarially for an initial period of time, which results in its swerving into the ego’s lane. After the adversarial time interval expires, the non-ego agent is assumed to return to its own lane, and resume cooperative behavior for the remainder of the time horizon.

time horizon. Section 4.5 demonstrates the spectrum of robustness which can be expressed in our formulation, compared with a purely cooperative game-theoretic approach, using multiple traffic scenarios. Section 4.6 summarizes our contributions and discusses directions for future research. More details can be found in a published paper, Chiu* and Fridovich Keil* “Encoding Defensive Driving as a Dynamic Nash Game” [21] (*Equal contribution).

4.1 Related Work

This section reviews prior literature on several of the prevailing formulations of safety used in the design of multi-agent and uncertain systems—*adversarial reachability*, *multi-agent forward reachability*, and *probabilistic constraint satisfaction*. We compare and contrast these against the novel methodology in our work. We conclude this section with a brief summary of modern algorithms for iteratively solving dynamic games.

Adversarial Reachability

Adversarial reachability methods [35, 78, 39, 4] involve the construction of a zero-sum differential game between two agents, the ego agent and an adversary. The Nash equilibrium of this game satisfies a Hamilton-Jacobi-Isaacs (HJI) partial differential equation, which can be numerically solved via state space discretization. In this zero-sum framework, the ego agent assumes that the adversarial agent is constantly attempting to compromise the ego’s safety, and will thus compute and execute a control strategy that steers the ego’s trajectory away from any feasible trajectory of the non-ego agent.

Adversarial reachability appropriately describes many intricate dynamic interactions, such as capture-the-flag, reach-avoid, and pursuit-evasion-defense games [55, 54, 39]. However, it suffers from several significant limitations. First, this zero-sum formulation can only model dynamic interactions between two agents, or two groups of colluding agents, with opposing goals. This is inadequate for many motion-planning tasks, such as those involving traffic scenarios, which must account for the presence of an arbitrary number of agents, with possibly an arbitrary number of goals. Second, the adversarial nature of the zero-sum game leads to the construction of extremely conservative ego trajectories, since the ego must imagine the worst-case non-ego behaviors that can possibly transpire Figure 4.2. Our approach, on the other hand, avoids the first issue by considering a general-sum game applicable to N -player scenarios. Moreover, we avoid the second issue by modeling antagonistic non-ego behavior using the novel notion of an adversarial-to-cooperative time horizon, as shown in Figure 4.2, rather than as a worst-case bounded disturbance. By modeling non-ego agents as first adversarial, then cooperative, we avoid overly conservative ego strategies corresponding to purely adversarial non-ego trajectories that are unlikely to materialize.

Multi-Agent Forward Reachability

Reachability-based formulations can also be used for safety-critical path planning in a non-game theoretic manner. For instance, forward reachable sets (FRS) of the ego agent can be computed offline by numerically solving the Hamilton-Jacobi-Bellman equation, then used to aid online motion planning modules in the generation of obstacle-avoiding trajectories. This is the approach taken by the Reachability-based Trajectory Design for Dynamical environments (RTD-D) and RTD-Interval (RTD-I) algorithms presented in Vaskov et al. and Yu et al. [108, 113], in which not-at-fault ego trajectories are generated by leveraging the offline-computed FRS of the ego agent and online obstacle motion predictions from an external module. Although the resulting trajectories avoid at-fault collisions, this framework does not allow the ego agent to account for the dynamic reactions of non-ego agents to its behavior. In contrast, our work explicitly models the dynamic obstacles as non-ego agents, within the framework of a dynamic feedback game.

Geometric prediction modules form another framework for using reachability-based methods in a non-game-theoretic setting. For instance, Koschi et al. and Pek et al. [62, 86] ensure constraint satisfaction by computing fail-safe ego trajectories which avoid an overapproximation of all dynamically feasible non-ego trajectories. This is posed as an optimal control problem (rather than a dynamic game), with avoidance of the set of all feasible non-ego trajectories serving as a state constraint. Although these approaches ensure that the ego will not collide with the non-ego agent, they do so at the cost of generating overly conservative maneuvers, particularly in situations when the non-ego agent may not be purely adversarial throughout the entire time horizon. By contrast, our formulation generates less conservative trajectories by assuming that non-ego agents display hostile behavior only during a fixed subset of the overall time horizon.

Probabilistic Constraint Satisfaction

Probabilistic constraint satisfaction is another commonly used framework for establishing safety guarantees in motion planning. These approaches bound the probability that an ego agent, operating in an unpredictable environment with stochastic disturbance, becomes unsafe [81]. In particular, risk-sensitive algorithms guard the ego agent from low-probability events that may result in highly dangerous outcomes. For example, Farshidian et al. [37] generates risk-sensitive trajectories by optimizing an exponential-quadratic cost term that amplifies the cost of low-probability, yet highly dangerous outcomes. Meanwhile, Pilipovsky et al. [36] associate individual constraint violations with different penalties, and optimizes their allocation over the horizon. Likewise, Du et al. [33] use a probabilistic framework to capture state uncertainties of an autonomous robotic agent, and uncertainties in the geometry or dynamics of obstacles in its environment. However, the control strategies generated by these methods merely account for the nonzero probability of unsafe outcomes occurring any time within the entire time horizon. By contrast, our work allows the ego to explicitly encode adversarial non-ego behavior inside a specific subset of the time horizon, when such

behavior is most expected to occur.

Algorithms for Solving Dynamic Games

Several alternative methods exist in the literature for numerically solving the general-sum differential games considered in this work. For instance, these games can be formulated collections of coupled Hamilton-Jacobi equations. The solutions to these equations, which correspond to local Nash equilibrium strategies [100, 99], can subsequently be numerically evaluated via grid-based approaches or other methods for state space discretization. Unfortunately, these algorithms suffer computational costs, and requires memory storage that scales exponentially in the state dimension [9]. Therefore, they are unsuitable for modeling the high-dimensional, multi-player interactions considered in this chapter.

Another category of numerical methods for dynamic games involves Iterative Best Response (IBR) algorithms [40, 111]. These algorithms iterate through the players, repeatedly solving the optimal control problem of finding the best-response strategy of each player, assuming all other players' strategies are currently fixed at the values in the previous iterate. Replacing the full dynamic game with a sequence of optimal control problems reduces computation time at each iteration; however, IBR algorithms can still be computationally inefficient overall.

Our work in this chapter employs ILQGames [46], an iterative linear-quadratic algorithm, as our primary game solver. ILQGames iteratively computes linear approximations of the ego agent's and non-ego agents' dynamics, and quadratic approximations of the ego agent's and non-ego agents' cost, effectively generating a linear quadratic game to approximate the original general-sum game. Then, the algorithm computes the feedback Nash equilibrium of the linear quadratic game, to approximate the feedback Nash equilibrium of the original game. ILQGames incurs computational complexity cubic in the number of players and linear in the time horizon. Although other game solvers, such as the Augmented Lagrangian GAME-theoretic Solver (ALGAMES) [23], exist, all known implementations of these methods are restricted to an open-loop [32, 31]—rather than feedback—information structure and have equivalent computational complexity.

4.2 Preliminaries

Consider the N -player finite horizon general-sum differential game with deterministic, noise-free nonlinear system dynamics:

$$\dot{x} = f(t, x, u_{1:N}). \quad (4.1)$$

Here, $x \in \mathbb{R}^n$ is the state of the system, obtained by concatenating the dynamical quantities of interest of each player, $t \in \mathbb{R}$ denotes time, $u_i \in \mathbb{R}^{m_i}$ is the control input of player i , for each $i \in \{1, \dots, N\} := [N]$, and $u_{1:N} := (u_1, \dots, u_N) \in \mathbb{R}^m$, where $m := \sum_{i=1}^N m_i$. The dynamics map $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is assumed to be continuous in t and continuously differentiable in x and u_i , for each $i = 1, \dots, N$ and each $t \in [0, T]$. Since we wish to ensure

the safety of one particular player amidst their interactions with all other players, we refer to Player 1 as the *ego agent*, and the other players as *non-ego agents*. Each player's objective is defined as the integral of a running cost $g_i : [0, T] \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ over the time horizon $[0, T]$:

$$\text{cost}_i(u_{1:N}(\cdot)) = \int_0^T g_i(t, x(t), u_{1:N}(t)) dt, \quad (4.2)$$

for each $i \in \{1, \dots, N\}$. The running costs g_i encode implicit dependence on the state trajectory $x(\cdot) : [0, T] \rightarrow \mathbb{R}^n$ and explicit dependence on the control signals $u_i(\cdot) : [0, T] \rightarrow \mathbb{R}^m$.

To minimize its cost, each player selects a control strategy to employ over the time horizon $[0, T]$, as described below. We assume that, at each time $t \in [0, T]$, each player i observes the state $x(t)$, but no other control input $\{u_j(t) \mid j \neq i\}$, and uses this information to design their control, i.e.

$$u(t) := \gamma_i(t, x(t)),$$

where $\gamma_i : [0, T] \times \mathbb{R}^n \rightarrow \mathbb{R}^{m_i}$, defined as Player i 's *strategy*, is assumed to be measurable. We define the *strategy space of Player i* , denoted Γ_i , as the collection of all of Player i 's possible strategies, and denote, with a slight abuse of notation, the overall cost cost_i of each Player i by:

$$\text{cost}_i(\gamma_1; \dots; \gamma_N) := \text{cost}_i(\gamma_1(\cdot, x(\cdot)), \dots, \gamma_N(\cdot, x(\cdot))).$$

In practice, we shall solve for strategies γ_i that are time-varying, affine functions of x .

We now define the *Nash equilibrium* of the above game.

Definition 4.2.1. (*Nash equilibrium*, [5, Chapter 6]) *The strategy set $(\gamma_1^*, \dots, \gamma_N^*)$ is said to be a Nash equilibrium if no player is unilaterally incentivized to deviate from his or her strategy. Precisely, the following inequality must hold for each player i :*

$$\begin{aligned} \text{cost}_i^* &:= \text{cost}_i(\gamma_1^*, \dots, \gamma_{i-1}^*, \gamma_i^*, \gamma_{i+1}^*, \dots, \gamma_N^*) \\ &\leq \text{cost}_i(\gamma_1^*, \dots, \gamma_{i-1}^*, \gamma_i, \gamma_{i+1}^*, \dots, \gamma_N^*), \forall \gamma_i \in \Gamma_i. \end{aligned} \quad (4.3)$$

Computing a global Nash equilibrium is intractable for dynamic games with general dynamics and cost functions. As such, in this work, we concern ourselves with finding a *local* Nash equilibrium, which is defined similarly to (4.3), but with the inequalities only constrained to hold within a neighborhood of the strategy set $(\gamma_1^*, \dots, \gamma_N^*)$. Moreover, in the dynamic games considered in this chapter, we impose additional constraints on the dynamical quantities of each player, to model appropriate behavior between autonomous agents in traffic scenarios. These constraints will translate into a set of *state constraints*, and will significantly affect the set of Nash equilibria of the game. As such, in this work, we search for a (similarly defined) *generalized local Nash equilibrium*.

4.3 Methods

Our main contribution is a novel formulation of safety, best understood through the lens of defensive driving. In Section 4.3, we describe how, in the ego agent’s mind, the concept of defensive driving can be encoded into the running cost of each non-ego agent, i.e. $g_i(x, u_{1:N})$, for each $i \in \{2, \dots, N\}$. To demonstrate this defensive driving framework in practice, we simulate realistic traffic scenarios; Section 4.3 details the dynamics, costs, and constraints imposed on the various agents in these simulations. Finally, in Section 4.4, we summarize the ILQGames algorithm as the main feedback game solver used in this work.

Encoding Defensive Driving as a Running Cost

In our framework, the ego agent (Player 1) operates under the assumption that all other agents are momentarily distracted. To encode this “imagined” scenario, the ego agent imagines the overall time horizon $[0, T]$ as divided into two sub-intervals, the *adversarial interval* $[0, T_{\text{adv}}]$ and *cooperative interval* $[T_{\text{adv}}, T]$, with $0 < T_{\text{adv}} < T$. During the adversarial interval, the ego agent imagines the other agents to be “momentarily distracted,” and desires to act *defensively*. This phenomenon is modeled using an adversarial running cost $g_{\text{adv},i} : \mathbb{R}^n \times \mathbb{R}^{Nm} \rightarrow \mathbb{R}$ for each $i \in \{2, \dots, N\}$. On the other hand, during the cooperative interval, the ego agent supposes that the other agents have reverted to their “normal” or “cooperative” manner, and thus proceeds to select control signals for the remainder of the time horizon in a less conservative manner. This behavior is captured using a cooperative running cost $g_{\text{coop},i} : \mathbb{R}^n \times \mathbb{R}^{Nm} \rightarrow \mathbb{R}$ for each $i \in \{2, \dots, N\}$. In other words, the running cost of each non-ego agent g_i can be piecewisely defined as follows:

$$g_i(t, x, u_{1:N}) = \begin{cases} g_{\text{adv},i}(x, u_{1:N}), & t \in [0, T_{\text{adv}}), \\ g_{\text{coop},i}(x, u_{1:N}), & t \in [T_{\text{adv}}, T]. \end{cases}$$

In this scenario, the net integrated cost J_i , first defined in (4.2) can be written as follows:

$$J_i = \int_0^{T_{\text{adv}}} g_{\text{adv},i}(x, u_{1:N}) dt + \int_{T_{\text{adv}}}^T g_{\text{coop},i}(x, u_{1:N}) dt. \quad (4.4)$$

With increasing T_{adv} , the ego agent imagines an increasingly adversarial encounter and acts more and more defensively as a result. In practice, the user or system designer would select a suitable T_{adv} before operation, e.g., by choosing the largest T_{adv} such that the solution deviates from a nominal solution with $T_{\text{adv}} = 0$ sufficiently little.

Simulation Setup

To test this construction, we simulate two traffic encounters in ILQGames [46] that involve significant interaction (see Sec. 4.5), in which a responsible human driver would likely drive defensively. Our method attempts to capture the spectrum of this “defensive” behavior as

T_{adv} , the duration of the adversarial time horizon, is varied. In each setting, each agent (in this case, each car) has augmented bicycle dynamics, i.e.:

$$\begin{aligned} \dot{p}_{x,i} &= v_i \sin \theta_i, & \dot{v}_i &= a_i, \\ \dot{p}_{y,i} &= v_i \cos \theta_i, & \dot{\phi}_i &= \omega_i, \\ \dot{\theta}_i &= (v_i/L_i) \tan \phi_i, & \dot{a}_i &= j_i, \end{aligned} \quad (4.5)$$

where $x = (p_{x,i}, p_{y,i}, \theta_i, v_i, \phi_i, a_i)_{i=1}^N$ encapsulates the position, heading, speed, front wheel angle, and acceleration of all vehicles, $u_i = (\omega_i, j_i)$ represents each vehicle’s front wheel rate and tangent jerk, and L_i is each player’s inter-axle distance.

We define $g_{\text{adv},i}$ and $g_{\text{coop},i}$ as weighted combinations of the following functions, with different behavior encouraged through the use of different weighting coefficients. We denote $p_i = (p_{x,i}, p_{y,i})$ for each agent position, $d_{\ell_i}(p_i)$, defined below, for the distance between an agent and the corresponding lane centerline ℓ_i in the $(p_{x,i}, p_{y,i})$ -plane, and d_{prox} for a constant desired minimum proximity between agents:

$$\text{lane center: } \left[d_{\ell_i}(p_i) := \min_{p_\ell \in \ell_i} \|p_\ell - p_i\| \right]^2 \quad (4.6)$$

$$\text{ideal speed: } (v_i - v_{\text{ref},i})^2 \quad (4.7)$$

$$\text{cooperative: } \mathbf{1}\{\|p_i - p_j\| < d_{\text{prox}}\} (d_{\text{prox}} - \|p_i - p_j\|)^2 \quad (4.8)$$

$$\text{adversarial: } \|p_i - p_j\|^2 \quad (4.9)$$

$$\text{input: } u_i^T R_{ii} u_i. \quad (4.10)$$

Recall that, for non-ego agents, the “adversarial” cost is only present during the adversarial horizon $[0, T_{\text{adv}})$ and the “cooperative” cost is present thereafter during the cooperative horizon $[T_{\text{adv}}, T]$. We also enforce the following inequality constraints, where d_{lane} denotes the lane half-width, and \underline{v}_i and \bar{v}_i denote speed limits:

$$\text{proximity: } \|p_i - p_j\| > d_{\text{prox}} \quad (4.11)$$

$$\text{lane: } |d_{\ell_i}(p_i)| < d_{\text{lane}} \quad (4.12)$$

$$\text{speed range: } \underline{v}_i < v_i < \bar{v}_i, \quad (4.13)$$

Here, the “proximity” constraint is enforced for only the ego agent, to force the ego to bear responsibility for satisfying joint state constraints which encode his or her own safety (e.g. non-collision). In addition, all agents must satisfy individual constraints that encode reasonable conduct in traffic (e.g., staying within a range of speeds). All constraints are enforced over the entire time horizon $[0, T]$. For all tests, we use a time horizon $T = 15$ s and discretize time (following [46] and [5]) at 0.1 s intervals.

Receding Horizon Control

In realistic motion-planning scenarios, such as the real-life traffic encounters simulated in this work, the ego agent cannot perfectly predict the trajectories of non-ego agents throughout the

time horizon $[0, T]$. As such, in addition to adopting the defensive driving framework, we need to further robustify the ego agent’s trajectory against unpredictable and possibly adversarial non-ego decisions. To do so, we encode the ego agent to follow its own Nash strategy for a short time, before re-computing an updated strategy by re-solving the dynamic game in a receding horizon fashion. Receding horizon control has a long history of use in motion planning [104, 111, 46]. Adversarial robust motion planning approaches typically assume that non-ego agents persistently exhibit worst-case behavior, while the cooperative game framework assumes all players wish to avoid collision. By contrast, our paradigm interpolates smoothly between these two extremes, generating trajectories whose conservativeness varies depending on the length of the adversarial time horizon. This is demonstrated in Section 4.5, where we compare the performance of receding horizon and non-receding horizon simulations of several traffic scenarios. We shall study the implications on robustness in future work.

4.4 Implementation Details: ILQGames

The traffic simulations in this work are solved approximately to local feedback Nash equilibria in real time using ILQGames, a recently developed, open-source C++ based game solver algorithm introduced in Fridovich-Keil et al. [46]. ILQGames iteratively solves linear-quadratic games in a receding horizon manner, by linearizing dynamics and quadraticizing costs at each step, and incurs computational complexity that is cubic in the number of players [46].

As discussed above, we must also account for both equality and inequality constraints on the game trajectory. While we note that [46] does not address constrained Nash games, here we incorporate constraints using augmented Lagrangian methods [80]. For a more detailed discussion of constraint-handling in feedback Nash games, please refer to Laine et al. [63]. Although other game solvers, such as ALGAMES [23] and Iterative Best Response algorithms [111], can likewise handle constraints, their applications are restricted to open-loop games. A thorough treatment of constraints in games can be found in Peng et al. [85].

4.5 Results

We present simulation results for various traffic scenarios in which a responsible traffic participant would likely drive defensively. First, we consider a simple situation involving oncoming vehicles on a straight road, as a proof of concept. Then, we analyze a more complicated intersection example with a crosswalk. In both cases, the ILQGames algorithm solves the defensive driving game quickly, in under 1 s.

Oncoming Example

In this example, the ego car is traveling North on a straight road when it encounters another car traveling South. Since the road has a lane in each direction, “ideally” the ego vehicle

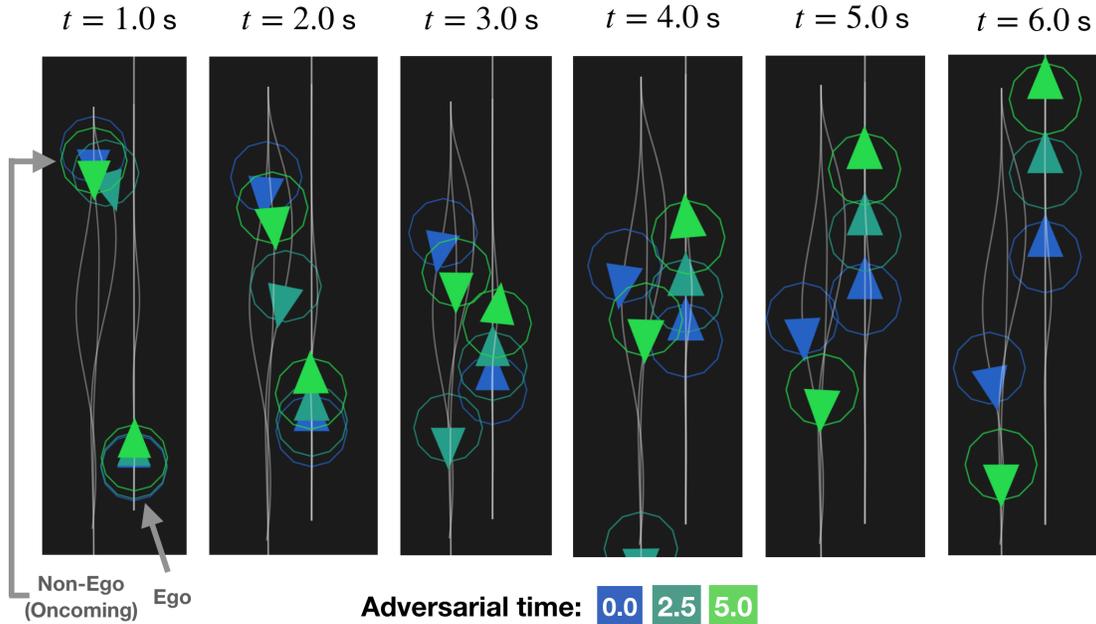


Figure 4.3: **Oncoming example.** The ego vehicle (right lane, heading upwards) and the oncoming vehicle (left lane, heading downwards) perform increasingly extreme maneuvers as T_{adv} increases, in this “oncoming” scenario. Dark blue, turquoise, and light green are used to represent the agents’ location for $T_{adv} = 0, 2.5, 5$ s, respectively. Panels show agent positions as time elapses. When $T_{adv} = 0$ s, the ego vehicle does not deviate significantly from its lane from $t = 1$ to 6 s because it anticipates that the non-ego vehicle will behave cooperatively throughout the entire time horizon by swerving to avoid a collision. However, when $T_{adv} = 5$ s, the ego vehicle actively swerves outward to avoid the non-ego agent. This minor course adjustment is sufficient to dissuade the oncoming vehicle from making a stronger effort to attempt a collision, since it would occur in the future, after T_{adv} .

would not deviate too far from its lane or speed. However, to drive more defensively, the ego vehicle should plan as though the oncoming Southbound car were to act noncooperatively. Our method encodes precisely this type of defensive planning. Figure 4.3 shows the planned trajectories that emerge for increasing T_{adv} . As shown, the ego vehicle (bottom) imagines more aggressive maneuvers for itself and the oncoming car (top) as T_{adv} increases. Note, however, that these are merely *imagined* trajectories and that (a) the ego vehicle can always choose to follow this trajectory only for an initial period of time, and recompute its trajectory thereafter, and (b) the oncoming vehicle will make its own decisions and will *not* generally follow this “partially adversarial” trajectory. We solve each of these problems (with fixed T_{adv}) in under 0.5 s.

Merging Example

Here, we present a six-player example in which the ego car is positioned on the merging lane on a highway, slightly behind the point at which an on ramp merges into the lane. The ego vehicle, with initial position at $(x, y) = (0, -20)$ m, must successfully navigate the highway while avoid collision with five non-ego vehicles—two on the passing lane to the left of the ego, two on the ramp (P5, P6), about to merge into the ego’s lane (P2, P4), and one directly in front of the ego (P3).

Figure 4.4 shows the resulting approximate local Nash equilibria of the defensive driving game for increasing T_{adv} . As before, the “apprehension” of the ego vehicle towards other vehicles increases with the length of the adversarial time horizon. In the ego’s mind, as T_{adv} increases, P6 speeds up and swerves rightwards in an attempt to collide into P1. Interestingly, P2 and P4, whose initial positions are in front of that of P1, actually slow down. This is because, although P2 and P4 are motivated to collide into P1, they are subject to heading and lane costs; thus, the best maneuver they can execute to approach and collide into P1 is to slow down.

Next, we explore the case in which the initial position of P6 were slightly behind or ahead that of P1 (Figure 4.5), with T_{adv} fixed at 2 seconds. The figure on the right here is the same as the figure on the right in Figure 4.4, with P6 behind P1. However, on the left that when P6 is initialized ahead of P1, the ego P1 does not swerve that severely away from P6, but instead decides to make a more significant attempt to avoid P3 and P4 at the intersection of the ramp and the highway lanes.

Three-Player Intersection Example

We introduce a more complicated scenario designed to model the behavior of two vehicles and a pedestrian at an intersection. As shown in Figure 4.6, the ego vehicle is present in the intersection alongside a non-ego vehicle heading in the opposite direction, who wishes to make a left turn, and a pedestrian, who wishes to cross the road. To reach their goal locations, these three agents must cross paths in the intersection. When $T_{adv} = 0$ s, the ego vehicle continues straight along its lane because it anticipates that the non-ego vehicle will behave cooperatively throughout the entire time horizon. In particular, it anticipates that the non-ego vehicle will continue along its curved path at nominal speed, resulting in a collision-free trajectory. However, as with the oncoming example, the ego vehicle’s trajectory becomes increasingly more conservative as the adversarial time horizon increases in length. When $T_{adv} = 1$ s, the ego vehicle actively swerves rightwards to avoid the non-ego vehicle. This is because in this scenario, the oncoming vehicle is initially slower than the ego vehicle, and will thus approach the intersection at the same time as the ego vehicle. As before, each problem is solved in well under 0.75 s in single-threaded operation on a standard laptop, via the ILQGames algorithm [46]. This performance indicates real-time capabilities which will be explored in future work on hardware.

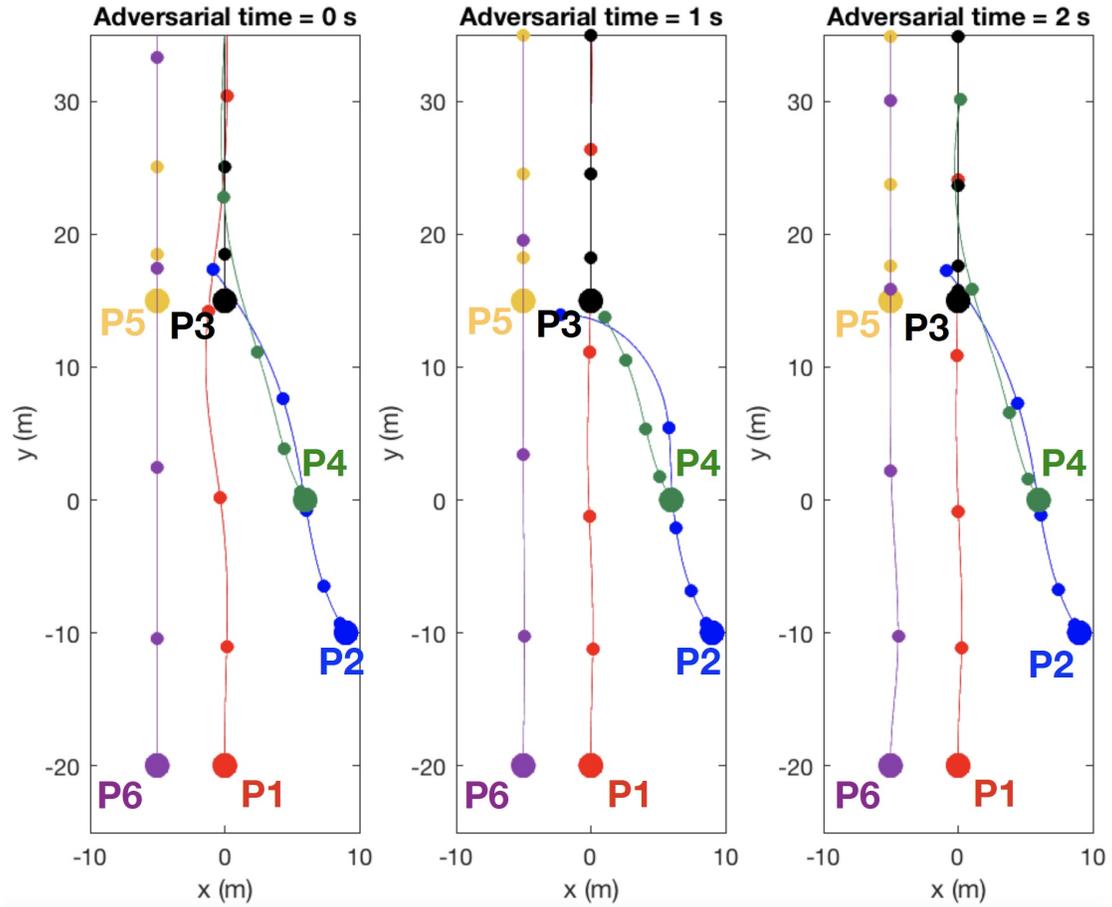


Figure 4.4: **Merging Example.** The ego vehicle (P1, shown in red, heading upwards) is traveling along a highway while avoiding collisions with other vehicles (P2-P5, shown in various colors), three of which are merging or have just merged onto the highway. Notice that the ego vehicle’s apprehension towards the other vehicles increases when the adversarial sub-interval increases. In particular, as T_{adv} increases, P6, whose initial position is behind that of P1, speeds up and swerve rightwards in an attempt to collide into P1. Meanwhile, P2 and P4, whose initial positions are in front of that of P1, actually slow down.

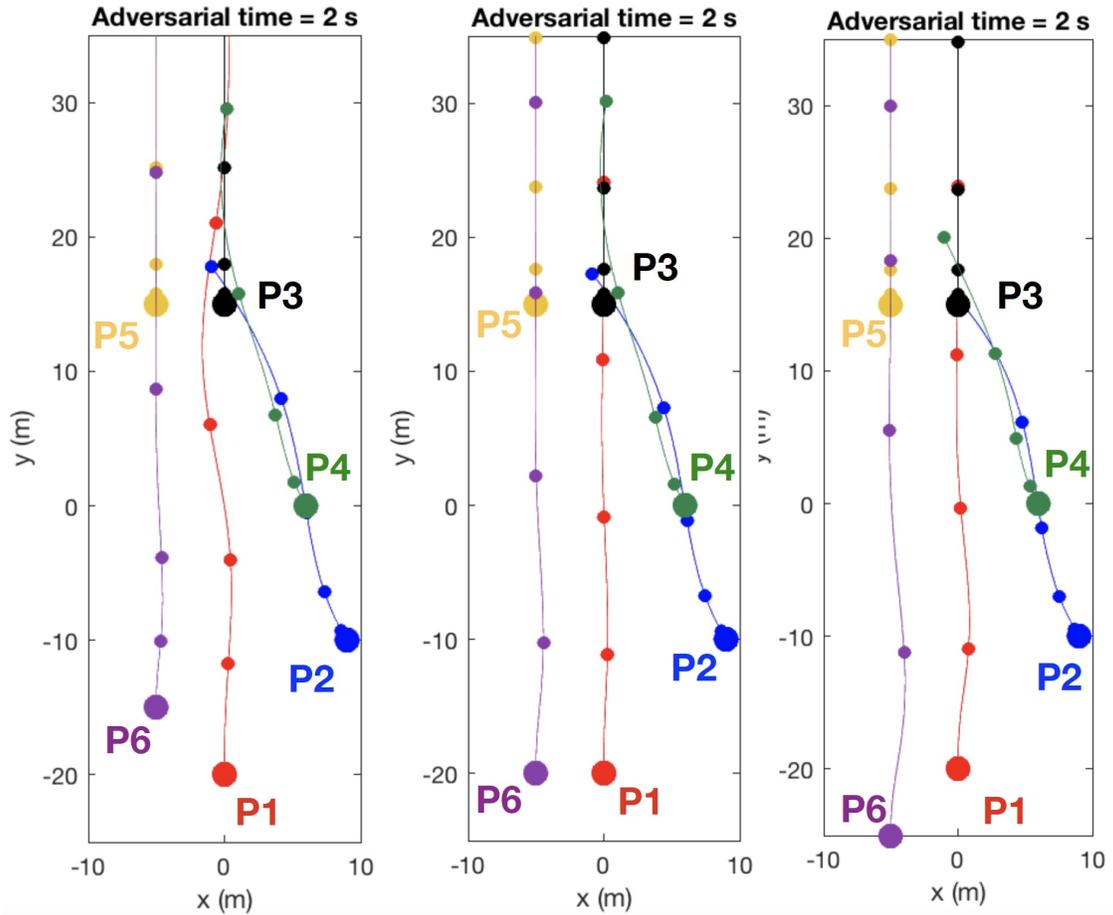


Figure 4.5: **Merging Example.** Consider again the merging example first depicted in Figure 4.4, with the same color code, and with the same fixed $T_{adv} = 2$ s. Here, we examine what happens if the initial position of P6 were slightly behind or ahead that of P1. Note that the figure on the right here is identical to the figure on the right in Figure 4.4, with P6 behind P1. On the left, however, when P6 is initialized ahead of P1, the ego P1 does not swerve that severely away from P6, but instead decides to make a more significant attempt to avoid P3 and P4 at the intersection of the ramp and the highway lanes.

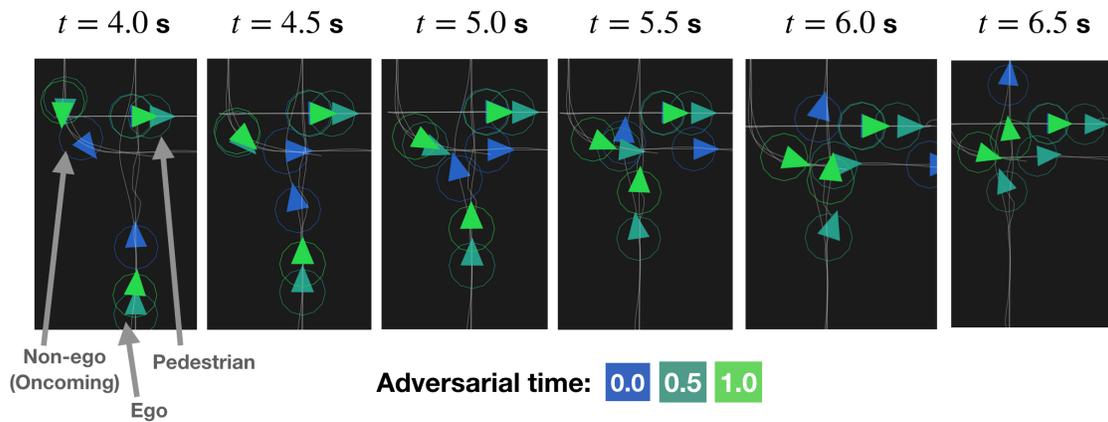


Figure 4.6: **Three Player Intersection Example.** The ego vehicle (right lane, heading upwards) navigates an intersection while avoiding collision with an oncoming vehicle (left lane, heading downwards initially before making a left turn) and a pedestrian (horizontal path at the intersection, left to right). Dark blue, turquoise, and light green are used to represent the agents’ location at $T_{adv} = 0, 0.5, 1$ s, respectively. As with the oncoming example, these three agents perform increasingly extreme maneuvers as T_{adv} increases. In particular, when $T_{adv} = 0$ s, the ego vehicle anticipates that the non-ego will continue along its curved path at its nominal speed, allowing it to approach the intersection before the ego vehicle. Thus, the ego vehicle swerves leftwards, to avoid the non-ego agent as it makes its left turn and continues rightwards in the figure, resulting in a collision-free trajectory. (The pedestrian’s trajectories for $T_{adv} = 0$ s and $T_{adv} = 0.5$ s coincide.) However, when $T_{adv} = 1$ s, the ego vehicle accelerates and swerves rightwards to avoid the non-ego vehicle. As before, this suffices to dissuade the oncoming vehicle from performing more collision-seeking behavior, since it forces any potential collision to occur after T_{adv} . The pedestrian’s speed also noticeably decreases for the same reason.

4.6 Discussion

This chapter presented a novel formulation of robustness in motion planning for multi-agent problems. Inspired by defensive driving, our method explicitly models other agents as adversarial in only a limited, initial portion of the overall planning interval. Instead of forcing the ego to avoid all feasible non-ego trajectories, we use a piecewise-defined game cost structure to endow the ego with the perspective that other agents are temporarily distracted. As such, our approach generates far less conservative behavior than purely adversarial methods, such as Hamilton-Jacobi-Isaacs optimal control. Simulation results illustrate that this novel formulation of safety can be used to solve these “defensive” problems in real-time. We are eager to implement this method in hardware and test its performance in a receding time horizon with other (human) drivers.

Chapter 5

Game-Theoretic Priors for SLAM

To navigate through real-world traffic safely and efficiently, autonomous vehicles must monitor their surroundings for obstacles or other disturbance sources, while simultaneously interacting with multiple self-interested agents in the scene. This entails accurately representing an uncharted and possibly dynamic environment, while executing robust motion plans that account for multi-player interactions in the scene. This requires a careful fusion of state estimation, prediction, and path planning modules in vehicle autonomy stacks. However, current, state-of-the-art autonomous navigation pipelines typically treat these problems separately and do not incorporate direct feedback between them.

Regarding estimation, algorithms tackling the Simultaneous Localization and Mapping (SLAM) problem aim to accurately reconstruct an uncharted environment while also localizing the “ego” player within it [66, 14, 25, 26]. In recent years, popular inference algorithms for SLAM, such as factor graph-based methods, have also been used to solve challenging motion planning and optimal control problems jointly with SLAM problems in unknown environments [27]. However, these approaches often do not account for the behavior of self-interested dynamic players in the vicinity. It is therefore an open question whether these approaches can truly generate safe, efficient, and robust motion plans for an ego player operating in the vicinity of other players.

Interactions between individually self-interested mobile agents are naturally modeled as dynamic games between rational actors with differing objectives [53, 5, 100, 99]. Recent advances in game-theoretic motion planning exploit this structure to predict the responses of other players to one’s own decisions, and identify a desirable equilibrium strategy [38, 46]. However, game-theoretic formulations of noncooperative multi-player interactions are generally not considered in SLAM tasks.

In this chapter, we formulate the SLAM task from the perspective of an ego player, who is interacting with multiple other players while simultaneously estimating all players’ positions and all landmark locations. Inspired by the dynamic game theory literature, we first establish mild assumptions under which this problem can be formulated as a potential game. We then present a factor graph-based algorithm to solve this game and prove that it is guaranteed to converge to a local equilibrium. Unlike existing SLAM methods, this

approach tightly integrates estimation, prediction, and decision-making *for multiple players, simultaneously*. Experiment results illustrate that, compared to standard bundle adjustment, the introduction of game-theoretic interaction priors leads to higher localization and map reconstruction accuracy in realistic traffic scenarios. Moreover, our algorithm is robust with respect, both to perturbations in the distributions of initial conditions, and across a wide range of observation noise.

The rest of this chapter is structured as follows. Section 5.1 surveys related work on factor-graph based algorithms for SLAM, dynamic SLAM, and game-theoretic path planning. Section 5.2 introduces fundamental concepts in dynamic game theory relevant to the multi-player interactions explored by our SLAM algorithm. Section 5.3 then formulates the SLAM problem under study as a potential game, and presents our IBR-based algorithm for solving it. The empirical results presented in Section 5.4 demonstrate that our algorithm outperforms standard bundle adjustment algorithms in localization and map reconstruction accuracy in realistic traffic scenarios. Finally, Section 5.5 summarizes our method and presents future research directions. More details can be found in a published paper, Chiu and Fridovich-Keil “GTP-SLAM: Game-Theoretic Priors for Simultaneous Localization and Mapping in Multi-Agent Scenarios” [18].

5.1 Related Work

Simultaneous Localization and Mapping

Simultaneous Localization and Mapping is a fundamental state estimation task with a well-developed literature in the robotics community [66, 14]. A standard method for solving SLAM problems is to reformulate the underlying maximum a posteriori (MAP) estimation problem into a nonlinear least squares problem, which can then be solved via factor graph optimization [30, 56].

In recent years, factor graphs have been used to formulate a wide range of robotics problems beyond the SLAM task in static environments, including model predictive control and trajectory tracking [102, 27]. Factor graph-based methods have also been used to solve the dynamic SLAM problem, which involves the reconstruction of uncharted environments with dynamic players [115] who share measurement information, and perform estimation, and prediction in a cooperative game framework [116, 51]. These methods typically infer time-dependent variables pertaining to multiple players without accounting for players’ interactive, and likely noncooperative, behavior [60, 103]. By contrast, our approach explicitly accounts for purposeful and potentially noncooperative interactions between multiple players by using iterative best response to search for local Nash equilibria of the players’ variables.

Multi-Player Path Planning via Dynamic Games

In robotics applications, interactions between multiple players are naturally modeled as dynamic games. In particular, scenarios in which two groups of players have opposing objectives, such as robust control problems and pursuit-evasion games, are often formulated as zero-sum dynamic games [38, 39]. Meanwhile, problems in which multiple players have only partially conflicting objectives, such as path planning in busy traffic, are posed as general-sum dynamic games [46, 87]. Although solutions to continuous-time dynamic games are characterized by coupled Hamilton-Jacobi-Bellman (HJB) PDEs [100, 99, 4], solving these equations is typically intractable due to the so-called “curse of dimensionality,” [7] i.e., their computation time grows exponentially in the state space dimension. For this reason, such methods are impractical in many multi-player scenarios of interest.

In contrast, our algorithm uses an iterative best response (IBR) scheme, in which each player takes a turn solving for their optimal strategy while assuming all other players’ strategies are fixed [40, 110, 59, 114]. By replacing the dynamic game with a sequence of optimal control problems, the computational burden of solving for a local Nash equilibrium strategy is substantially reduced. Indeed, IBR has been successfully applied to a wide range of multi-player interaction scenarios, such as hierarchical planning for autonomous driving [40] and racing [110]. Moreover, IBR is suitable for our application because it can be embedded in a factor graph-based framework, by iteratively solving estimation problems over the variables relevant to each player, while holding all other players’ variables fixed.

Our approach also draws inspiration from the potential games literature, which exploits the symmetric cost structure of multi-player interactions in certain robotics applications [59, 114]. Recent literature indicates that iterative methods which exploit this symmetry can be more efficient than those that do not [59]. Our approach draws inspiration from this observation: we recast the multi-agent SLAM problem under study as a potential game, and perform IBR in a manner that preserves its potential structure.

5.2 Setup and Notation

Below, we introduce core concepts in dynamic game theory. Readers are directed to [5] for further details.

Open-Loop Nash Equilibria

Consider the N -player, K -stage general-sum dynamic game Γ , with nonlinear, discrete-time system dynamics for each player $i \in [N] := \{1, \dots, N\}$ and time $k \in [K] := \{1, \dots, K\}$, given by:

$$x_{k+1}^i \sim \mathcal{N}(f_k^i(x_k^i, u_k^i), \Sigma_f), \quad (5.1)$$

where $x_k^i \in \mathbb{R}^{n_i}$, $u_k^i \in \mathbb{R}^{m_i}$, and $f_k^i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \rightarrow \mathbb{R}^{n_i}$ are respectively the state, control input, and (differentiable) state transition map of player i at time k , and Σ_f is the associated co-

variance matrix. Below, for each player $i \in [N]$, we use the shorthand $x^i := (x_1^i, \dots, x_{K+1}^i) \in \mathbb{R}^{n_i(K+1)}$, and $u^i := (u_1^i, \dots, u_K^i) \in \mathbb{R}^{m_i K}$. Moreover, we define $x := (x^1, \dots, x^N) \in \mathbb{R}^{n(K+1)}$ and $u := (u^1, \dots, u^N) \in \mathbb{R}^{mK}$, where $n = \sum_i n_i$ and $m = \sum_i m_i$.

Our method jointly estimates the trajectories of all players from the perspective of one particular player, referred to below as player 1 or the *ego player*. Other players are termed *non-ego players*. The ego player observes N_ℓ landmarks, whose global positions in \mathbb{R}^{d_ℓ} are given by $\ell := (\ell_1, \dots, \ell_{N_\ell}) \in \mathbb{R}^{d_\ell N_\ell}$. Players also observe each others' positions. These measurements, at each time $k \in [K]$, are given by:

$$z_k^\alpha \sim \mathcal{N}(h(x_k^1, \ell_\alpha), \Sigma_h), \quad (\text{landmark-agent}) \quad (5.2)$$

$$\bar{z}_k^{ij} \sim \mathcal{N}(\bar{h}(x_k^i, x_k^j), \Sigma_{\bar{h}}), \quad (\text{inter-agent}) \quad (5.3)$$

where $z_k^\alpha \in \mathbb{R}^z$ is the measurement of landmark α by the ego player, for each $\alpha \in [N_\ell]$, while $\bar{z}_k^{ij} \in \mathbb{R}^z$ is the measurement by player i of player j , for each $i, j \in [N]$, $i \neq j$. Here, Σ_h and $\Sigma_{\bar{h}}$ denote associated covariance matrices. Additionally, each player's objective is defined by $L^i(x, u)$, with $L^i : \mathbb{R}^{n(K+1)} \times \mathbb{R}^{mK} \rightarrow \mathbb{R}$ for each $i \in [N]$, $k \in [K]$. Here, we presume that the ego player knows other players' objectives L^i . While this seems a strong assumption in practice, recent work has established that it is possible to infer unknown parameters of players' objectives in such games efficiently [87, 65]. Thus equipped, we now define the *Nash equilibrium* of the GTP-SLAM problem.

Definition 5.2.1. [*Open-Loop Nash Equilibrium*, [5, Ch. 6]] We call $u^* := (u^{1,*}, \dots, u^{N,*})$ an open-loop Nash equilibrium of Γ if no player can lower their cost by unilaterally deviating from their control $u^{i,*}$ while all other players' controls, $u^{-i,*}$, remains fixed, i.e.,

$$L^i(u^{i,*}, u^{-i,*}) \leq L^i(u^i, u^{-i,*}), \quad \forall u^i \in \mathbb{R}^{m_i K}. \quad (5.4)$$

Potential Dynamic Games

Our approach leverages well-established convergence guarantees of iterative best response (IBR) algorithms in the setting of potential games. For clarity, we define a finite-stage potential game as follows.

Definition 5.2.2 (Potential Dynamic Game, [59], [42]). An N -player, K -stage general-sum dynamic game Γ is called a **potential game** if there exists an optimal control problem, defined over all players' controls (u^1, \dots, u^N) , whose solutions are Nash equilibria of the game Γ . In other words, there exists a **potential function** $p : \mathbb{R}^{d_u N K} \rightarrow \mathbb{R}$ such that, for any player $i \in [N]$ and any controls $(u^1, \dots, u^N) \in \mathbb{R}^{d_u N K}$ and any alternative player i controls \bar{u}^i , we have:

$$p(u^i, u^{-i}) - p(\bar{u}^i, u^{-i}) = L^i(u^i, u^{-i}) - L^i(\bar{u}^i, u^{-i}).$$

In Section 5.3, we will recast the multi-player, noncooperative SLAM problem of interest into a potential game, and establish mild assumptions under which an appropriate IBR algorithm converges.

5.3 Methods

Our main contribution is GTP-SLAM, a novel SLAM algorithm for multi-player scenes, motivated by iterative best response. GTP-SLAM aims to jointly estimate the dynamic states and control inputs of all players in the scene, as well as landmark positions. It does so from the ego player’s perspective, while accounting for noncooperative, game-theoretic interactions between the players.

Constructing the GTP-SLAM Factor Graph

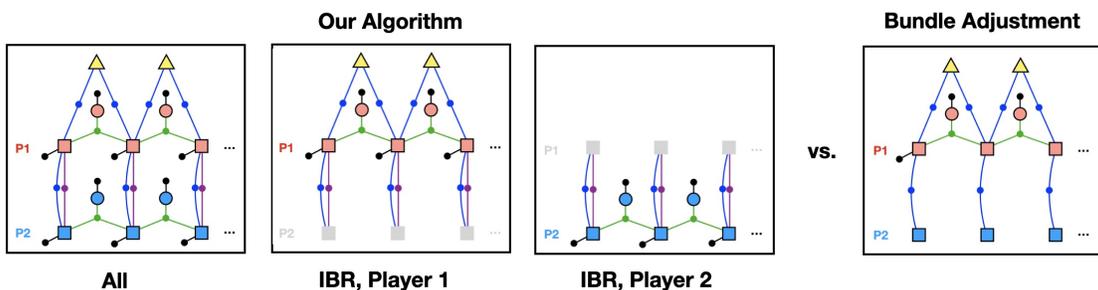


Figure 5.1: Factor graphs for GTP-SLAM, our IBR-based algorithm (Left) and a standard bundle adjustment approach (Right), for a two-player example. Red nodes describe dynamic variables (states x , controls u) for Player 1, while blue nodes describe analogous quantities for Player 2. Gray nodes represent variables temporarily held constant. Square, circular, and triangular nodes represent states, controls, and landmarks, respectively. Green factors (lines and nodes) represent dynamics constraints, blue factors represent landmark and inter-player distance measurements, and black factors represent priors on states and controls, e.g. to track lanes, to align the vehicle headings, and minimize overall control effort.

We begin by expressing the players’ noncooperative preferences as factors in a bipartite factor graph. Each factor is a function which encodes vector-valued *residual error* among the connected variables; as a whole, factors allow us to compute the joint likelihood of all input variables. Following standard Gaussian assumptions, we use the Mahalanobis distance associated to each factor (i.e., $\|v\|_{\Sigma}^2 := v^{\top} \Sigma^{-1} v$ for factor v and covariance Σ) to compute the negative log-likelihood of a collection of variables. Concretely, then, we construct a factor graph from the following terms:

$$\text{priors} : g_k^i(x_k^i), \hat{g}_k^i(u_k^i) \quad (5.5)$$

$$\text{dynamics} : f_k^i(x_k^i, u_k^i) - x_{k+1}^i \quad (5.6)$$

$$\text{interactions} : b(x_k^i, x_k^j) \quad (5.7)$$

$$\text{landmark measurements} : h(x_k^1, \ell_{\alpha}) - z_k^{\alpha} \quad (5.8)$$

$$\text{inter-player observations : } \bar{h}(x_k^i, x_k^j) - \bar{z}_k^{ij}, \quad (5.9)$$

which are color-coded in Figure 5.1. For example, the ternary dynamics factor (5.6) computes the difference between vehicles' states and those predicted by the appropriate state transition function (5.1). Likewise, the factors (5.7) between pairs of states belonging to players $i \neq j$ describe interactions between pairs of players. For example, to encode collision avoidance, we may set $b(x, x') := 1/\|x - x'\|_2$. The factors $h(x_k^1, \ell_\alpha) - z_k^\alpha$ denote the difference between expected and actual landmark measurements made by the ego player. Finally, $\bar{h}(x_k^i, x_k^j) - \bar{z}_k^{ij}$, where $i, j \in [N], i \neq j, \min\{i, j\} = 1$, denotes inter-player position measurements between the ego player and each non-ego player.

The maximum a posteriori (MAP) estimation problem faced by each player, then, is the minimization of a sum of squared factors. In other words, each player's individual decision problem is a nonlinear least squares problem, when other players' variables are held constant. Neglecting interaction factors, landmarks, and inter-player measurements (which couple players' variables together), we compute the partial log-likelihood of each player i 's variables as:

$$J^i(x^i, u^i) := \sum_{k=1}^K \left(\|g_k^i(x_k^i)\|_{\Sigma_g}^2 + \|\hat{g}_k^i(u_k^i)\|_{\Sigma_{\hat{g}}}^2 + \|f_k^i(x_k^i, u_k^i) - x_{k+1}^i\|_{\Sigma_f}^2 \right). \quad (5.10)$$

Note that (5.10) does not include interaction terms (b) or measurements (z or \bar{z}), since each interaction term depends upon multiple players' variables, and because measurements pertaining to landmarks and other players' states are only assumed to be collected by the ego player. If these terms are included, the ego player's full estimation problem is given by:

$$\begin{aligned} L^1(x, u, \ell) := & J^1(x, u) + \sum_{\alpha=1}^{N_\ell} \sum_{k=1}^K \|h(x_k^1, \ell_\alpha) - z_k^\alpha\|_{\Sigma_h}^2 \\ & + \sum_{j \neq 1} \sum_{k=1}^K \|\bar{h}(x_k^1, x_k^j) - \bar{z}_k^{1j}\|_{\Sigma_{\bar{h}}}^2 + \sum_{j \neq 1} \sum_{k=1}^K \|b(x_k^1, x_k^j)\|_{\Sigma_b}^2, \end{aligned} \quad (5.11)$$

while each non-ego player's MAP problem is given by:

$$\begin{aligned} L^i(x, u) := & J^i(x^i, u^i) + \sum_{k=1}^K \|\bar{h}(x_k^i, x_k^1) - \bar{z}_k^{i1}\|_{\Sigma_{\bar{h}}}^2 \\ & + \sum_{j \neq i} \sum_{k=1}^K \|b(x_k^i, x_k^j)\|_{\Sigma_b}^2, \end{aligned} \quad (5.12)$$

for each $i \in [N] \setminus \{1\}$.

GTP-SLAM as a Potential Game

Next, we illustrate that the GTP-SLAM problem of Section 5.3 is a potential game (Lemma 5.3.1, Proposition 5.3.2). This connection to potential games is critical, as it suggests a locally-convergent solution method for GTP-SLAM problems given in Section 5.3 (Corollary 5.3.3). The following results are based upon established concepts in the literature [42, 114, 59]; here, we illustrate their pertinence to the noncooperative SLAM problem. The following is a direct corollary of [59], Theorems 1 and 2.

Lemma 5.3.1. *Consider an N -player, K -stage dynamic game Γ , with fixed initial condition $x_1 := (x_1^1, \dots, x_1^N) \in \mathbb{R}^n$. Suppose the system dynamics of each player given by (5.1), and the cost function of each player i is of the form:*

$$L^1(x, u, \ell) := C^1(x^1, u^1, \ell) + \sum_{j=2}^N C^{1j}(x^1, x^j),$$

$$L^i(x, u) := C^i(x^i, u^i) + \sum_{j \neq i} C^{ij}(x^i, x^j), \quad \forall i \in [N] \setminus \{1\}$$

respectively, where $C^i : \mathbb{R}^{n_i(K+1)} \times \mathbb{R}^{m_i K} \rightarrow \mathbb{R}$, and $C^{ij} : \mathbb{R}^{n_i(K+1)} \times \mathbb{R}^{n_j(K+1)} \rightarrow \mathbb{R}$ satisfy:

$$C^{ij}(x^i, x^j) = C^{ji}(x^j, x^i), \quad \forall i, j \in [N], i \neq j.$$

Then Γ is a potential game corresponding to the optimal control problem of minimizing the potential function:

$$p(x, u) := C^1(x^1, u^1) + \sum_{i=2}^N C^i(x^i, u^i) + \sum_{i, j \in [N], i < j} C^{ij}(x^i, x^j), \quad (5.13)$$

subject to the dynamics (5.1), for each $i \in [N]$.

Note that L^1 and C^1 also depend upon landmarks ℓ ; we make this dependence implicit for notational convenience.

Proof. This proposition follows directly from analogous proofs established in [59, 49], rephrased here for completeness.

Below, we make implicit the dependence of L^1 and C^1 on the landmarks ℓ , for notational convenience. Let $p(x, u)$ be given by (5.13), with a minimizer given by $u^* := (u^{1,*}, \dots, u^{N,*}) \in \mathbb{R}^{mK}$, $x^* := (x^{1,*}, \dots, x^{N,*}) \in \mathbb{R}^{n(K+1)}$. For any player $r \in [N]$, and any unilateral deviation in player r 's controls away from u^* , i.e.,

$$u := (u^r, u^{-r,*}) = (u^{1,*}, \dots, u^{r-1,*}, u^r, u^{r+1,*}, \dots, u^{N,*}) \in \mathbb{R}^{mK},$$

with corresponding state trajectory:

$$(x^r, x^{-r,*}) = (x^{1,*}, \dots, x^{r-1,*}, x^r, x^{r+1,*}, \dots, x^{N,*}) \in \mathbb{R}^{n(K+1)},$$

we have:

$$\begin{aligned}
0 &\leq p(x^r, x^{-r,*}, u^r, u^{-r,*}) - p(x^{r,*}, x^{-r,*}, u^{r,*}, u^{-r,*}) \\
&= \left(C^r(x^r, u^r) + \sum_{i \neq r} C^{ir}(x^{i,*}, x^r) + \sum_{i \neq r} C^i(x^{i,*}, u^{i,*}) + \sum_{\substack{1 \leq i < j \leq n \\ i, j \neq r}} C^{ij}(x^{i,*}, x^{j,*}) \right) \\
&\quad - \left(C^r(x^{r,*}, u^{r,*}) + \sum_{i \neq r} C^{ir}(x^{i,*}, x^{r,*}) + \sum_{i \neq r} C^i(x^{i,*}, u^{i,*}) + \sum_{\substack{1 \leq i < j \leq n \\ i, j \neq r}} C^{ij}(x^{i,*}, x^{j,*}) \right) \\
&= \left(C^r(x^r, u^r) + \sum_{i \neq r} C^{ir}(x^{i,*}, x^r) \right) - \left(C^r(x^{r,*}, u^{r,*}) + \sum_{i \neq r} C^{ir}(x^{i,*}, x^{r,*}) \right) \\
&= L^r(x^r, x^{-r,*}, u^r, u^{-r,*}) - L^r(x^{r,*}, u^{r,*}).
\end{aligned}$$

This condition matches that which defines open-loop Nash equilibria (Definition 5.2.1). Hence, u^* is an open-loop Nash equilibrium of the game Γ . \square

Given the result of Lemma 5.3.1, we now show that the GTP-SLAM game structure given in (5.10) is consistent with a potential game.

Proposition 5.3.2. *The GTP-SLAM game, with players' objectives given by (5.10), is a potential game.*

Proof. In the context of (5.10), we have:

$$\begin{aligned}
C^1(x^1, u^1, \ell) &:= J^1(x^1, u^1) + \sum_{\alpha=1}^{N_\ell} \sum_{k=1}^K \|h(x_k^1, \ell_\alpha) - z_k^\alpha\|_{\Sigma_h}^2, \\
C^{1j}(x^1, x^j) &:= \sum_{k=1}^K \|\bar{h}(x_k^1, x_k^j) - \bar{z}_k^{1j}\|_{\Sigma_{\bar{h}}}^2 + \sum_{k=1}^K \|b(x_k^1, x_k^j)\|_{\Sigma_b}^2, \quad \forall j \in [N] \setminus \{1\}, \\
C^i(x^i, u^i) &:= J^i(x^i, u^i), \quad \forall i \in [N] \setminus \{1\}, \\
C^{i1}(x^i, x^1) &:= \sum_{k=1}^K \|\bar{h}(x_k^i, x_k^1) - \bar{z}_k^{i1}\|_{\Sigma_{\bar{h}}}^2 + \|b(x_k^i, x_k^1)\|_{\Sigma_b}^2, \quad \forall i \in [N] \setminus \{1\}, \\
C^{ij}(x^i, x^j) &:= \sum_{k=1}^K \|b(x_k^i, x_k^j)\|_{\Sigma_b}^2, \quad \forall i, j \in [N] \setminus \{1\}, i \neq j.
\end{aligned}$$

Thus, by Lemma 5.3.1, the game encoded in GTP-SLAM is a potential dynamic game. \square

Iterative Best Response

To find Nash equilibria of the GTP-SLAM game with objectives given by (5.11) and (5.12), we employ Algorithm 13, an approach inspired by iterative best response (IBR). Specifically, Algorithm 13 proceeds in rounds, where each player i minimizes its MAP objective L^i while holding variables pertaining to other players $j \neq i$ fixed. Convergence is guaranteed by the following corollary, due to [114].

Corollary 5.3.3. *Algorithm 13 converges to an open-loop Nash equilibrium when applied to potential games of the form of Definition 5.2.2, if the maximum number of iterations Q is set to ∞ and the convergence tolerance ϵ is set to 0.*

Proof. The proof of this Corollary follows directly from Zanardi et al. Proposition 1 [114]. \square

Algorithm 13: Solving the GTP-SLAM Problem

Data: Maximum number of iterations Q , Convergence tolerance ϵ , Cost functions L^i .

Result: Nash equilibrium variables $x^* = (x^{1,*}, \dots, x^{N,*}), u^* = (u^{1,*}, \dots, u^{N,*}), \ell^*$.

```

1  $q \leftarrow 1$ 
2 while  $q \leq Q$  and  $\|x^{i,q} - x^{i,q-1}\|_2 < \epsilon, \forall i \in [N]$  do
3    $(x^{1,q}, u^{1,q}, \ell^q) \leftarrow \arg \min_{(x^1, u^1, \ell)} L^1(x, u, \ell)$ 
4   for  $i \in [N] \setminus \{1\}$  do
5      $(x^{i,q}, u^{i,q}) \leftarrow \arg \min_{(x^i, u^i)} L^i(x, u)$ 
6   end
7    $q \leftarrow q + 1$ 
8 end
9 return  $(x^{1,q}, \dots, x^{N,q}), (u^{1,q}, \dots, u^{N,q}), \ell^q$ 

```

5.4 Experiment Results

Simulation Setup

To demonstrate the importance of game-theoretic priors in multi-player SLAM problems, we simulate a highway driving scenario (Figure 5.2). Specifically, four vehicles change lanes over a kilometer-long stretch of highway while avoiding collision, maintaining a desired speed, and collecting range and bearing measurements of surrounding landmarks via lidar. These preferences are encoded directly in players' individual cost functions $L^i, \forall i \in [N]$. We assume that each vehicle follows Dubins paths, i.e., moves with constant speed (30 m s^{-1} , here), and can control its yaw rate. Vehicle motion is discretized at intervals of 0.2s. These dynamics

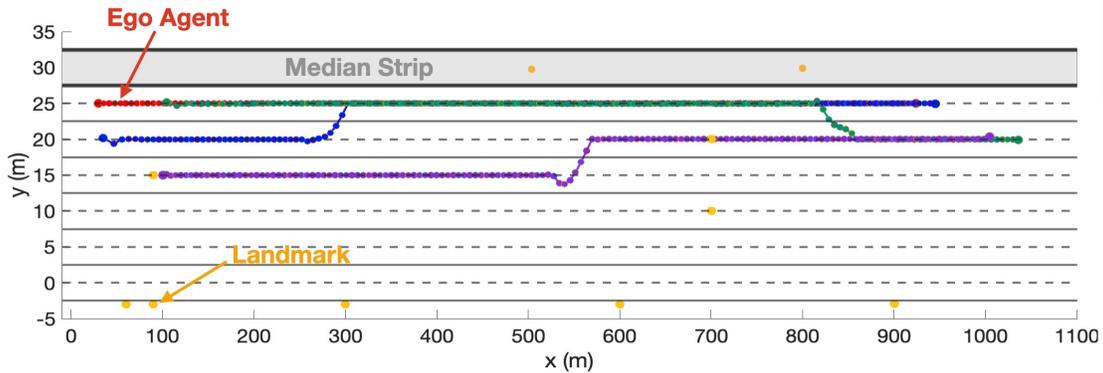


Figure 5.2: Schematic of the highway example. Here, players 1 (red), 2 (blue), 3 (green), and 4 (purple) navigate a kilometer-long stretch of highway and interact with each other while performing lane changes. The ego player detects landmarks in the scene, which describe objects common to realistic highway scenarios, e.g., speed limit signs, exit signs, light poles, etc.

constitute the state transition maps f_k^i . Moreover, we assume that the highway is sparsely populated with occasional landmarks, e.g., exit signs, speed limit signs, and light poles, shown as yellow circles in Figure 5.2.

A local Nash equilibrium of the highway driving game is found by applying Algorithm 13 with fixed initial states for all players and neglecting measurement likelihood factors. To understand the role of game-theoretic interactions in SLAM problems, we conduct a Monte Carlo study of the highway driving scenario of Figure 5.2, with results recorded in Figure 5.8. For each noise standard deviation level in the set $\{0.05, 0.10, \dots, 0.95, 1.0\}$, we ran 50 experiments, each with a slightly perturbed set of initial conditions. For each experiment, we simulated random measurements of all landmarks, and of all non-ego players' planar coordinates, with respect to the ego player's local frame. We then ran Algorithm 13 to convergence, and compared the results to a standard bundle adjustment approach that neglected game-theoretic priors. That is, by ignoring (5.7) for the ego player and (5.5), (5.6), (5.7) for all non-ego players, the GTP-SLAM problem reduces to a single MAP problem which may be solved jointly for all players at once. Throughout all simulations, we use GTSAM [28] to construct the factors above, compute Jacobians, and implement Levenberg-Marquardt steps [80] for both GTP-SLAM and bundle adjustment.

Empirical Results

Figures 5.8 and 5.5 record the total localization and map reconstruction error of GTP-SLAM (red) and standard bundle adjustment (blue). Individual plots of localization error are presented in Figures 5.6 and 5.3, while individual plots of map reconstruction error are presented

in Figures 5.7 and 5.4. Compared to the bundle adjustment baseline, the localization and map reconstruction error for GTP-SLAM is lower across all noise standard deviation levels, and degrades more gracefully as noise levels increase. In particular, conventional bundle adjustment becomes numerically unstable at low noise levels, as evinced by the spike in the root-mean-square error (0.9 m) incurred at a measurement noise standard deviation of 0.3. By contrast, the introduction of game-theoretic priors appears to yield a more well-conditioned estimation problem, resulting in lower error overall, and accuracy (characterized by the root-mean-square errors) that degrades more gracefully with the standard deviation of the measurement noise. In summary, these results indicate that game-theoretic priors introduce additional structure in an otherwise complex estimation problem, enabling reliable recovery of vehicle states and map landmarks. In particular, whereas conventional bundle adjustment struggles to reconstruct poses with later timestamps, possibly due to the corresponding lack of landmark measurements at those times, GTP-SLAM does not suffer significantly from this drawback.

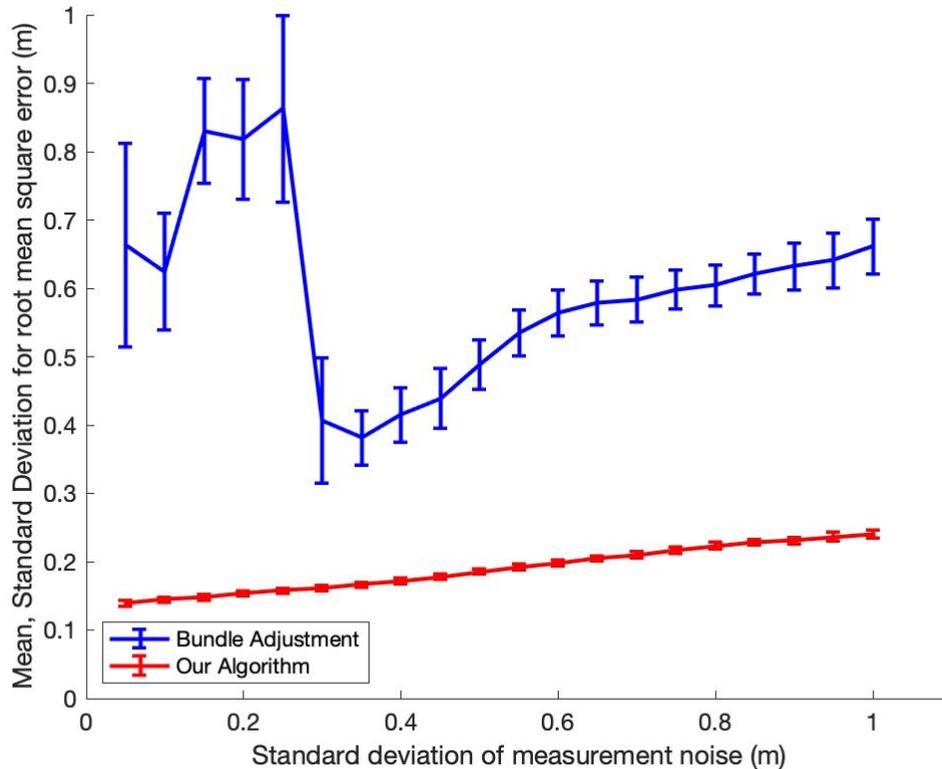


Figure 5.3: Mean and standard deviations of root-mean-square error of vehicle poses vs. standard deviation of measurement noise.

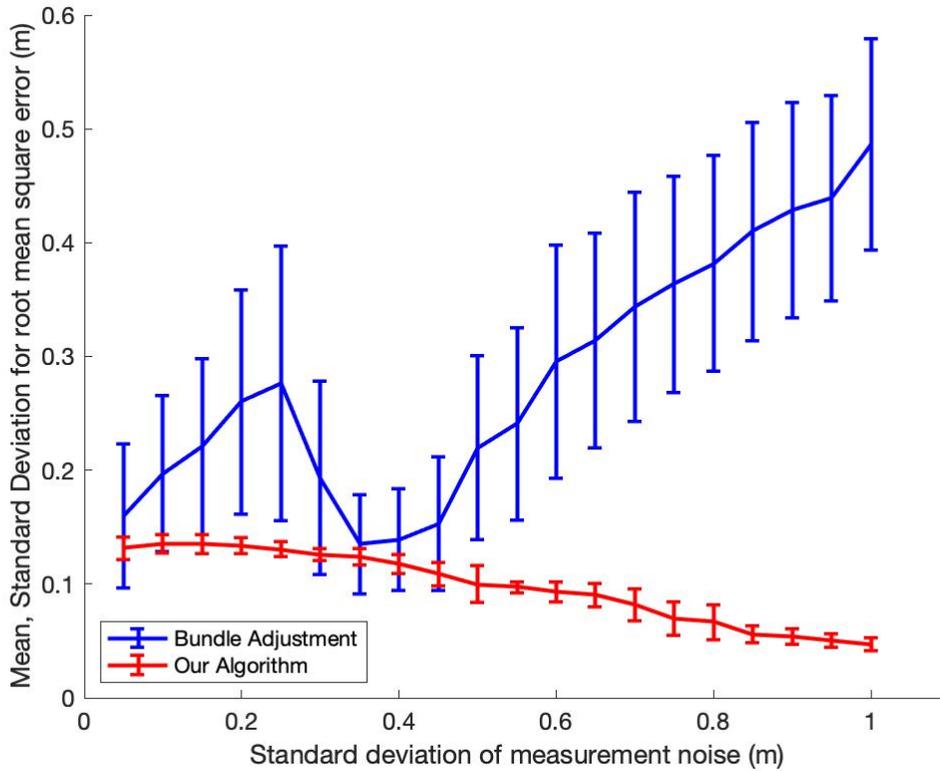


Figure 5.4: Mean and standard deviations of root-mean-square error of landmark positions vs. standard deviation of measurement noise.

5.5 Discussion

Inspired by recently developed iterative dynamic game algorithms, we present a novel method for Simultaneous Localization and Mapping (SLAM) in dynamic scenes in which multiple players interact noncooperatively. Our approach exploits the structure of potential games to ensure reliable convergence. Empirical results illustrate that our algorithm outperforms standard bundle adjustment methods in localization and map reconstruction accuracy.

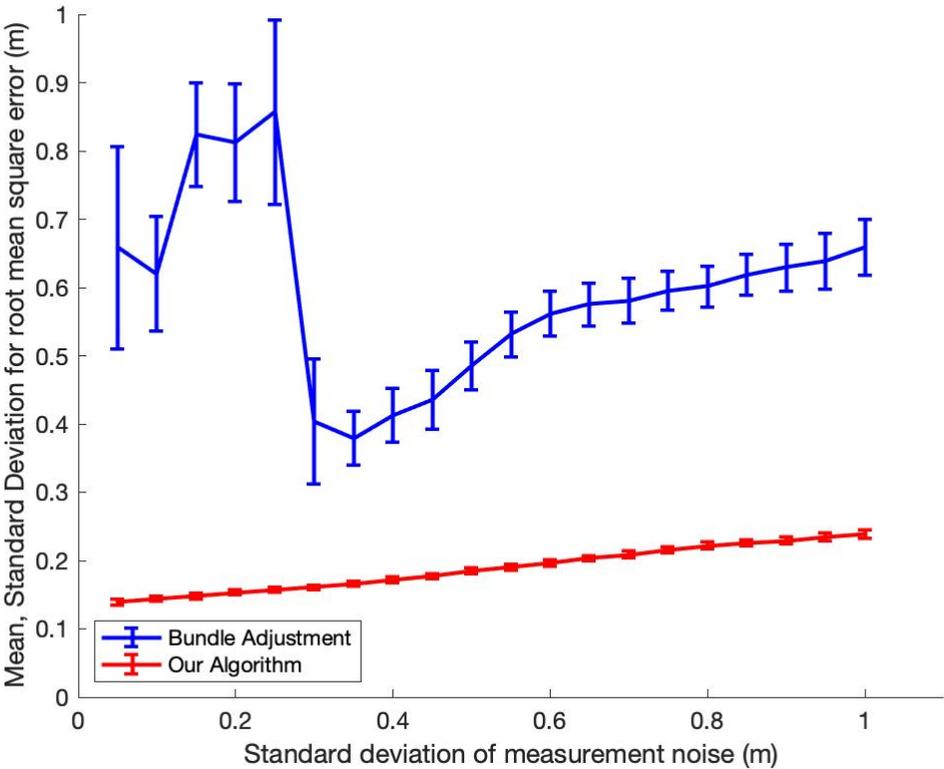


Figure 5.5: Mean and standard deviations of root-mean-square combined error of estimated vehicle and landmark positions vs. standard deviation of measurement noise.

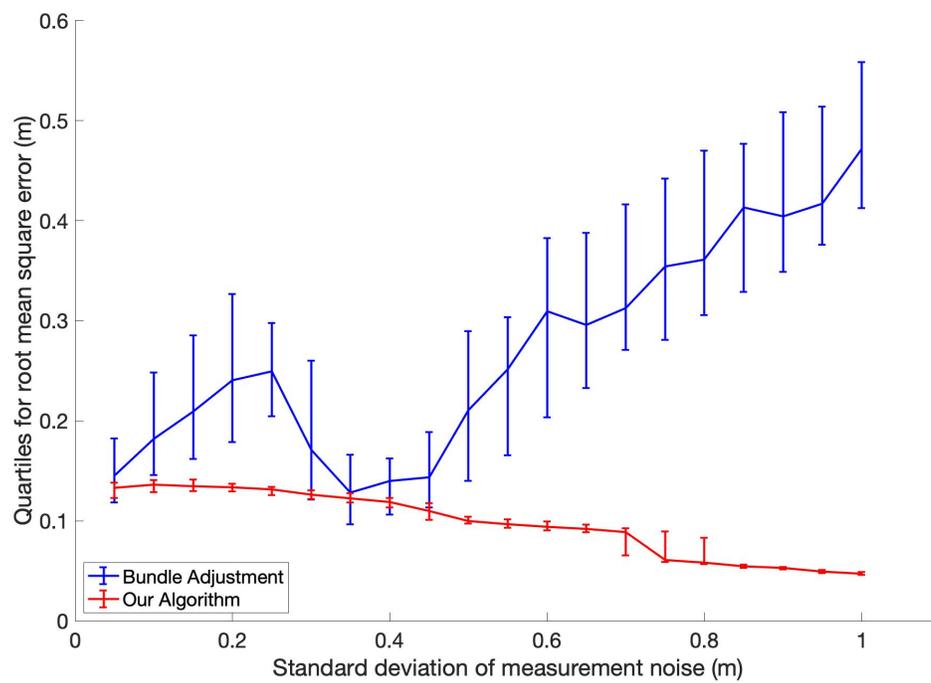


Figure 5.6: First and third quartiles of root-mean-square error of vehicle poses vs. standard deviation of measurement noise.

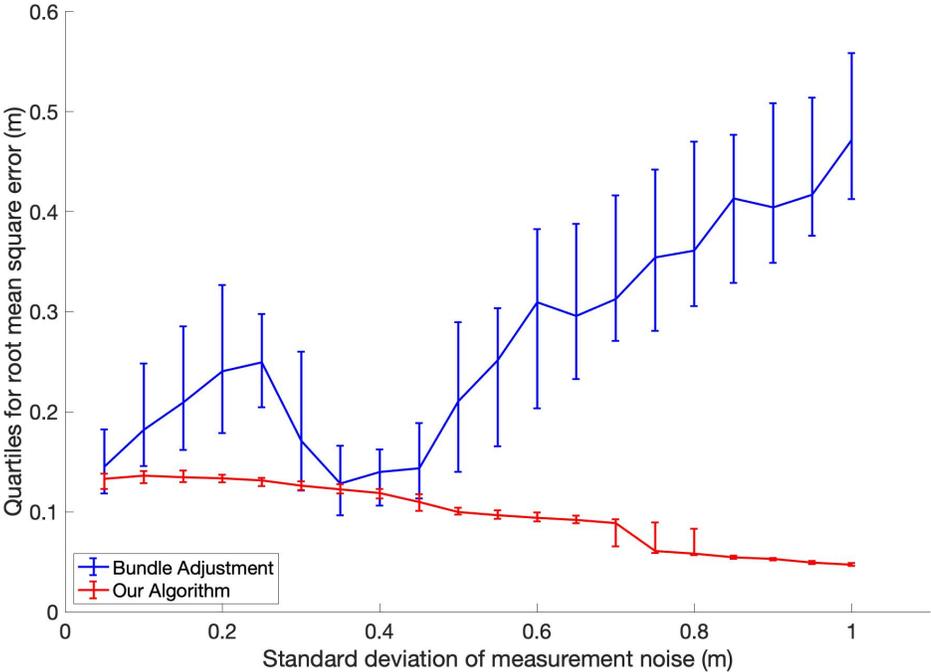


Figure 5.7: First and third quartiles of root-mean-square error of landmark positions vs. standard deviation of measurement noise.

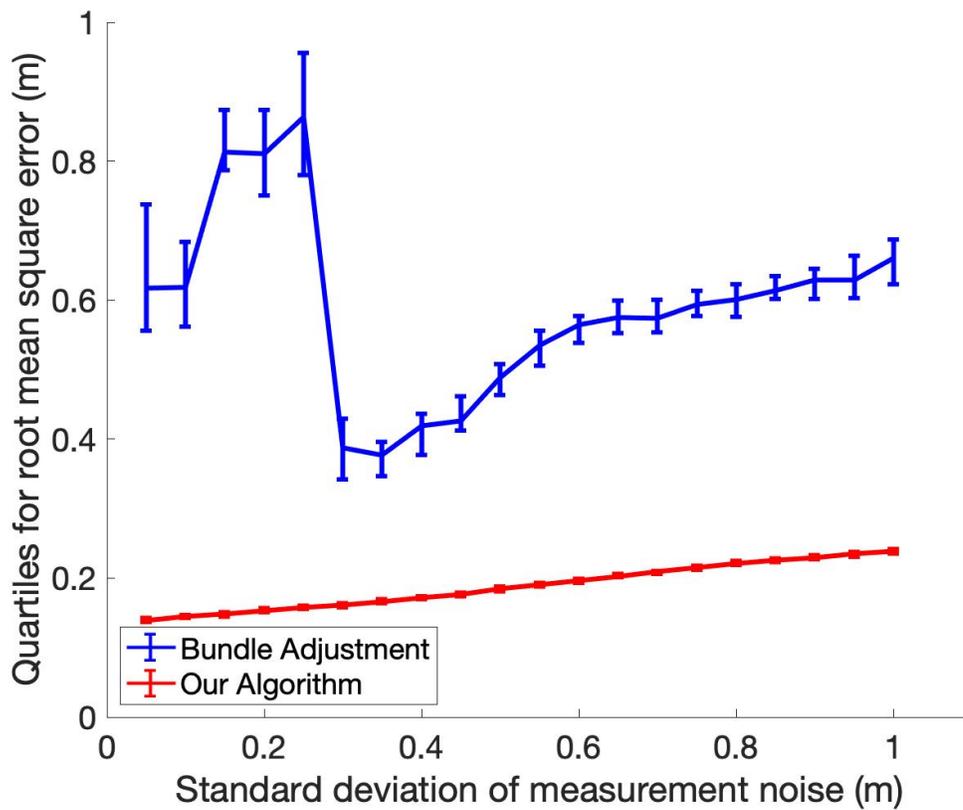


Figure 5.8: First and third quartiles of root-mean-square combined error of estimated vehicle and landmark positions vs. standard deviation of measurement noise.

Part III

Adaptive Tolling for Transportation Networks

Chapter 6

Adaptive Tolling for Arc-Based Traffic Assignment

In Chapters 6 and 7, we broaden our focus to encompass challenges in societal-scale navigation in modern transportation networks, in which self-interested commuters are aided by learning-enabled route recommendation algorithms. In particular, we study congestion management on transportation networks via tools from dynamic game theory and mechanism design. Reducing congestion on transportation networks is critical in urban environments, since the aggregate behavior of self-interested travelers often significantly increases commute time and pollution levels. In recent years, congestion pricing, i.e., tolling, has emerged as an increasingly popular tool for regulating traffic flows [73, 68]. The design of tolls that can effectually induce socially optimal traffic loads requires a realistic traffic assignment model (TAM) that captures commuters' routing preferences.

Many traffic assignment models (TAMs) in the literature employ a route-based modeling paradigm, which assumes that travelers make a single decision among routes that connect their origin and destination in a directed acyclic traffic network and do not deviate from their selection mid-route [112, 41, 92]. However, the exhaustive enumeration of routes is computationally infeasible for most real-world traffic networks, whose route counts can be exponential in the number of arcs they contain. Moreover, route-based models do not capture correlations between the total costs of routes that share arcs. To address these issues, my collaborators and I use an *arc-based TAM* [2, 3, 83, 84, 22, 58] to capture commuters' routing decisions. In this paradigm, commuters navigate through a traffic network by sequentially selecting among outgoing edges at each intermediate node. Toll design for arc-based TAMs is relatively under-studied, with the only exception of [58], in which the authors show that, similar to route based TAMs, marginal tolling also achieves social optimality in arc-based TAMs.

Fundamentally, tolls are designed and deployed with the aim of steering the equilibrium behavior of agents towards social optimality, by adding external incentives to their utility functions. A key, unrealistic assumption in this setting is that agents' behaviors are in equilibrium at all times. By contrast, real-world agents often iteratively update their strategies

based on repeated interactions, only eventually converging to an equilibrium outcome [47]. Although learning rules for route-based TAMs which provably converge to the equilibrium strategies do exist [72, 93], the development of analogous learning mechanisms for arc-based TAMs is relatively recent, e.g., in [22], which introduces a perturbed best response based dynamics. Thus, it is of interest to study tolling under the assumption that commuters’ strategies undergo such dynamic adaptation rules, instead of remaining fixed at equilibrium.

Prior work focused on empirical results also design tolls in dynamic environments by using reinforcement learning to iteratively update the toll on each arc. For instance, Chen et al. formulated the toll design problem as a Markov Decision Process (MDP) with high-dimensional state and action spaces, and apply a novel policy gradient algorithm to dynamically design tolls [15]. Mirzaei et al. used policy gradient methods to design incremental tolls on each link based on the difference between the observed and free-flow travel times [77]. Qiu et al. framed dynamic tolling as an instance of cooperative multi-agent reinforcement learning, and then applies graph convolutional networks to tractably solve the problem [89]. Likewise, Wang et al. use a cooperative actor-critic algorithm to tractably update a dynamic tolling scheme [109]. However, these methods operate on high-dimensional spaces, and are thus often computationally expensive. Moreover, they typically lack theoretical guarantees of convergence. Mirzaei et al. designed incremental tolls on each link, based on the difference between the observed and free-flow travel times, by using policy gradient methods [77]. Qiu et al. framed dynamic tolling as an instance of cooperative multi-agent reinforcement learning, and then applies graph convolutional networks to tractably solve the problem [89]. Likewise, Wang et al. deployed a cooperative actor-critic algorithm to update a dynamic tolling scheme [109]. However, these methods operate on high-dimensional spaces and are thus often computationally expensive, and usually do not readily admit theoretical guarantees of convergence.

In light of the strengths and shortcomings of the adaptive tolling mechanisms described above, this chapter presents an adaptive tolling scheme in the arc-based TAM detailed in [22]. We prove the existence of a toll which induces socially optimal congestion levels. Furthermore, we propose dynamics equations for adaptive tolling that drives the commuters’ routing preferences towards these socially optimal congestion levels. Our proof utilizes the theory of constant step-size two-timescale stochastic approximation [11], which allows us to decouple the toll and arc selection dynamics, and establish their convergence via two separate Lyapunov-based proofs. Although marginal tolling provably leads to socially efficient traffic allocation in a route-based TAM [92], to the best of our knowledge, our marginal tolling scheme is the first to induce socially optimal traffic flows in an arc-based setting. More details can be found in the following published papers: Chiu*, Maheshwari*, Su, and Sastry “Arc-based Traffic Assignment: Equilibrium Characterization and Learning” [22] (*Equal contribution) and Chiu, Maheshwari, Su, and Sastry “Dynamic Tolling in Arc-based Traffic Assignment Models” [20].

Remark 6.0.1. *The core methodology used in our work was first introduced in Maheshwari, Kulkarni, Wu, and Sastry, “Dynamic Tolling for Inducing Socially Optimal Traffic Loads”*

Table 6.1: Arc correspondences between the graphs in Figure 6.1: The original network (top left) and the CoDAG (top right).

Original	a_1^O	a_2^O	a_3^O	a_4^O	a_5^O	a_6^O	a_7^O	a_8^O	a_9^O
CoDAG	a_1^T	a_2^T	a_4^T	a_7^T	a_5^T	a_6^T	a_3^T	a_{11}^T	a_{12}^T
					a_9^T	a_8^T	a_{10}^T		

[72] to study adaptive tolling under the setting of parallel-link networks. Our method extends the scope of that work to transportation networks with bi-directional arcs, in the context of arc-based TAMs.

6.1 Preliminaries

Consider a traffic network described by a directed graph $G_O = (I_O, A_O)$, where I_O and A_O denote nodes and arcs, respectively. An example is shown in Figure 6.1 (top left); note that G_O can contain bidirectional arcs. Let the *origin nodes* and *destination nodes* be two disjoint subsets of I_O . To simplify our exposition, we assume that I_O contains only one origin $o \in I$ and one destination $d \in I$, although the results presented below straightforwardly extend to the multiple origin-destination-pair scenario. commuters navigate through the network, from origin o to destination d , by sequentially selecting arcs at every intermediate node. This process produces congestion on each arc, which in turn determines travel times. The *cost* on each arc is then obtained by summing the travel time and toll. Specifically, each arc $a \in A_O$ is associated with a toll $p_a \in \mathbb{R}^{|A_O|}$, and a positive, strictly increasing *latency function* $s_a : [0, \infty) \rightarrow [0, \infty)$, which gives travel time as a function of traffic flow. The cost on arc $a \in A_O$ is then given by:

$$c_a(w_a, p_a) = s_a(w_a) + p_a.$$

Finally, let the demand of (infinitesimal) commuters entering from origin node o be denoted by g_o .

Note that sequential arc selection on networks with bidirectional arcs can result in a cyclic route. For example, a traveler navigating the left traffic network in Figure 6.1 using sequential arc selection may cycle between nodes i_2^O and i_3^O . To resolve this issue, we consider arc selection on the *condensed DAG (CoDAG) representation* of the original network G_O , a directed acyclic graph (DAG) representation, as proposed in [22]. The Condensed DAG representation preserves all acyclic routes from origin o to destination d in G_O , but precludes cyclic routes by design. Details regarding the construction and properties of CoDAG representations are provided in [22], Section II.

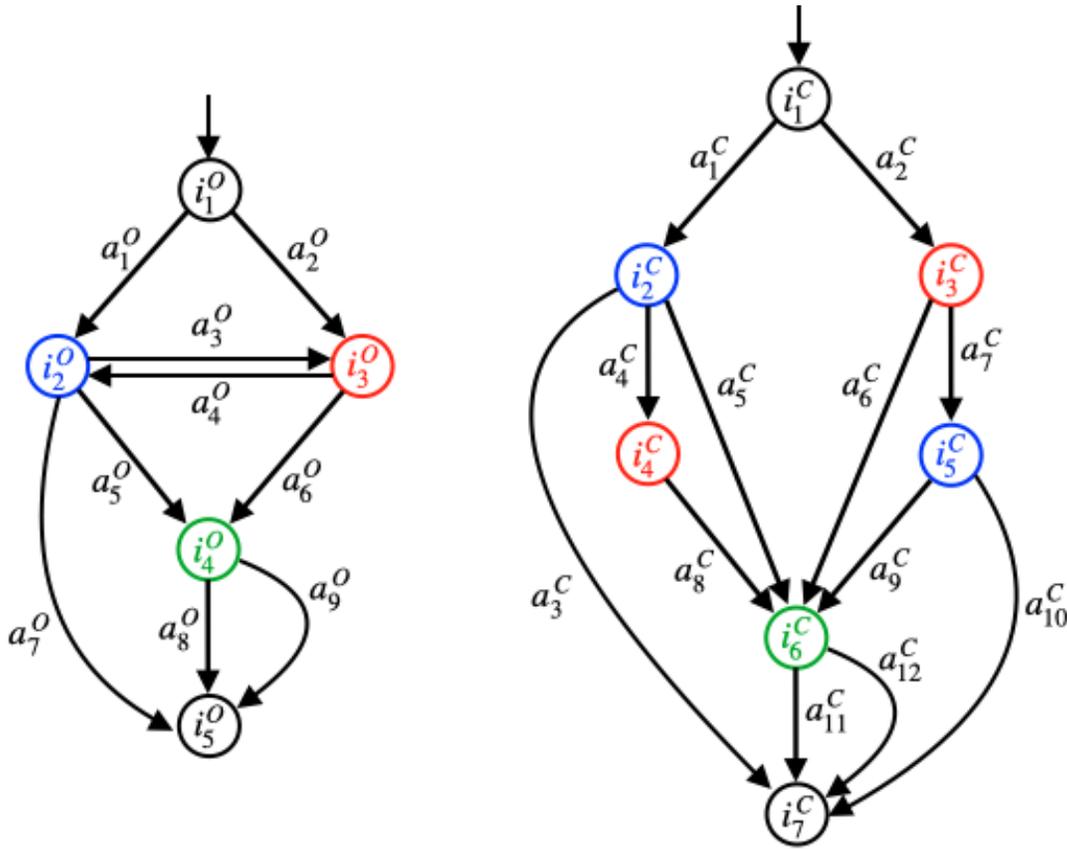


Figure 6.1: Example of a single-origin single-destination original network G_O (top left, with superscript O), and its corresponding condensed DAG, or CoDAG, representation G (top right, with superscript C). Arc correspondences between the two networks are given by Table 6.1, while node correspondences are indicated by color.

Preliminaries on DAG: Depth and Height

The exposition in subsequent sections of this paper requires the following definitions associated with the CoDAG representation. Let $G = (I, A)$ be a CoDAG representation of an original network G_O , where I and A are the nodes and arcs in G respectively. With a slight abuse of notation, note that the origin-destination pair of G is also (o, d) . Furthermore, let \mathbf{R} be the set of all acyclic routes in G which start at the origin node o and end at the destination node d .

Below, we will recursively define dynamical quantities, such as the time evolution of the traffic flows $w \in \mathbb{R}^{|A|}$ and the latency-to-go $z \in \mathbb{R}^{|A|}$, in a component-wise fashion, either from the origin of the Condensed DAG G towards the destination, or from the destination to the origin. To facilitate these recursive definitions, we require the following characterizations

regarding the depths and heights of arcs in a Condensed DAG G .

Definition 6.1.1 (Routes). A **route** r in a DAG G is an ordered subset of arcs A of the form $r = \{a_1, \dots, a_{|r|}\}$, with $i_{a_1} = o$, $j_{a_{|r|}} = d$, and $i_{a_k} = j_{a_{k-1}}$ for each $k \in \{2, \dots, |r|\}$. We denote by \mathbf{R} the set of all acyclic routes in G .

First, we define the concept of the depth of a directed acyclic graph (DAG), which will be crucial for the remaining exposition.

Definition 6.1.2 (Depth of a DAG). Given a DAG $G = (I, A)$ describing a single-origin single-destination traffic network, the depth of G , denoted $\ell(G)$, is defined by:

$$\ell(G) := \max_{a \in A} \ell_a.$$

In our method, we consider only acyclic routes in traffic networks with finitely many edges, so we have $\ell(G) < \infty$. Moreover, the case $\ell(G) = 1$ corresponds to a parallel link network, for which the results of the following proposition have already been analyzed in [72]. Therefore, we assume below that $\ell(G) \geq 2$.

Proposition 6.1.3. Given a Condensed DAG $G = (I, A)$ with the route set \mathbf{R} :

1. For any $a \in A$, we have $\ell_a = 1$ if and only if $i_a = o$. Similarly, if $\ell_a = \ell(G)$, then $j_a = d$.
2. For any fixed $r \in \mathbf{R}$, and any $a, a' \in r$ with $\ell_{a,r} < \ell_{a',r}$, we have $\ell_a < \ell_{a'}$ i.e., arcs along a route have strictly increasing depth from the origin to the destination.
3. Fix any $a \in A$, and any $r \in \mathbf{R}$ containing a such that $\ell_{a,r} = \ell_a$. Then, for any $a' \in \mathbf{R}$ preceding a in r , we have $\ell_{a',r} = \ell_{a'}$.
4. For each depth $k \in [\ell(G)] := \{1, \dots, \ell(G)\}$, there exists some $a \in A$ such that $\ell_a = k$.

Proof.

1. If $\ell_a \neq 1$, then $\ell_a \geq 2$, so there exists at least one route $r \in \mathbf{R}$ containing $a \in A$ such that $\ell_{a,r} \geq 2$. Thus, $i_a \neq o$ (otherwise the first $\ell_{a,r} - 1$ arcs of r would form a cycle). Conversely, if $i_a \neq o$, then no route $r \in \mathbf{R}$ contains $a \in A$ as its first arc, i.e., $\ell_{a,r} \geq 2$ for each $r \in \mathbf{R}$ containing a . Thus, $\ell_a = \max_{r \in \mathbf{R}: a \in r} \ell_{a,r} \geq 2$; in particular, $\ell_a \neq 1$. This establishes that $\ell_a = 1$ if and only if $i_a = o$.

Now, suppose by contradiction that there exists some $a \in A$ such that $\ell_a = \ell(G)$ but $j_a \neq d$. Fix any $r \in \mathbf{R}$ such that $a \in r$ and $\ell_{a,r} = \ell_a$. Then a cannot be at the end of \mathbf{R} , since by definition, routes must end at d . Let $a' \in r$ be the arc immediately after a in r . Then $\ell_{a'} \geq \ell_{a',r} = \ell_{a,r} + 1 = \ell(G) + 1$, a contradiction to the definition of $\ell(G)$.

2. Fix $r \in \mathbf{R}$, $a, a' \in r$ such that $\ell_{a,r} < \ell_{a',r}$. If $\ell_a = 1$, then $\ell_{a'} \geq \ell_{a',r} > \ell_{a,r} = 1 = \ell_a$, and we are done. Suppose $\ell_a \geq 2$. By definition of ℓ_a , there exists some route r_2 such that $\ell_{a,r_2} = \ell_a$. Construct a new route $r_3 \in \mathbf{R}$ by replacing the first $\ell_{a,r}$ arcs of r with the first ℓ_{a,r_2} arcs of r_2 . Then $\ell_{a'} \geq \ell_{a',r_3} = \ell_{a',r} - \ell_{a,r} + \ell_{a,r_2} > \ell_{a,r_2} = \ell_a$.
3. Fix any $a \in A$, and any $r \in \mathbf{R}$ containing a such that $\ell_{a,r} = \ell_a$. Suppose by contradiction that there exists some $a' \in \mathbf{R}$, preceding a in r , for which $\ell_{a'} \geq \ell_{a',r} + 1$. Then, by applying the second part of this lemma along the $(\ell_{a,r} - \ell_{a',r})$ arcs of \mathbf{R} from a' to a , we find that $\ell_a \geq \ell_{a'} + (\ell_{a,r} - \ell_{a',r}) \geq \ell_{a,r} + 1 = \ell_a + 1$, a contradiction.
4. Fix any arc $a \in A$ with $\ell_a = \ell(G)$. Then there exists some $r \in \mathbf{R}$ containing a such that $\ell_{a,r} = \ell_a = \ell(G)$. It follows from the third part of this proposition that, for each $k \in [\ell(G)]$, the k -th arc in \mathbf{R} is of depth k .

□

Next, we define the concept of height of a directed acyclic graph (DAG), which will be crucial for the remaining exposition.

Definition 6.1.4 (Height of a DAG). *Given a DAG $G = (I, A)$ describing a single-origin single-destination traffic network, the height of G , denoted $m(G)$, is defined by:*

$$m(G) := \max_{a \in A} m_a.$$

Since the traffic network under study is finite, and we consider only acyclic routes, we have $m(G) < \infty$. Moreover, the case $m(G) = 1$ corresponds to a parallel link network, for which the results of the following proposition have already been extensively analyzed in [72]. We will henceforth assume that $m(G) \geq 2$.

Proposition 6.1.5. *Given an Condensed DAG $G = (I, A)$ with the route set \mathbf{R} :*

1. *For any $a \in A$, we have $m_a = 1$ if and only if $j_a = d$. Similarly, if $m_a = m(G)$, then $i_a = o$.*
2. *For any fixed $r \in \mathbf{R}$, and any $a, a' \in r$ with $m_{a,r} < m_{a',r}$, we have $m_a < m_{a'}$ i.e., arcs along a route from the origin to the destination have strictly decreasing depth.*
3. *Fix any $a \in A$, and any $r \in \mathbf{R}$ containing a such that $m_{a,r} = m_a$. Then, for any $a' \in \mathbf{R}$ following a in r , we have $m_{a',r} = m_{a'}$.*
4. *For each height $k \in [m(G)] := \{1, \dots, m(G)\}$, there exists an arc $a \in A$ such that $m_a = k$.*

The proof of Proposition 6.1.5 parallels that of Proposition 6.1.3, and is omitted for brevity.

We define $[\cdot] : A \rightarrow A_O$ to be a map from each CoDAG arc $a \in A$ to the corresponding arc in the original graph, $[a] \in A_O$ (as shown in Table 6.1). For each arc $a \in A$, let i_a and j_a denote the start and terminal nodes, and for each node $i \in I$, let $A_i^-, A_i^+ \subset A$ denote the set of incoming and outgoing arcs.

Cost Model

Below, we assume that every traveler has access to G_O , and to the same CoDAG representation $G = (I, A)$ of G_O ; in particular, G is used to perform sequential arc selection to generate acyclic routes. The commuters' aggregative arc selections generate network congestion. Specifically, for each $a \in A$, let the *flow* or *congestion level* on arc a be denoted by w_a , and let the total flow on the corresponding arc in the original network be denoted, with a slight abuse of notation, by $w_{[a]} := \sum_{a' \in [a]} w_{a'}^1$. commuters perceive the cost on each arc $a \in A$ as:

$$\tilde{c}_{[a]}(w_{[a]}, p_{[a]}) := c_{[a]}(w_{[a]}, p_{[a]}) + \nu_a = s_{[a]}(w_{[a]}) + p_{[a]} + \nu_a,$$

where ν_a is a zero-mean random variable. At each non-destination node $i \in I \setminus \{d\}$, commuters select among outgoing nodes $a \in A_i^+$ by comparing their perceived cost-to-go $\tilde{z}_a : \mathbb{R}^{|A|} \times \mathbb{R}^{|A_O|} \rightarrow \mathbb{R}$, given recursively by:

$$\begin{aligned} \tilde{z}_a(w, p) &:= \tilde{s}_{[a]}(w_{[a]}) + p_{[a]} + \min_{a' \in A_{j_a}^+} \tilde{z}_{a'}(w, p), & j_a \neq d, \\ \tilde{z}_a(w, p) &:= \tilde{s}_{[a]}(w_{[a]}) + p_{[a]}, & j_a = d. \end{aligned} \quad (6.1)$$

Consequently, the fraction of commuters who arrives at $i \in I \setminus \{d\}$ and choose arc $a \in A_i^+$ is given by:

$$P_{ij_a} := \mathbb{P}(\tilde{z}_a \leq \tilde{z}_{a'}, \forall a' \in A_i^+). \quad (6.2)$$

An explicit formula for the probabilities $\{P_{ij_a} : a \in A_i^+\}$, in terms of the statistics of \tilde{z}_a , is provided by the discrete-choice theory [8]. In particular, define $z_a(w) := \mathbb{E}[\tilde{z}_a(w)]$ and $\epsilon_a := \tilde{z}_a(w) - z_a(w)$, and define the latency-to-go at each node by:

$$\varphi_i(\{z_{a'}(w, p) : a' \in A_i^+\}) = \mathbb{E} \left[\min_{a' \in A_i^+} \tilde{z}_{a'}(w, p) \right]. \quad (6.3)$$

Then, from discrete-choice theory [8]:

$$P_{ij_a} = \frac{\partial \varphi_i}{\partial z_a}(z), \quad i \in I \setminus \{d\}, a \in A_i^+, \quad (6.4)$$

¹Unlike existing TAMs, in our model, the latency of arcs in G can be coupled, since multiple copies of the same arc in G_O may exist in G .

where, with a slight abuse of notation, we write $\varphi_i(z)$ for $\varphi_i(\{z_{a'} : a' \in A_i^+\})$.

To obtain a closed-form expression of φ , we employ the *logit Markovian model* [2, 3], under which the noise terms ϵ_a are described by the Gumbel distribution with scale parameter β . As a result, the expected minimum cost-to-go $z_a : \mathbb{R}^{|A|} \times \mathbb{R}^{|A^o|} \rightarrow \mathbb{R}$, associated with traveling on each arc $a \in A$, assumes the following form:

$$z_a(w, p) = s_{[a]} \left(\sum_{\bar{a} \in [a]} w_{\bar{a}} \right) + p_{[a]} - \frac{1}{\beta} \ln \left(\sum_{a' \in A_{j_a}^+} e^{-\beta z_{a'}(w, p)} \right). \quad (6.5)$$

Note that (6.5) is well-posed, as z_a can be recursively computed from the destination back to the origin ([22], Section III).

CoDAG Equilibrium

Here, we define the *condensed DAG (CoDAG) equilibrium* (Definition 6.1.6), based on the CoDAG representation of the original traffic network. Specifically, we show that the CoDAG equilibrium exists, is unique, and solves a strictly convex optimization problem (Theorem 6.1.7).

Definition 6.1.6 (Condensed DAG Equilibrium). Fix a toll vector $p \in \mathbb{R}^{|A^o|}$, and fix $\beta > 0$. We call an arc-flow vector $\bar{w}^\beta(p) \in \mathbb{R}^{|A|}$ a Condensed DAG (CoDAG) equilibrium at p if, for each $i \in I \setminus \{d\}$, $a \in A_i^+$:

$$\bar{w}_a^\beta(p) = \left(g_i + \sum_{a' \in A_i^+} \bar{w}_{a'}^\beta(p) \right) \frac{\exp(-\beta z_a(\bar{w}^\beta(p), p))}{\sum_{a' \in A_{i_a}^+} \exp(-\beta z_{a'}(\bar{w}^\beta(p), p))}, \quad (6.6)$$

where $g_i = g_0 \cdot \mathbf{1}(i = o)$, and $w \in \mathcal{W}$, where:

$$\mathcal{W} := \left\{ w \in \mathbb{R}^{|A|} : \sum_{a \in A_i^+} w_a = \sum_{a \in A_i^-} w_a, \forall i \neq o, d, \right. \\ \left. \sum_{a \in A^+} w_a = g_o, w_a \geq 0, \forall a \in A \right\} \quad (6.7)$$

characterizes the conservation of flow in the CoDAG G . Note that \mathcal{W} is convex and compact.

At a CoDAG equilibrium $\bar{w}^\beta(p)$, the fraction of commuters at any intermediate node $i \in I \setminus \{d\}$ who selects an arc $a \in A_i^+$ is given by $\bar{\xi}_a^\beta(p)$, as defined below:

$$\bar{\xi}_a^\beta(p) := \frac{\bar{w}_a^\beta(p)}{\sum_{a' \in A_i^+} \bar{w}_{a'}^\beta(p)}.$$

The CoDAG equilibrium bears some resemblance to the Markovian Traffic Equilibrium (MTE) introduced in Baillon and Cominetti [3]. However, the CoDAG formulation by design precludes the possibility of assigning cyclic routes, and is capable of capturing couplings between arcs in the CoDAG G that correspond to the same arc in the original network G_O (see [22], Remark 6).

Below, we show that, given any CoDAG representation G of an original network G_O and any fixed toll vector $p \in \mathbb{R}^{|A_O|}$, the CoDAG equilibrium exists and is unique. Specifically, the CoDAG equilibrium is the unique minimizer of a strictly convex optimization problem over a compact set. This characterization provides powerful insight into the mathematical properties of the CoDAG equilibrium flow, and its dependence on the toll vector. These properties will be used in our work to establish the existence of an optimal toll (Theorem 6.2.1) and the convergence of our discrete-time toll dynamics to the optimal toll (Theorem 6.3.1).

For each $[a] \in A_O$, define $F : \mathcal{W} \times \mathbb{R}^{|A_O|} \rightarrow \mathbb{R}$ by:

$$F(w, p) = \sum_{[a] \in A_O} \int_0^{w_{[a]}} [s_{[a]}(u) + p_{[a]}] du \quad (6.8)$$

$$+ \frac{1}{\beta} \sum_{i \neq d} \left[\sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right].$$

Theorem 6.1.7. *For each fixed toll vector $p \in \mathbb{R}^{|A_O|}$, the corresponding CoDAG equilibrium $\bar{w}^\beta(p) \in \mathcal{W}$ exists, is unique, and is the unique minimizer of $F(\cdot, p)$ over \mathcal{W} .*

Proof. (Proof Sketch) The proof parallels that of [22], Theorem 1 and Lemma 1. For details, please see [22], Section III and Appendix B. \square

To prove Theorem 6.1.7, we first show that for each fixed toll vector $p \in \mathbb{R}^{|A_O|}$, the map $F(\cdot, p)$ is strictly convex over \mathcal{W} (Lemma 6.1.8). Therefore, F has a unique minimizer in \mathcal{W} . It then suffices to show that the CoDAG equilibrium definition (Definition 6.1.6) matches the Karush-Kuhn-Tucker (KKT) conditions for the optimization problem (6.8).

Lemma 6.1.8. *For any fixed toll vector $p \in \mathbb{R}^{|A_O|}$, the map $F(\cdot, p) : \mathcal{W} \rightarrow \mathbb{R}$ is strictly convex.*

Proof. (Proof Sketch) For convenience, we define $f_{[a]} : \mathcal{W} \rightarrow \mathbb{R}$, $\chi_i : \mathbb{R}^{|A_i^+|} \rightarrow \mathbb{R}$, $F : \mathcal{W} \rightarrow \mathbb{R}$ for each $[a] \in A_O$, $i \in I \setminus \{d\}$ by:

$$f_{[a]}(w) := \int_0^{w_{[a]}} s_{[a]}(u) du, \quad \forall [a] \in A_O,$$

$$\chi_i(w_{A_i^+}) := \sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right), \quad \forall i \neq I \setminus \{d\},$$

where $w_{A_i^+} \in \mathbb{R}^{|A_i^+|}$ denotes the components of w corresponding to arcs in A_i^+ . Then:

$$F(w) = \sum_{[a] \in A_0} f_{[a]}(w) + \frac{1}{\beta} \sum_{i \in I \setminus \{d\}} \chi_i^\beta(w).$$

Also, for convenience, define:

$$\mathcal{W}_s := \left\{ w \in \mathbb{R}^{|A|} : \sum_{a \in A_i^+} w_a = \sum_{a \in A_i^-} w_a, \forall i \neq o, d, \sum_{a \in A_o^+} w_a = 0 \right\}. \quad (6.9)$$

Essentially, \mathcal{W}_s is the tangent space of the linear manifold with boundary \mathcal{W} . We can rewrite (6.9) as:

$$\mathcal{W}_s = \{e_{A_i^-} - e_{A_i^+} : i \neq o, d\}^\perp \cap \{e_{A_o^+}\}^\perp.$$

We can now establish the strict convexity of F .

We first establish the convexity of F . It suffices to show that $f_{[a]}$ and χ_i are convex for each $[a] \in A_0$, $i \in I \setminus \{d\}$. Note that each $f_{[a]}$ is convex since it is the composition of a convex function ($g(w) = \sum_{a \in A_0} \int_0^{w_a} s_a(u) du$) with a linear function ($w_{[a]} := \sum_{a' \in [a]} w_{a'}$). We show below that χ_i is convex, for each $i \in I \setminus \{d\}$.

Fix $i \in I \setminus \{d\}$. For any $a, a' \in A_i^+$ and each $w \in \mathcal{W}$:

$$\frac{\partial^2 \chi_i}{\partial w_a \partial w_{a'}}(w) = \frac{1}{w_a} \mathbf{1}\{a' = a\} - \frac{1}{\sum_{\bar{a} \in A_i^+} w_{\bar{a}}}.$$

Thus, for any $y \in \mathbb{R}^{|A_i^+|}$:

$$\begin{aligned} y^\top \nabla_w^2 \chi_i(w) y &= \sum_{a, a' \in A_i^+} y_a y_{a'} \frac{\partial^2 \chi_i}{\partial w_a \partial w_{a'}}(w) \\ &= \sum_{a \in A_i^+} \frac{y_a^2}{w_a} - \frac{1}{\sum_{\bar{a} \in A_i^+} w_{\bar{a}}} \cdot \sum_{a, a' \in A_i^+} y_a y_{a'} \\ &= \frac{1}{\sum_{\bar{a} \in A_i^+} w_{\bar{a}}} \left(\sum_{\bar{a} \in A_i^+} w_{\bar{a}} \cdot \sum_{a \in A_i^+} \frac{y_a^2}{w_a} - \left(\sum_{a' \in A_i^+} y_{a'} \right)^2 \right) \\ &= \frac{1}{\sum_{\bar{a} \in A_i^+} w_{\bar{a}}} \left(\sum_{\bar{a} \in A_i^+} (\sqrt{w_{\bar{a}}})^2 \cdot \sum_{a \in A_i^+} \left(\frac{y_a}{\sqrt{w_a}} \right)^2 - \left(\sum_{a' \in A_i^+} \sqrt{w_{a'}} \cdot \frac{y_{a'}}{\sqrt{w_{a'}}} \right)^2 \right) \\ &\geq 0, \end{aligned} \quad (6.10)$$

where the final inequality follows from the Cauchy-Schwarz inequality. Cauchy-Schwarz also implies that equality holds in (6.10) if and only if the vectors $(\sqrt{w_{\bar{a}}})_{\bar{a} \in A_i^+} \in \mathbb{R}^{|A_i^+|}$ and

$(y_a/\sqrt{w_a})_{a \in A_i^+} \in \mathbb{R}^{|A_i^+|}$ are parallel, i.e., if $(y_a)_{a \in A_i^+}$ and $(w_a)_{a \in A_i^+}$ are scalar multiples of each other. This shows that χ_i is convex, and $\dim(N(\nabla_w^2 \chi_i)) = 1$.

Second, suppose by contradiction that F is not strictly convex on \mathcal{W} . Then there exists some $\bar{w} \in \mathcal{W}$, $z \in \mathcal{W}_s \setminus \{0\}$ such that:

$$z^\top \nabla_w^2 F(\bar{w}) z = 0.$$

Since $\nabla_w^2 F(\bar{w})$ is symmetric positive semidefinite, this is equivalent to stating that z is in $N(\nabla_w^2 F(\bar{w}))$, the null space of $\nabla_w^2 F(\bar{w})$. Let A_z denote the set of arc indices for which z has a nonzero component, i.e.:

$$A_z := \{a' \in A : z_{a'} \neq 0\}.$$

Since z is not the zero vector, A_z is non-empty. Since there are a discrete and finite number of levels of G , there exists some $a \in A_z$ such that $\ell_a \leq \ell_{a'}$ for all $a' \in A_z$, i.e., $\ell_a = \min\{\ell_{a'} : a' \in A_z\}$. Without loss of generality, we consider the case $z_a > 0$ (if not, then replace z with $-z$, which would also be a nonzero vector in $N(\nabla_w^2 F(\bar{w}))$). We claim that $w_a \neq 0$, and that for all $a' \in A_i^+$:

$$z_{a'} = z_a \cdot \frac{w_{a'}}{w_a} \geq 0.$$

To see this, note that otherwise, the vectors $(z_a)_{a \in A_i^+} \in \mathbb{R}^{|A_i^+|}$ and $(w_a)_{a \in A_i^+}$ are not parallel, and so equality cannot be obtained in (6.10), i.e.,:

$$z^\top \nabla_w^2 \chi_i(\bar{w}) z > 0,$$

where, with a slight abuse of notation, we have defined $\chi_i(w) = \chi_i(A_i^+)$. As a result:

$$z^\top \nabla_w^2 F(\bar{w}) z = \sum_{[a] \in A} z^\top \nabla_w^2 f_{[a]}(\bar{w}) z + \frac{1}{\beta} \sum_{i' \neq d} z^\top \nabla_w^2 \chi_{i'}(\bar{w}) z \geq \frac{1}{\beta} z^\top \nabla_w^2 \chi_i(\bar{w}) z > 0,$$

a contradiction. Thus, $z_a > 0$, and $z_{a'} \geq 0$ for each $a' \in A_i^+$, so:

$$\sum_{a' \in A_i^+} z_{a'} > 0.$$

If $\ell_a = 1$, i.e., $i_a = o$, we arrive at a contradiction, since the fact that $z \in \mathcal{W}_s$ implies $\sum_{a' \in A_i^+} z_{a'} = 0$. If $\ell_a > 1$, we also arrive at a contradiction, since the fact that $z \in \mathcal{W}_s$ implies:

$$\sum_{\hat{a} \in A_{i_a}^-} z_{\hat{a}} = \sum_{a' \in A_i^+} z_{a'} > 0,$$

so there exists at least one $\hat{a} \in A_{i_a}^-$ with $z_{\hat{a}} > 0$. Then, by definition of $a \in A$, we have $\ell_a \leq \ell_{\hat{a}}$; this contradicts Proposition 6.1.3, Part 2, which implies that since $\hat{a} \in A_{i_a}^-$, there exists at least one arc containing \hat{a} immediately before $a \in A$, and thus $\ell_{\hat{a}} \leq \ell_a - 1$. These contradictions complete the proof of the strict convexity of F on \mathcal{W} . \square

Social Optimality

We now describe the socially optimal flow which would minimize the total latency over the entire transportation network. More specifically, we define below the notion of *perturbed social optimality* considered in our work.

Definition 6.1.9 (Perturbed Socially Optimal Flow). *We define a perturbed socially optimal flow with regularization parameter $\beta > 0$ to be a minimizer of the following convex optimization problem:*

$$\min_{w \in \mathcal{W}} \sum_{[a] \in A_O} w_{[a]} \cdot s_{[a]}(w_{[a]}) + \frac{1}{\beta} \sum_{i \neq d} \left[\sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right],$$

with \mathcal{W} given by (6.7), and $w_{[a]} := \sum_{a' \in [a]} w_{a'}$, as defined above.

In words, perturbed social optimality is characterized as the total latency experienced by commuters on each arc of the CoDAG G , augmented by an entropy term with regularization parameter β which captures stochasticity in the commuters' arc selections.

6.2 Optimal Toll: Existence and Uniqueness

Below, we characterize the *optimal toll* $\bar{p} \in \mathbb{R}^{|A_O|}$ for which the corresponding CoDAG equilibrium $\bar{w}^\beta(\bar{p})$ is perturbed socially optimal (see Definition 6.1.9). Throughout the rest of the paper, we call \bar{p} the *optimal toll*.

Theorem 6.2.1. *There exists a unique toll vector $\bar{p} \in \mathbb{R}^{|A_O|}$ that satisfies the following fixed-point equation:*

$$\bar{p}_{[a]} = \bar{w}_{[a]}^\beta(\bar{p}) \cdot \frac{ds_{[a]}}{dw} \bar{w}_{[a]}^\beta(\bar{p}), \quad \forall a \in A. \quad (6.11)$$

Moreover, $\bar{w}^\beta(\bar{p})$, the CoDAG equilibrium flow distribution corresponding to \bar{p} , is the perturbed socially optimal flow with regularization β .

To prove Theorem 6.2.1, we first show that $\bar{w}^\beta(p)$ is continuous and monotonic in the toll p (Lemmas 6.2.2 and 6.2.3). Then, we use these properties to establish the existence and uniqueness of a toll vector $\bar{p} \in \mathbb{R}^{|A_O|}$ satisfying the fixed-point equation (6.11) (Lemma 6.2.4). Finally, we prove that the CoDAG equilibrium flow allocation $\bar{w}^\beta(\bar{p})$ corresponding to \bar{p} is perturbed socially optimal (Lemma 6.2.5).

Below, we begin by establishing that the CoDAG equilibrium $\bar{w}^\beta(p)$ is a continuously differentiable and monotonic function of the toll $p \in \mathbb{R}^{|A_O|}$.

Lemma 6.2.2. *$\bar{w}^\beta(p)$ is continuously differentiable in p .*

Proof. Define $F : \mathcal{W} \times \mathbb{R}^{|A_O|} \rightarrow \mathbb{R}$ by:

$$\begin{aligned} F(w, p) := & \sum_{[a] \in A_O} \int_0^{w_{[a]}} [s_{[a]}(z) + p_{[a]}] dz \\ & + \frac{1}{\beta} \sum_{i \neq d} \left[\sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right]. \end{aligned} \quad (6.12)$$

The theory of constrained optimization implies that, for each p , the unique minimizer of $F(\cdot, p) : \mathcal{W} \rightarrow \mathbb{R}$ is completely characterized via a set of equality constraints, which we describe below. First, recall that since \mathcal{W} is a subset of an affine subspace of $\mathbb{R}^{|A|}$ characterized by $|I \setminus \{d\}|$ equality constraints, there exist $M \in \mathbb{R}^{|A| \times |I \setminus \{d\}|}$, of full column rank, and $b \in \mathbb{R}^{|I \setminus \{d\}|}$ such that:

$$\mathcal{W} = \{w \in \mathbb{R}^{|A|} : M^\top w + b = 0, w_a \geq 0, \forall a \in A\}.$$

Moreover, by using QR decomposition, we can assume that the columns of M are orthonormal. Next, let $B \in \mathbb{R}^{|A| \times (|A| - |I \setminus \{d\}|)}$ be given such that the columns of B have unit norm, are pair-wise orthogonal, and are each orthogonal to the subspace of $\mathbb{R}^{|A|}$ spanned by the columns of M , i.e., B^\top maps each vector in $\mathbb{R}^{|A|}$ to the coefficients of its projection onto the linear subspace orthogonal to \mathcal{W} , with respect to an ordered, orthonormal basis of that subspace. Then the theory of constrained optimization, and the strict convexity of $F(\cdot, p)$, imply that $\bar{w}^\beta(p)$, the unique minimizer of $F(\cdot, p)$, is completely characterized by the equations:

$$\begin{aligned} M^\top w + b &= 0, \\ B^\top \nabla_w F(w, p) &= 0. \end{aligned}$$

To this end, define $J : \mathbb{R}^{|A_O|} \times \mathbb{R}^{|A|} \rightarrow \mathbb{R}^{|A|}$ by:

$$J(w, p) := \begin{bmatrix} M^\top w + b \\ B^\top \nabla_w F(w, p) \end{bmatrix}.$$

Note that J is continuously differentiable almost everywhere, with:

$$\frac{\partial J}{\partial w}(w, p) = \begin{bmatrix} M^\top \\ B^\top \nabla_w^2 F(w, p) \end{bmatrix} \in \mathbb{R}^{|A| \times |A|}.$$

Suppose by contradiction that $\frac{\partial J}{\partial w}(w, p) \in \mathbb{R}^{|A| \times |A|}$ is singular at some (w, p) . Then $\frac{\partial J}{\partial w}(w, p)^\top \in \mathbb{R}^{|A| \times |A|}$ lacks full column rank, i.e.:

$$\dim(R(M) + R(\nabla_w^2 F(w, p)B)) = \text{rank}([M \quad \nabla_w^2 F(w, p)B]) \leq |A| - 1.$$

By the Boolean formula for sums of vector spaces:

$$\dim(R(M) \cap R(\nabla_w^2 F(w, p)B))$$

$$\begin{aligned}
 &= \dim(R(M)) + \dim(R(\nabla_w^2 F(w, p)B)) - \dim(R(M) + R(\nabla_w^2 F(w, p)B)) \\
 &= \dim(R(M)) + \dim(R(B)) - \dim(R(M) + R(\nabla_w^2 F(w, p)B)) \\
 &\geq |A| - (|A| - 1) \\
 &= 1.
 \end{aligned}$$

Thus, there exists some nonzero vector $v \in R(M) \cap R(\nabla_w^2 F(w, p)B)$. Since $v \in R(M)$, and the columns of B are orthogonal to $R(M)$, we have $B^\top v = 0$. Meanwhile, since $v \in R(\nabla_w^2 F(w, p)B)$, there exists some nonzero $w \in \mathbb{R}^{|A|-d}$ such that $u = \nabla_w^2 F(w, p)Bw$. Thus, we have:

$$0 = B^\top u = B^\top \nabla_w^2 F(w, p)Bw,$$

a contradiction, since the fact that B^\top has full row rank and $\nabla_w^2 F(w, p)$ is symmetric positive definite implies that $B^\top \nabla_w^2 F(w, p)B$ is symmetric positive definite, and $u \neq 0$ by construction. This establishes that $\frac{\partial J}{\partial w}(w, p) \in \mathbb{R}^{|A| \times |A|}$ is non-singular at each $(w, p) \in \mathbb{R}^{|A_O|} \times \mathbb{R}^{|A|}$. The existence and continuity of $\frac{d\bar{w}^\beta}{dp}(p)$ at each $p \in \mathbb{R}^{|A_O|}$ now follows from the Implicit Function Theorem. \square

Lemma 6.2.3. *For any $p, p' \in \mathbb{R}^{|A_O|}$:*

$$\sum_{a \in A} \left(\bar{w}_a^\beta(p') - \bar{w}_a^\beta(p) \right) (p'_{[a]} - p_{[a]}) \leq 0.$$

Proof. In this subsection, we show that for any $p, p' \in \mathbb{R}^{|A_O|}$:

$$\sum_{a \in A} \left(\bar{w}_a^\beta(p') - \bar{w}_a^\beta(p) \right) (p'_{[a]} - p_{[a]}) \leq 0.$$

By Theorem 6.1.7, $\bar{w}^\beta(p)$ is the unique minimizer, in \mathcal{W} , of the following strictly convex function of w :

$$\sum_{[a] \in A_O} \int_0^{w_{[a]}} [s_{[a]}(z) + p_{[a]}] dz + \frac{1}{\beta} \sum_{i \neq d} \left[\sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right].$$

Applying first-order conditions for optimality in constrained convex optimization, we obtain that, for each $w^1 \in \mathcal{W}$:

$$\sum_{a \in A} \left[s_{[a]}(\bar{w}_{[a]}^\beta(p)) + p_{[a]} + \frac{1}{\beta} \ln \left(\frac{\bar{w}_a^\beta(p)}{\sum_{a' \in A_i^+} \bar{w}_{a'}^\beta(p)} \right) \right] \cdot (w_a^1 - \bar{w}_a^\beta(p)) \geq 0.$$

Similarly, for $\bar{w}_{[a]}^\beta(p')$, we obtain that for each $w^2 \in \mathcal{W}$:

$$\sum_{a \in A} \left[s_{[a]}(\bar{w}_{[a]}^\beta(p')) + p'_{[a]} + \frac{1}{\beta} \ln \left(\frac{\bar{w}_a^\beta(p')}{\sum_{a' \in A_i^+} \bar{w}_{a'}^\beta(p')} \right) \right] \cdot (w_a^2 - \bar{w}_a^\beta(p')) \geq 0.$$

Taking $w^1 := \bar{w}_a^\beta(p')$, $w^2 := \bar{w}_a^\beta(p)$, and adding the above two inequalities, we have:

$$0 \leq \sum_{a \in A} (\bar{w}_a^\beta(p') - \bar{w}_a^\beta(p)) \cdot \left[s_{[a]}(\bar{w}_{[a]}^\beta(p)) - s_{[a]}(\bar{w}_{[a]}^\beta(p')) + p_{[a]} - p'_{[a]} + \frac{1}{\beta} \ln \left(\frac{\bar{w}_a^\beta(p)}{\sum_{a' \in A_{i_a}^+} \bar{w}_{a'}^\beta(p)} \right) - \frac{1}{\beta} \ln \left(\frac{\bar{w}_a^\beta(p')}{\sum_{a' \in A_{i_a}^+} \bar{w}_{a'}^\beta(p')} \right) \right].$$

Since the maps $w_a \mapsto s_{[a]}(w_{[a]})$ and $w_a \mapsto \ln(w_a / \sum_{a' \in A_{i_a}^+} w_{a'})$ are non-decreasing, by rearranging terms, we obtain:

$$\sum_{a \in A} (\bar{w}_a^\beta(p') - \bar{w}_a^\beta(p)) (p'_{[a]} - p_{[a]}) \leq 0,$$

as desired. Additionally, it also holds that

$$\sum_{[a] \in A_O} (\bar{w}_{[a]}^\beta(p') - \bar{w}_{[a]}^\beta(p)) (p_{[a]} - p'_{[a]}) \leq 0.$$

□

We then use the above lemmas to prove that the fixed-point equation (6.11) yields a unique solution.

Lemma 6.2.4. *There exists a unique $\bar{p} \in \mathbb{R}^{|A_O|}$, called the **optimal toll** in subsequent discussions, which satisfies (6.11):*

$$\bar{p}_{[a]} = \bar{w}_{[a]}^\beta(\bar{p}) \cdot \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(\bar{p})), \quad \forall [a] \in A_O.$$

Proof. Below, we show that there exists a unique $\bar{p} \in \mathbb{R}^{|A_O|}$ satisfying (6.11):

$$\bar{p}_{[a]} = \bar{w}_{[a]}^\beta(\bar{p}) \cdot \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(\bar{p})), \quad \forall [a] \in A_O.$$

Define $\psi : \mathbb{R}^{|A_O|} \rightarrow \mathbb{R}$ as:

$$\psi_{[a]}(p) := w_{[a]}(p) \cdot \frac{ds_{[a]}}{dw}(w_{[a]}(p)), \quad \forall [a] \in A_O.$$

Since $w_{[a]}(\cdot)$ is continuous (Lemma 6.2.2), and $s_{[a]}$ is continuously differentiable, the map ψ is continuous. Define the set:

$$K := \left\{ y \in \mathbb{R}^{|A_O|} : y \succeq 0, \|y\|_1 \leq |A_O| g_o \max_{[a] \in A_O} \frac{ds_{[a]}}{dw}(g_o) \right\}.$$

Observe that K is a compact and convex subset of $\mathbb{R}^{|A_O|}$, and ψ maps K to K , since for any $p \in K$, we have $\psi_a(p) \geq 0$ for each $a \in A$, and:

$$\begin{aligned} \|\psi(p)\|_1 &= \sum_{a \in A_O} \psi_a(p) \\ &= \sum_{a \in A_O} \bar{w}_{[a]}(p) \cdot \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}(p)) \\ &\leq \max_{a \in A_O} \frac{ds_{[a]}}{dw}(g_o) \cdot \sum_{a \in A_O} \bar{w}_{[a]}(p) \\ &\leq |A_O| g_o \cdot \max_{a \in A_O} \frac{ds_{[a]}}{dw}(g_o). \end{aligned}$$

Thus, by the Brouwer's fixed point theorem, there exists a fixed point $\bar{p} \in K \subset \mathbb{R}^{|A_O|}$ of ψ , i.e., there exists $\bar{p} \in \mathbb{R}^{|A_O|}$ satisfying (6.11), i.e.,:

$$\bar{p}_{[a]} = \bar{w}_{[a]}^\beta(\bar{p}) \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(\bar{p})), \quad \forall [a] \in A_O.$$

Next, we show that \bar{p} is unique up to Markovian Traffic Equilibrium on the original traffic network, i.e., any $p' \in \mathbb{R}^{|A_O|}$ satisfies (6.11) if and only if $\bar{w}_{[a]}^\beta(p') = \bar{w}_{[a]}^\beta(\bar{p})$ for each $a \in A$. To show this, suppose by contradiction that there exists some $p' \in \mathbb{R}^{|A_O|}$ satisfying (6.11), such that $\bar{w}_{[a]}^\beta(p') \neq \bar{w}_{[a]}^\beta(\bar{p})$ for some $[a] \in A_O$. Then:

$$\begin{aligned} &\bar{p}_{[a]} - p'_{[a]} \\ &= \bar{w}_{[a]}^\beta(\bar{p}) \cdot \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(\bar{p})) - \bar{w}_{[a]}^\beta(p') \cdot \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(p')) \\ &= \left[\bar{w}_{[a]}^\beta(\bar{p}) - \bar{w}_{[a]}^\beta(p') \right] \cdot \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(\bar{p})) + \bar{w}_{[a]}^\beta(p') \cdot \left[\frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(\bar{p})) - \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(p')) \right]. \end{aligned}$$

Rearranging terms, and invoking the strict convexity and increasing nature of each $s_{[a]}$, and the fact that $\bar{w}_{[a]}^\beta(\bar{p}) \neq \bar{w}_{[a]}^\beta(p')$ for some $[a] \in A_O$, we obtain:

$$\begin{aligned} &\sum_{a \in A} \left[\bar{w}_a^\beta(\bar{p}) - \bar{w}_a^\beta(p') \right] (\bar{p}_{[a]} - p'_{[a]}) \\ &= \sum_{[a] \in A_O} \left[\bar{w}_{[a]}^\beta(\bar{p}) - \bar{w}_{[a]}^\beta(p') \right] (\bar{p}_{[a]} - p'_{[a]}) \\ &= \sum_{[a] \in A_O} \left[\bar{w}_{[a]}^\beta(\bar{p}) - \bar{w}_{[a]}^\beta(p') \right]^2 \cdot \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(\bar{p})) \\ &\quad + \sum_{[a] \in A_O} \bar{w}_{[a]}^\beta(p') \left[\bar{w}_{[a]}^\beta(\bar{p}) - \bar{w}_{[a]}^\beta(p') \right]^2 \cdot \left[\frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(\bar{p})) - \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(p')) \right] \end{aligned}$$

> 0 ,

which contradicts Theorem 6.1.7 .

The above arguments establish that if $p' \in \mathbb{R}^{|A_O|}$ satisfies (6.11), then $\bar{w}_{[a]}^\beta(p') = \bar{w}_{[a]}^\beta(\bar{p})$ for each $[a] \in A_O$. Through (6.11), we then have, for each $[a] \in A_O$:

$$\bar{p}_{[a]} = \bar{w}_{[a]}^\beta(\bar{p}) \cdot \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(\bar{p})) = \bar{w}_{[a]}^\beta(p') \cdot \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(p')) = p'_{[a]},$$

so $p' = \bar{p}$. This concludes the proof. \square

Finally, we prove that the CoDAG equilibrium flow corresponding to $\bar{p} \in \mathbb{R}^{|A_O|}$ is perturbed socially optimal.

Lemma 6.2.5. $\bar{w}^\beta(\bar{p})$ is perturbed socially optimal.

Proof. Below, we show that $\bar{w}^\beta(\bar{p})$ is perturbed socially optimal. Let $w^* \in \mathbb{R}^{|A|}$ denote the perturbed socially optimal load. Recall that, by Theorem 6.1.7 and the definition of the perturbed socially optimal load:

$$\begin{aligned} \bar{w}^\beta(\bar{p}) &= \arg \min_{w \in \mathcal{W}} \left\{ \sum_{[a] \in A_O} \int_0^{w_{[a]}} [s_{[a]}(z) + p_{[a]}] dz \right. \\ &\quad \left. + \frac{1}{\beta} \sum_{i \neq d} \left[\sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right] \right\}, \\ w^* &= \arg \min_{w \in \mathcal{W}} \left\{ \sum_{[a] \in A_O} w_{[a]} \ln w_{[a]} \right. \\ &\quad \left. + \frac{1}{\beta} \sum_{i \neq d} \left[\sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right] \right\}. \end{aligned}$$

The proof follows by verifying that the variational inequalities corresponding to the above two optimization problems are the same. These two variational inequalities in question are respectively given by:

$$\begin{aligned} \sum_{[a] \in A_O} \left[s_{[a]}(\bar{w}_{[a]}^\beta(\bar{p})) + \bar{w}_{[a]}^\beta(\bar{p}) \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(\bar{p})) \right. \\ \left. + \frac{1}{\beta} \ln \left(\frac{\bar{w}_{[a]}^\beta(\bar{p})}{\sum_{a' \in A_{i_a}^+} \bar{w}_{[a']}^\beta(\bar{p})} \right) \right] (w_a - \bar{w}_{[a]}^\beta(\bar{p})) > 0, \\ \forall w \in \mathcal{W}, w \neq \bar{w}_{[a]}^\beta(\bar{p}), \end{aligned}$$

$$\sum_{[a] \in A_o} \left[s_{[a]}(w_{[a]}^*) + w_{[a]}^* \frac{ds_{[a]}}{dw}(w_{[a]}^*) + \frac{1}{\beta} \ln \left(\frac{\bar{w}_{[a]}^*}{\sum_{a' \in A_{i_a}^+} w_{[a']}^*} \right) \right] (w_a - w_a^*) > 0,$$

$$\forall w \in \mathcal{W}, w \neq w^*,$$

and are thus, indeed, identical. This confirms that $\bar{w}^\beta(\bar{p}) = w^*$, and concludes the proof. \square

Together, Lemmas 6.2.2, 6.2.3, 6.2.4, and 6.2.5 prove Theorem 6.2.1.

6.3 Dynamics and Convergence

Discrete-time Dynamics

Below, we introduce discrete-time stochastic dynamics to describe flow and toll evolution on the traffic network G . Formally, g_o non-atomic units of traveler enter G at the origin node o at each time step $n \geq 0$. At each non-destination node $i \in I \setminus \{d\}$, a $\xi_a[n]$ fraction of commuters chooses an outgoing arc $a \in A_i^+$. We shall refer to $\xi_a[n]$ as the *aggregate arc selection probability*. Consequently, the flow induced on any arc $a \in A$ satisfies:

$$W_a[n] = \left(g_{i_a} + \sum_{a' \in A_{i_a}^+} W_{a'}[n] \right) \cdot \xi_a[n]. \quad (6.13)$$

At the end of each time step n , commuters arrive at the destination node d and observe a noisy estimate of the cost-to-go values and tolls on all arcs in the network (including arcs not traversed during that time step). Let $K_i > 0$ be node-dependent constants, to be specified shortly, and let $\{\eta_i[n+1] \in \mathbb{R} : i \in I, n \geq 0\}$ be independent bounded random variables² in $[\underline{\mu}, \bar{\mu}]$, with $0 < \underline{\mu} < \mu < \bar{\mu} < 1/\max\{K_i : i \in I \setminus \{d\}\}$ and $\mathbb{E}[\eta_{i_a}[n+1]] = \mu$ at each node $i \in I$ and discrete time index $n \geq 0$. Based on the observed latencies, at time $n+1$, for each non-destination node $i \in I \setminus \{d\}$, a $\eta_i[n+1] \cdot K_i$ fraction of commuters at node $i \in I$ switches to the outgoing arc that minimizes the observed cost-to-go. Meanwhile, $1 - \eta_i[n+1] \cdot K_i$ fraction of commuters chooses the same arc they selected at time step n . We assume that $\{\eta_i[n+1] \in \mathbb{R} : i \in I, n \geq 0\}$ are independent bounded random variables³ in $[\underline{\mu}, \bar{\mu}]$, with $0 < \underline{\mu} < \mu < \bar{\mu} < 1$ and $\mathbb{E}[\eta_{i_a}[n+1]] = \mu$ for each node $i \in I$ and discrete time index $n \geq 0$. Thus, the arc selection probabilities evolve according to the following *perturbed best-response*

²The random variables $\{\eta_a[n] : a \in A, n \geq 0\}$ are assumed to be independent of travelers' perception uncertainties.

³The random variables $\{\eta_a[n] : a \in A, n \geq 0\}$ are assumed to be independent of commuters' perception uncertainties.

dynamics:

$$\xi_a[n+1] = \xi_a[n] + \eta_{i_a}[n+1] \cdot K_{i_a} \cdot \left(-\xi_a[n] + \frac{\exp(-\beta[z_a(W[n], P[n])])}{\sum_{a' \in A_{i_a}^+} \exp(-\beta[z_{a'}(W[n], P[n])])} \right). \quad (6.14)$$

We assume that $\xi_a[0] > 0$ for each $a \in A$, i.e., each arc has some strictly positive initial traffic flow. This captures the stochasticity in commuters' perception of network congestion, which causes each arc to be assigned a nonzero probability of being selected.

At each time step $n+1 \geq 0$, the tolls $P_{[a]}[n] \in \mathbb{R}^{|A_O|}$ on each arc $[a] \in A_O$ are updated by interpolating between the tolls implemented at time step n , and the marginal latency of that arc given the flow at time step n . That is:

$$P_{[a]}[n+1] = P_{[a]}[n] + \gamma \left(-P_{[a]}[n] + W_{[a]}[n] \cdot \frac{ds_{[a]}}{dw}(W_{[a]}[n]) \right), \quad (6.15)$$

with $\gamma \in (0, 1)^4$, where with a slight abuse of notation, we denote $W_{[a]} := \sum_{a' \in [a]} W_{a'}$. Note that the update (6.15) is distributed, i.e., for each arc in the original network, the updated toll depends only on the flow of that arc, and not on the flow of any other arc. Moreover, we assume that $\gamma \ll \mu$, i.e., the toll updates (6.15) occur at a slower timescale compared to the arc selection probability updates (6.14).

Convergence Results

In this subsection, we show that the arc selection probability and toll updates (6.14)-(6.15) converge in the neighborhood of the socially optimal flow $\bar{w}^\beta(\bar{p})$ and the corresponding toll \bar{p} respectively.

Theorem 6.3.1. *The joint evolution of arc selection probability and toll updates (6.14)-(6.15) satisfies*

$$\limsup_{n \rightarrow \infty} \mathbb{E} [\|\xi[n] - \bar{\xi}^\beta(\bar{p})\|_2^2 + \|P[n] - \bar{p}\|_2^2] = O\left(\mu + \frac{\gamma}{\mu}\right).$$

Consequently, for each $\delta > 0$:

$$\limsup_{n \rightarrow \infty} \mathbb{P}(\|\xi[n] - \bar{\xi}^\beta(\bar{p})\|_2^2 + \|P[n] - \bar{p}\|_2^2 \geq \delta) = O\left(\frac{\mu}{\delta} + \frac{\gamma}{\delta\mu}\right).$$

⁴Our result also holds if γ is a random variable with bounded support.

To prove Theorem 6.3.1, we employ the theory of two-timescale stochastic approximation [10]. Consequently, the asymptotic behavior of (6.14)-(6.15) can be studied by studying the convergence properties of the corresponding continuous-time dynamical system. Since the tolls are updated at a slower rate compared to the traffic flows ($\gamma \ll \mu$), we consider the evolution of continuous-time flows $w(t)$ under a fixed toll $p \in \mathbb{R}^{|A_O|}$, and continuous-time tolls $p(t)$ with flow converged at the corresponding CoDAG equilibrium $\bar{w}^\beta(p(t))$ at each time. Specifically, for any fixed toll $p \in \mathbb{R}^{|A_O|}$, on each arc $a \in A$, the arc selection probabilities evolve as follows:

$$w_a(t) = \xi_a(t) \cdot \left(g_{i_a} + \sum_{a' \in A_{i_a}^-} w_{a'}(t) \right), \quad (6.16)$$

$$\dot{\xi}_a(t) = K_{i_a} \left(-\xi_a(t) + \frac{\exp(-\beta \cdot z_a(w(t), p))}{\sum_{a' \in A_{i_a}^+} \exp(-\beta \cdot z_{a'}(w(t), p))} \right). \quad (6.17)$$

Meanwhile, on each arc $[a] \in A_O$ on the original network, we consider the following continuous-time toll dynamics:

$$\dot{p}_{[a]}(t) = -p_{[a]}(t) + \bar{w}_{[a]}^\beta(p(t)) \cdot \theta_{[a],1}. \quad (6.18)$$

We prove that, for each fixed toll $p \in \mathbb{R}^{|A_O|}$, the corresponding continuous-time ξ -dynamics (6.17) globally asymptotically converges to the corresponding CoDAG equilibrium $\bar{w}^\beta(p) \in \mathbb{R}^{|A|}$. Moreover, the continuous-time toll dynamics (6.18) globally converges to the optimal toll $\bar{p} \in \mathbb{R}^{|A_O|}$.

Lemma 6.3.2. *Suppose $w(0) \in \mathcal{W}$, i.e., the initial flow satisfies flow continuity. For each fixed toll vector $p \in \mathbb{R}^{|A_O|}$, the continuous-time flow dynamics induced by the arc-selection dynamics (6.17) globally asymptotically converges to the corresponding CoDAG equilibrium $\bar{w}^\beta(p)$.*

Proof. We recursively write the continuous-time evolution of the arc flows $w(\cdot)$ as follows, from (6.17) and (6.16). At any $w \in \mathcal{W}$, for each $a \notin A_o^+$:

$$\begin{aligned} & \dot{w}_a(t) \\ &= \dot{\xi}_a(t) \cdot \left(g_{i_a} + \sum_{\hat{a} \in A_{i_a}^-} w_{\hat{a}}(t) \right) + \xi_a(t) \cdot \sum_{a' \in A_{i_a}^-} \dot{w}_{a'}(t) \end{aligned} \quad (6.19)$$

$$\begin{aligned} &= K_{i_a} \left(-\xi_a(t) + \frac{\exp(-\beta z_a(w(t), p))}{\sum_{a' \in A_{i_a}^+} \exp(-\beta z_{a'}(w(t), p))} \right) \cdot \left(g_{i_a} + \sum_{\hat{a} \in A_{i_a}^-} w_{\hat{a}}(t) \right) \\ & \quad + \xi_a(t) \cdot \sum_{a' \in A_{i_a}^-} \dot{w}_{a'}(t) \end{aligned} \quad (6.20)$$

$$\begin{aligned}
 &= -K_{i_a} w_a(t) + K_{i_a} \cdot \left(g_{i_a} + \sum_{\hat{a} \in A_{i_a}^-} w_{\hat{a}}(t) \right) \cdot \frac{\exp(-\beta z_a(w(t), p))}{\sum_{a' \in A_{i_a}^+} \exp(-\beta z_{a'}(w(t), p))} \\
 &\quad + \frac{w_a(t)}{\sum_{a' \in A_{i_a}^+} w_{a'}(t)} \cdot \sum_{\hat{a} \in A_{i_a}^-} \dot{w}_{\hat{a}}(t) \\
 &= -K_{i_a} \left(1 - \frac{\sum_{a' \in A_{i_a}^-} \dot{w}_{a'}}{K_{i_a} \cdot \sum_{\hat{a} \in A_{i_a}^+} w_{\hat{a}}} \right) w_a \\
 &\quad + K_{i_a} \cdot \left(g_{i_a} + \sum_{\hat{a} \in A_{i_a}^-} w_{\hat{a}}(t) \right) \cdot \frac{\exp(-\beta z_a(w(t), p))}{\sum_{a' \in A_{i_a}^+} \exp(-\beta z_{a'}(w(t), p))},
 \end{aligned} \tag{6.21}$$

for each $a \in A$. More formally, we define each component $h : \mathcal{W} \rightarrow \mathbb{R}^{|A|}$ recursively as follows. First, for each $a \in A_o^+$, we set:

$$h_a(w, p) := K_o \left(-w_a + g_o \cdot \frac{\exp(-\beta z_a(w, p))}{\sum_{a' \in A_o^+} \exp(-\beta z_{a'}(w, p))} \right).$$

Suppose now that, for some arc $a \in A$, the component $h_a : \mathcal{W} \rightarrow \mathbb{R}$ of h has been defined for each $\hat{a} \in A_{i_a}^-$. Then, we set:

$$h_a(w, p) := -K_{i_a} \left(1 - \frac{\sum_{a' \in A_{i_a}^-} h_{a'}(w, p)}{K_{i_a} \cdot \sum_{\hat{a} \in A_{i_a}^+} w_{\hat{a}}} \right) w_a + K_{i_a} \cdot \sum_{a' \in A_{i_a}^-} w_{a'} \cdot \frac{\exp(-\beta z_a(w, p))}{\sum_{a' \in A_o^+} \exp(-\beta z_{a'}(w, p))}.$$

By iterating through the above definition forward through the Condensed DAG G from origin to destination (in other words, along nodes of increasing depth), we can completely specify each h_a in a well-posed manner (For a more rigorous characterization of this iterative procedure, see Appendix A, Proposition 1). We then define the w -dynamics corresponding to the ξ -dynamics (6.17) by:

$$\dot{w} = h(w, p). \tag{6.22}$$

Now, recall the objective $F : \mathcal{W} \times \mathbb{R}^{|A_o|} \rightarrow \mathbb{R}$ of the optimization problem that characterizes \bar{w}^β , first stated in Theorem 6.1.7 as Equation (6.8), reproduced below:

$$\begin{aligned}
 F(w, p) &:= \sum_{[a] \in A_o} \int_0^{w_{[a]}} [s_{[a]}(z) + p_{[a]}] dz \\
 &\quad + \frac{1}{\beta} \sum_{i \neq d} \left[\sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right].
 \end{aligned}$$

Roughly speaking, our main approach is to show that F is a Lyapunov equation for the best-response dynamics in (6.22). Specifically, let \mathcal{W}_s denote the tangent space to \mathcal{W} , and let $\Pi_{\mathcal{W}_s}$ denote the orthogonal projection onto \mathcal{W}_s . Under the continuous-time flow dynamics (6.17) and (6.16):

$$\begin{aligned} & \frac{d}{dt}F(w(t), p) \\ &= \dot{w}(t)^\top \nabla_w F(w(t), p) \end{aligned} \quad (6.23)$$

$$= \dot{w}(t)^\top \Pi_{\mathcal{W}_s} \nabla_w F(w(t), p) \quad (6.24)$$

$$\begin{aligned} &= \dot{w}(t)^\top \Pi_{\mathcal{W}_s} (\nabla_w f(w(t), p) + \nabla \chi^\beta(w(t))) \\ &= \dot{w}(t)^\top \Pi_{\mathcal{W}_s} \left((s_{[a]}(w_{[a]}(t)) + p_{[a]})_{a \in A} + \nabla \chi^\beta(w(t)) \right) \end{aligned} \quad (6.25)$$

$$\begin{aligned} &= \dot{w}(t)^\top \Pi_{\mathcal{W}_s} \left[-\nabla \chi^\beta \left(\left(\left(g_{i_a} + \sum_{a' \in A_{i_a}^-} w_{a'}(t) \right) \cdot \frac{\exp(-\beta z_a(w(t), p))}{\sum_{\bar{a} \in A_i^+} \exp(-\beta z_{\bar{a}}(w(t), p))} \right)_{a \in A} \right) \right. \\ & \quad \left. + \nabla \chi^\beta(w(t)) \right] \end{aligned} \quad (6.26)$$

$$\begin{aligned} &= \dot{w}(t)^\top \Pi_{\mathcal{W}_s} \left[-\nabla \chi^\beta \left(\left(\left(g_{i_a} + \sum_{a' \in A_{i_a}^-} w_{a'}(t) \right) \cdot \frac{\exp(-\beta z_a(w(t), p))}{\sum_{\bar{a} \in A_i^+} \exp(-\beta z_{\bar{a}}(w(t), p))} \right)_{a \in A} \right) \right. \\ & \quad \left. + \nabla \chi^\beta \left(\left(\left(1 - \frac{\sum_{a' \in A_{i_a}^-} h_{a'}(w(t), p)}{K_{i_a} \cdot \sum_{\hat{a} \in A_{i_a}^+} w_{\hat{a}}(t)} \right) w_a(t) \right)_{a \in A} \right) \right] \end{aligned} \quad (6.27)$$

$$\begin{aligned} &= \left[\left(-K_{i_a} \cdot \left(1 - \frac{\sum_{a' \in A_{i_a}^-} h_{a'}(w(t), p)}{K_{i_a} \cdot \sum_{\hat{a} \in A_{i_a}^+} w_{\hat{a}}(t)} \right) w_a(t) \right. \right. \\ & \quad \left. \left. + K_{i_a} \cdot \left(g_{i_a} + \sum_{a' \in A_{i_a}^-} w_{a'}(t) \right) \cdot \frac{\exp(-\beta z_a(w(t), p))}{\sum_{\bar{a} \in A_i^+} \exp(-\beta z_{\bar{a}}(w(t), p))} \right)_{a \in A} \right]^\top \\ & \quad \left[-\nabla \chi^\beta \left(\left(\left(g_{i_a} + \sum_{a' \in A_{i_a}^-} w_{a'}(t) \right) \cdot \frac{\exp(-\beta z_a(w(t), p))}{\sum_{\bar{a} \in A_i^+} \exp(-\beta z_{\bar{a}}(w(t), p))} \right)_{a \in A} \right) \right. \\ & \quad \left. + \nabla \chi^\beta \left(\left(\left(1 - \frac{\sum_{a' \in A_{i_a}^-} h_{a'}(w(t), p)}{K_{i_a} \cdot \sum_{\hat{a} \in A_{i_a}^+} w_{\hat{a}}(t)} \right) w_a(t) \right)_{a \in A} \right) \right] \end{aligned}$$

< 0 .

We explain the equalities (6.23) = (6.24), (6.25) = (6.26) and (6.26) = (6.27) below. After (6.26), the rest of the proof follows by substituting in the definition of \dot{w} , recalling from “(6.23) = (6.24)” that $\dot{w} \in \mathcal{W}_s$, and invoking the strict convexity of χ^β .

Verifying (6.23) = (6.24) From the equations leading up to (6.21), we have, for each $w \in \mathcal{W}$:

$$\begin{aligned}
 \dot{w}_a(t) &= h_a(w(t), p) \\
 &= -K_{i_a} \left(1 - \frac{\sum_{a' \in A_{i_a}^-} \dot{w}_{a'}}{K_{i_a} \cdot \sum_{\hat{a} \in A_{i_a}^+} w_{\hat{a}}} \right) w_a \\
 &\quad + K_{i_a} \cdot \left(g_{i_a} + \sum_{a' \in A_{i_a}^-} w_{a'}(t) \right) \cdot \frac{\exp(-\beta z_a(w(t), p))}{\sum_{a' \in A_{i_a}^+} \exp(-\beta z_{a'}(w(t), p))} \\
 &= -K_{i_a} \cdot w_a(t) + K_{i_a} \cdot \left(g_{i_a} + \sum_{a' \in A_{i_a}^-} w_{a'}(t) \right) \cdot \frac{\exp(-\beta z_a(w(t), p))}{\sum_{a' \in A_{i_a}^+} \exp(-\beta z_{a'}(w(t), p))} \\
 &\quad + \frac{w_a}{\sum_{\hat{a} \in A_{i_a}^+} w_{\hat{a}}} \cdot \sum_{a' \in A_{i_a}^-} \dot{w}_{a'}(t) \\
 &= -K_{i_a} \cdot w_a(t) + K_{i_a} \cdot \left(g_{i_a} + \sum_{a' \in A_{i_a}^-} w_{a'}(t) \right) \cdot \frac{\exp(-\beta z_a(w(t), p))}{\sum_{a' \in A_{i_a}^+} \exp(-\beta z_{a'}(w(t), p))} \\
 &\quad + \xi_a(t) \cdot \sum_{a' \in A_{i_a}^-} \dot{w}_{a'}(t).
 \end{aligned}$$

Fix any node $i \in I$ in the Condensed DAG, and consider the sum of the above equation over the arc subset A_i^+ :

$$\sum_{a' \in A_i^+} \dot{w}_{a'}(t) = - \sum_{a' \in A_i^+} w_{a'}(t) + \left(g_i + \sum_{\hat{a} \in A_{i_a}^-} w_{\hat{a}}(t) \right) \cdot 1 + 1 \cdot \sum_{\hat{a} \in A_{i_a}^-} \dot{w}_{\hat{a}}(t).$$

Rearranging terms, we obtain:

$$\frac{d}{dt} \left(\sum_{a' \in A_i^+} w_{a'} - \sum_{\hat{a} \in A_{i_a}^-} w_{\hat{a}} - g_i \right) = - \left(\sum_{a' \in A_i^+} w_{a'} - \sum_{\hat{a} \in A_{i_a}^-} w_{\hat{a}} - g_i \right).$$

Since $w(0) \in \mathcal{W}$ by assumption, we have the initial condition $(\sum_{a' \in A_i^+} w_{a'} - \sum_{\hat{a} \in A_{i_a}^-} w_{\hat{a}} - g_i)(0) = 0$ for the above linear time-invariant differential equation. We thus conclude that, for each $t \geq 0$:

$$\sum_{a' \in A_i^+} w_{a'}(t) - \sum_{\hat{a} \in A_{i_a}^-} w_{\hat{a}}(t) - g_i = 0.$$

Since this holds for any arbitrary node $i \in I$, we have $w(t) \in \mathcal{W}$ for all $t \geq 0$.

Verifying (6.25) = (6.26) We will show that:

$$\begin{aligned} \Pi_{\mathcal{W}_s} \left[\left(s_{[a]}(w_{[a]}(t)) \right)_{a \in A} + \nabla \chi^\beta \left(\left(\left(g_{i_a} + \sum_{a' \in A_{i_a}^-} w_{a'}(t) \right) \right. \right. \right. \\ \left. \left. \left. \cdot \frac{\exp(-\beta z_a(w(t), p))}{K_{i_a} \cdot \sum_{\bar{a} \in A_i^+} \exp(-\beta z_{\bar{a}}(w(t), p))} \right) \right)_{a \in A} \right] = 0, \end{aligned} \quad (6.28)$$

which would a fortiori establish the desired claim (6.25) = (6.26). To do so, first note that, for each $i \neq d$, $a \in A_i^+$:

$$\frac{\partial \chi^\beta}{\partial w_a}(w) = \frac{1}{\beta} \cdot \left[\ln w_a + 1 - \ln \left(\sum_{a \in A_i^+} w_a \right) - 1 \right] = \frac{1}{\beta} \ln \left(\frac{w_a}{\sum_{a \in A_i^+} w_a} \right). \quad (6.29)$$

Thus, we have:

$$\begin{aligned} & \frac{\partial \chi^\beta}{\partial w_a} \left(\left(\left(g_{i_a} + \sum_{a' \in A_{i_a}^-} w_{a'} \right) \cdot \frac{\exp(-\beta z_a(w, p))}{\sum_{\bar{a} \in A_i^+} \exp(-\beta z_{\bar{a}}(w, p))} \right)_{a \in A} \right) \\ &= \frac{1}{\beta} \ln \left(\frac{\exp(-\beta z_a(w, p))}{\sum_{\bar{a} \in A_i^+} \exp(-\beta z_{\bar{a}}(w, p))} \right) \\ &= -z_a(w, p) - \frac{1}{\beta} \ln \left(\sum_{\bar{a} \in A_i^+} \exp(-\beta z_{\bar{a}}(w, p)) \right) \\ &= -z_a(w, p) + \varphi_{i_a}(w, p). \end{aligned}$$

Concatenating these partial derivatives to form the gradient, we can now verify (6.28) by observing that:

$$\begin{aligned} & \Pi_{\mathcal{W}_s} \left[\left(s_{[a]}(w_{[a]}) + p_{[a]} \right)_{a \in A} \right. \\ & \quad \left. + \nabla \chi^\beta \left(\left(\left(g_{i_a} + \sum_{a' \in A_{i_a}^-} w_{a'} \right) \cdot \frac{\exp(-\beta z_{\hat{a}}(w, p))}{\sum_{\bar{a} \in A_i^+} \exp(-\beta z_{\bar{a}}(w, p))} \right)_{\hat{a} \in A} \right)_{a \in A} \right] \\ &= \Pi_{\mathcal{W}_s} \left(s_{[a]}(w_{[a]}) + p_{[a]} - z_a(w, p) + \varphi_{i_a}(w, p) \right)_{a \in A} \\ &= \Pi_{\mathcal{W}_s} \left(\varphi_{i_a}(w, p) - \varphi_{j_a}(w, p) \right)_{a \in A} \\ &= \Pi_{\mathcal{W}_s} \left[\sum_{a \in A} \varphi_{i_a}(w, p) e_a - \sum_{a \in A} \varphi_{j_a}(w, p) e_a \right] \end{aligned}$$

$$\begin{aligned}
 &= \Pi_{\mathcal{W}_s} \left[- \sum_{\hat{a} \in A_d^-} \varphi_{j_{\hat{a}}}(w, p) e_{\hat{a}} + \sum_{a' \in A_o^+} \varphi_{i_{a'}}(w, p) e_{a'} \right. \\
 &\quad \left. + \sum_{i \neq \{o, d\}} \left(\sum_{a' \in A_i^+} \varphi_i(w, p) e_{a'} - \sum_{\hat{a} \in A_i^-} \varphi_i(w, p) e_{\hat{a}} \right) \right] \\
 &= \Pi_{\mathcal{W}_s} \left[0 + \varphi_o(w, p) e_{A_o^+} + \sum_{i \neq \{o, d\}} \varphi_i(w, p) (e_{A_i^-} - e_{A_i^+}) \right] \\
 &= 0,
 \end{aligned}$$

where the last equality follows by definition of $\Pi_{\mathcal{W}_s}$.

Verifying (6.26) = (6.27) We will show that:

$$\nabla \chi^\beta(w) = \nabla \chi^\beta \left(\left(\left(1 - \frac{\sum_{a' \in A_{i_a}^-} \dot{w}_{a'}}{K_{i_a} \cdot \sum_{\hat{a} \in A_{i_a}^+} w_{\hat{a}}} \right) \cdot w_a \right)_{a \in A} \right),$$

which is equivalent to showing that (6.26) = (6.27). From (6.29), we have for each $a \in A$:

$$\begin{aligned}
 \frac{\partial \chi^\beta}{\partial w_a} \left(\left(\left(1 - \frac{\sum_{a' \in A_{i_a}^-} h_{a'}(w, p)}{K_{i_a} \cdot \sum_{\hat{a} \in A_{i_a}^+} w_{\hat{a}}} \right) \cdot w_a \right)_{a \in A} \right) &= \frac{1}{\beta} \ln \left(\frac{\left(1 - \frac{\sum_{a' \in A_{i_a}^-} h_{a'}(w, p)}{K_{i_a} \cdot \sum_{\hat{a} \in A_{i_a}^+} w_{\hat{a}}} \right) w_a}{\sum_{\bar{a} \in A_{i_a}^+} \left(1 - \frac{\sum_{a' \in A_{i_{\bar{a}}}^-} h_{a'}(w, p)}{K_{i_{\bar{a}}} \cdot \sum_{\hat{a} \in A_{i_{\bar{a}}}^+} w_{\hat{a}}} \right) w_{\bar{a}}} \right) \\
 &= \frac{1}{\beta} \ln \left(\frac{\left(1 - \frac{\sum_{a' \in A_{i_a}^-} h_{a'}(w, p)}{K_{i_a} \cdot \sum_{\hat{a} \in A_{i_a}^+} w_{\hat{a}}} \right) w_a}{\left(1 - \frac{\sum_{a' \in A_{i_a}^-} h_{a'}(w, p)}{K_{i_a} \cdot \sum_{\hat{a} \in A_{i_a}^+} w_{\hat{a}}} \right) \cdot \sum_{\bar{a} \in A_{i_a}^+} w_{\bar{a}}} \right) \\
 &= \frac{1}{\beta} \ln \left(\frac{w_a}{\sum_{\bar{a} \in A_{i_a}^+} w_{\bar{a}}} \right) \\
 &= \frac{\partial \chi^\beta}{\partial w_a}(w).
 \end{aligned}$$

The second equality above follows because, for each $\bar{a} \in A_{i_a}^+$, we have $i_{\bar{a}} = i_a$. This verifies that (6.26) = (6.27).

Above, we proved that the map F strictly decreases along any trajectory originating in $\mathcal{W} \setminus \{\bar{w}^\beta\}$ and following the best-response dynamics 6.22. The convergence of the dynamics (6.22) to the Condensed DAG equilibrium (6.1.6) now follows by invoking either Sandholm, Corollary 7.B.6 [93], or Sastry, Proposition 5.22 and Theorem 5.23 (LaSalle's Principle and its corollaries) [94]. \square

Theorem 6.3.3. *There exists some $\delta > 0$ such that for any initial $p(0) \in \mathbb{R}^{|A_O|}$ satisfying $\|p(0) - \bar{p}\|_2 < \delta$, where $\bar{p} \in \mathbb{R}^{|A_O|}$ denotes the socially optimal toll, and $\|\cdot\|_*$, the continuous-time dynamical system for the evolution of the tolls $p(t) \in \mathbb{R}^{|A_O|}$:*

$$\dot{p}_{[a]} = -p_{[a]} + \tilde{w}_{[a]}^\beta(p) \cdot \frac{ds_{[a]}}{dw}(\tilde{w}_{[a]}^\beta(p)) \quad (6.30)$$

converges to $\bar{p} \in \mathbb{R}^{|A_O|}$.

Proof. Define $\psi^\beta : \mathbb{R}^{|A_O|} \rightarrow \mathbb{R}^{A_O}$ by:

$$\psi_{[a]}^\beta(p) := \bar{w}_{[a]}^\beta(p) \cdot \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(p)),$$

for each $p \in \mathbb{R}^{|A_O|}$ and $[a] \in A_O$.

First, we claim that the eigenvalues of $\frac{d\psi^\beta}{dp}$ are real and strictly negative, and thus lie on the open left half complex plane. To see this, note that:

$$\frac{d\psi^\beta}{dp}(p) = D(p) \frac{d\tilde{w}^\beta}{dp}(p), \quad (6.31)$$

where $D(p) \in \mathbb{R}^{|A_O| \times |A_O|}$ is a diagonal matrix with diagonal entries:

$$D_{[a][a]}(p) = \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(p)) + \bar{w}_{[a]}^\beta(p) \cdot \frac{d^2s_{[a]}}{dw^2}(\bar{w}_{[a]}^\beta(p))$$

for each $[a] \in A_O$. Our approach for analyzing (6.31) will be to show that $D(p)$ and $\frac{d\tilde{w}^\beta}{dp}(p)$ are symmetric positive definite and symmetric negative definite, respectively. To this end, note that since $s_{[a]}$ is strictly convex and increasing for each $[a] \in A_O$, each $D_{[a][a]}(p)$ is non-negative, so $D(p)$ is symmetric positive definite.

Meanwhile, to show that $\frac{d\tilde{w}^\beta}{dp}$ is symmetric negative semidefinite, let $C \in \mathbb{R}^{|A| \times |A_O|}$ be defined by $C_{a,[a']} := \mathbf{1}\{a \in [a']\}$, where $\mathbf{1}\{\cdot\}$ is the indicator function that returns 1 when the input argument is true, and 0 otherwise. (Note that C has full column rank, due to the surjectivity of the map $\pi : a \mapsto [a]$). Then:

$$\frac{d\tilde{w}^\beta}{dp}(p) = C^\top \frac{d\tilde{w}^\beta}{dw}(p). \quad (6.32)$$

To characterize $\frac{d\tilde{w}^\beta}{dp}(p)$, we invoke the Implicit Function Theorem. As established in the proof of Proposition 6.2.2, there exists a continuously differentiable function $J : \mathbb{R}^{|A_O|} \times \mathbb{R}^{|A|} \rightarrow \mathbb{R}^{|A|}$, and matrices $M \in \mathbb{R}^{|A| \times d}$ and $B \in \mathbb{R}^{|A| \times (|A| - d)}$, such that $J(p, w^\beta(p)) = 0$ for each $p \in \mathbb{R}^{|A_O|}$, the columns of B and the columns of M are orthonormal, $R(M)$ and $R(B)$ are orthogonal subspaces whose direct sum is $\mathbb{R}^{|A|}$, and:

$$\frac{\partial J}{\partial w}(w, p) = \begin{bmatrix} M^\top \\ B^\top \nabla_w^2 F(w, p) \end{bmatrix} \in \mathbb{R}^{|A| \times |A|},$$

where, as in the proof of Proposition 6.2.2, $F : \mathcal{W} \times \mathbb{R}^{|A_o|} \rightarrow \mathbb{R}$ is given by:

$$F(w, p) = \sum_{[a] \in A_o} \int_0^{w_{[a]}} [s_{[a]}(u) + p_{[a]}] du + \frac{1}{\beta} \sum_{i \neq d} \left[\sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right].$$

Thus, the Implicit Function Theorem implies that:

$$\begin{aligned} \frac{d\bar{w}^\beta}{dp}(p) &= - \left[\frac{\partial J}{\partial w}(\bar{w}^\beta(p), p) \right]^{-1} \frac{\partial J}{\partial p}(\bar{w}^\beta(p), p) \\ &= - \begin{bmatrix} M^\top \\ B^\top \nabla_w^2 F(\bar{w}^\beta(p), p) \end{bmatrix}^{-1} \begin{bmatrix} O \\ B^\top \frac{d}{dp} \nabla_w F(\bar{w}^\beta(p), p) \end{bmatrix}, \end{aligned} \quad (6.33)$$

where $\nabla_w F(w, p) \in \mathbb{R}^{|A|}$, and $\frac{\partial}{\partial p} \nabla_w F(w, p) \in \mathbb{R}^{|A| \times |A_o|}$. To study (6.33) further, we wish to rewrite the $B^\top \nabla_w^2 F(w, p)$ term. To this end, note that since $[M \ B] \in \mathbb{R}^{|A| \times |A|}$ is an orthogonal matrix, and $\nabla_w^2 F(w, p)$ is symmetric positive definite (since $F(p, \cdot)$ is strictly convex for each $p \in \mathbb{R}^{|A_o|}$), the matrix:

$$Q := \begin{bmatrix} M^\top \\ B^\top \end{bmatrix} \nabla_w^2 F(\bar{w}^\beta(p), p) [M \ B] \in \mathbb{R}^{|A| \times |A|}$$

is symmetric positive definite as well. Now, let $Q_{11} := M^\top \nabla_w^2 F(w, p) M \in \mathbb{R}^{d \times d}$, $Q_{12} := M^\top \nabla_w^2 F(w, p) B \in \mathbb{R}^{d \times (|A| - d)}$, and $Q_{22} := B^\top \nabla_w^2 F(w, p) B \in \mathbb{R}^{(|A| - d) \times (|A| - d)}$ denote the various block matrices of Q , as shown below:

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^\top & Q_{22} \end{bmatrix}.$$

We then have:

$$\begin{aligned} B^\top \nabla_w^2 F(w, p) &= [O \ I] \begin{bmatrix} M^\top \\ B^\top \end{bmatrix} \nabla_w^2 F(w, p) \\ &= [O \ I] Q \begin{bmatrix} M^\top \\ B^\top \end{bmatrix} \\ &= Q_{12}^\top M^\top + Q_{22} B^\top, \end{aligned}$$

where the matrices O and $i \in I$ above are the zero matrix of dimension $(|A| - d) \times d$ and identity matrix of dimension $(|A| - d) \times (|A| - d)$, respectively. Substituting back into (6.33), we obtain:

$$\begin{aligned} \frac{d\bar{w}^\beta}{dp}(p) &= - \begin{bmatrix} M^\top \\ B^\top \nabla_w^2 F(w, p) \end{bmatrix}^{-1} \begin{bmatrix} O \\ B^\top \frac{d}{dp} \nabla_w F(w, p) \end{bmatrix} \\ &= - \begin{bmatrix} M^\top \\ Q_{12}^\top M^\top + Q_{22} B^\top \end{bmatrix}^{-1} \begin{bmatrix} O \\ B^\top \frac{d}{dp} \nabla_w F(w, p) \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
 &= - \left(\begin{bmatrix} I & O \\ Q_{12}^\top & Q_{22} \end{bmatrix} \begin{bmatrix} M^\top \\ B^\top \end{bmatrix} \right)^{-1} \begin{bmatrix} O \\ B^\top \end{bmatrix} \frac{d}{dp} \nabla_w F(w, p) \\
 &= - [M \ B] \begin{bmatrix} I & O \\ -Q_{22}^{-1} Q_{12}^\top & Q_{22}^{-1} \end{bmatrix} \begin{bmatrix} O \\ B^\top \end{bmatrix} \frac{d}{dp} \nabla_w F(w, p) \tag{6.34} \\
 &= -BQ_{22}^{-1}B^\top C \tag{6.35} \\
 &= -B(B^\top \nabla_w^2 F(w, p)B)^{-1}B^\top C,
 \end{aligned}$$

where the computation of $\frac{d}{dp} \nabla_w F(w, p)$ to show the equality between (6.34) and (6.35) follows by observing that, since for each $a \in A$:

$$\frac{\partial}{\partial w_a} F(w, p) = s_{[a]}(w_{[a]}) + p_{[a]} + \frac{1}{\beta} \ln w_a - \frac{1}{\beta} \ln \left(\sum_{a' \in A_{ia}^+} w_{a'} \right),$$

we have $(\frac{d}{dp} \nabla_w F(w, p))_{a, [a']} = \mathbf{1}\{a \in [a']\} = C_{a, [a']}$, i.e., $\frac{d}{dp} \nabla_w F(w, p) = C$. Together, (6.32) and (6.35) give:

$$\begin{aligned}
 \frac{d\tilde{w}^\beta}{dp}(p) &= C^\top \frac{d\tilde{w}^\beta}{dp}(p) = -C^\top BQ_{22}^{-1}B^\top C \\
 &= -C^\top B(B^\top Q_{22}B)^{-1}B^\top C,
 \end{aligned}$$

This establishes that $\frac{d\tilde{w}^\beta}{dp}(p)$ is symmetric negative semidefinite.

Returning to our original objective of showing that the eigenvalues of $\frac{d\psi^\beta}{dp}$ are negative, recall that from (6.31), we have:

$$\frac{d\psi^\beta}{dp}(p) = D(p) \frac{d\tilde{w}^\beta}{dp}(p).$$

Now, consider the Lyapunov function $V : \mathbb{R}^{|A_0|} \rightarrow \mathbb{R}$, given by:

$$V(p) := (p - \bar{p})^\top D(\bar{p})^{-1} (p - \bar{p}).$$

Given a square matrix M of any dimension, let $\sigma_{\min}(M)$ denote the smallest singular values of M , respectively. Let $\delta_1 > 0$ be given such that, for each $\hat{p} \in \mathbb{R}^{|A_0|}$ satisfying $\|\hat{p} - \bar{p}\|_2 \leq \delta_1$:

$$\begin{aligned}
 \left\| \frac{d\psi^\beta}{dp}(\hat{p}) \right\|_2 &\leq 2 \left\| \frac{d\psi^\beta}{dp}(\bar{p}) \right\|_2, \\
 \|D(\hat{p})^{-1} - D(\bar{p})^{-1}\|_2 &\leq \frac{\sigma_{\min}(D(\bar{p})^{-1})}{4 \left\| \frac{d\psi^\beta}{dp}(\bar{p}) \right\|_2}.
 \end{aligned}$$

Set:

$$\delta := \frac{\sigma_{\min}(D(\bar{p})^{-1})}{\left\| D(\bar{p})^{-1} \right\|_2} \cdot \delta_1 \in (0, \delta_1].$$

Then, for any $\hat{p} \in \mathbb{R}^{|A_0|}$ such that $\|\hat{p} - \bar{p}\|_2 < \delta$, we have:

$$\begin{aligned} \|\hat{p} - \bar{p}\|_2 &< \delta_1, \\ V(p) &< \|D(\bar{p})^{-1}\|_2 \cdot \delta = \sigma_{\min}(D(\bar{p})^{-1}) \cdot \delta_1, \end{aligned}$$

and:

$$\begin{aligned} \dot{V} &= \frac{dV^\top}{dp} \dot{p} \\ &= 2(p - \bar{p})^\top D(\bar{p})^{-1} \dot{p} \\ &= 2(p - \bar{p})^\top D(\bar{p})^{-1} (-p + \psi^\beta(p)) \\ &= -2(p - \bar{p})^\top D(\bar{p})^{-1} (p - \bar{p}) + 2(p - \bar{p})^\top D(\bar{p})^{-1} (-\bar{p} + \psi^\beta(p)) \\ &= -2V + 2(p - \bar{p})^\top D(\bar{p})^{-1} (\psi^\beta(p) - \bar{p}) \\ &= -2V + 2(p - \bar{p})^\top D(\bar{p})^{-1} (\psi^\beta(p) - \psi^\beta(\bar{p})) \\ &= -2V + 2(p - \bar{p})^\top D(\bar{p})^{-1} \left(\int_0^1 \frac{d\psi^\beta}{dp}(\bar{t}\bar{p} + (1-t)p) dt \right) (p - \bar{p}) \\ &= -2V + 2(p - \bar{p})^\top \left(\int_0^1 D(\bar{p})^{-1} D(\bar{t}\bar{p} + (1-t)p) \frac{d\tilde{w}^\beta}{dp}(\bar{t}\bar{p} + (1-t)p) dt \right) (p - \bar{p}) \\ &= -2V + 2(p - \bar{p})^\top \left(\int_0^1 \frac{d\tilde{w}^\beta}{dp}(\bar{t}\bar{p} + (1-t)p) dt \right) (p - \bar{p}) \\ &\quad + 2(p - \bar{p})^\top \left(\int_0^1 \left[D(\bar{p})^{-1} - D(\bar{t}\bar{p} + (1-t)p)^{-1} \right] \right. \\ &\quad \left. D(\bar{t}\bar{p} + (1-t)p) \frac{d\tilde{w}^\beta}{dp}(\bar{t}\bar{p} + (1-t)p) dt \right) (p - \bar{p}) \\ &\leq -2V + 0 + 2\|p - \bar{p}\|_2^2 \cdot \frac{\sigma_{\min}(D(\bar{p})^{-1})}{4 \left\| \frac{d\psi^\beta}{dp}(\bar{p}) \right\|_2} \cdot 2 \left\| \frac{d\psi^\beta}{dp}(\bar{p}) \right\|_2 \\ &= -2V + \sigma_{\min}(D(\bar{p})^{-1}) \|p - \bar{p}\|_2^2 \\ &\leq -2V + V \\ &\leq -V. \end{aligned}$$

We now claim that if $p(0) \in \mathbb{R}^{|A_0|}$ satisfies $V(p(0)) \leq \epsilon := \sigma_{\min}(D(\bar{p})^{-1}) \cdot \delta_1$, then we indeed have $V(p(t)) < \epsilon$ for each $t \geq 0$, and thus $\dot{V} \leq -V$ for each $t \geq 0$, with equality if and only if $p = \bar{p}$. To see this, suppose by contradiction that $V(p(t_1)) \geq \epsilon$ for some $t_1 \geq 0$. Since $V(p(t))$ is continuously differentiable, and $V(p(0)) < \epsilon$, we have $\tau := \inf\{t \geq 0 : \dot{V} > 0\} \in [0, t_1)$. Thus, for each $t \in [0, \tau)$, we have $\dot{V} \leq 0$; since $V(p(0)) < \epsilon$, we have $V(\tau) \leq V(p(0)) < \epsilon$. The continuity of V then implies the existence of some $\delta > 0$ such that $V(p(t)) < \epsilon$ for each $t \in [0, \tau + \delta]$, and thus $\dot{V} \leq 0$ for each $t \in [0, \tau + \delta]$,

contradicting the definition of τ . Thus, if $V(p(0)) = (p(0) - \bar{p})^\top D(\bar{p})^{-1}(p(0) - \bar{p}) < \epsilon$, we have $\dot{V} \leq -V$ for all $t \geq 0$, with equality if and only if $p(t) = \bar{p}$. This establishes the local convergence (in fact, at an exponential rate) of $p(t)$ to \bar{p} , as claimed. \square

To conclude the proof of Theorem 6.3.1, it remains to check that the discrete-time dynamics (6.14)-(6.15), and the continuous-time dynamics (6.17)-(6.18), satisfy the technical conditions in Lemmas 6.3.4 and 6.3.5. In particular, Lemma 6.3.4 establishes that flows and tolls are uniformly bounded across the arc and time indices, while Lemma 6.3.5 asserts that the continuous-time flow and toll dynamics maps are Lipschitz continuous.

Lemma 6.3.4. *The discrete-time flow and toll dynamics induced by (6.14)-(6.15) satisfy:*

1. For each $a \in A$: $\{M_a[n+1] : n \geq 0\}$ is a martingale difference sequence with respect to the filtration $\mathcal{F}_n := \sigma(\cup_{a \in A} (W_a[1], \xi[1], p[1], \dots, W_a[n], \xi[n], p[n]))$.
2. There exist $C_w, C_m, C_p > 0$, such that, for **any** node-dependent constants $\{K_i : i \in I\}$, each $a \in A$ and each $n \geq 0$, we have $W_a[n] \in [C_w, g_o]$, $P_a[n] \in [0, C_p]$, $|M_a[n]| \leq C_m$.

Likewise, the continuous-time flow and toll dynamics induced by (6.17) and (6.18) satisfy the following condition—For each $a \in A$, $t \geq 0$, we have $w_a(t) \in [C_w, g_o]$ and $p_a(t) \in [0, C_p]$.

Proof. Please see [20], Appendix B.2. \square

Lemma 6.3.5. *The continuous-time flow dynamics (6.16) and toll dynamics (6.18) satisfy:*

1. The map $\bar{\xi}^\beta : \mathbb{R}^{|A_o|} \rightarrow \mathbb{R}^{|A|}$ is Lipschitz continuous.
2. For each $a \in A$, the restriction of the cost-to-go map $z_a : \mathcal{W} \times \mathbb{R}^{|A_o|} \rightarrow \mathbb{R}$ to the set of realizable flows and tolls, i.e., $\mathcal{W}' \times [0, C_p]^{|A_o|}$, is Lipschitz continuous.
3. The map from the probability transitions $\xi \in \prod_{i \in I \setminus \{d\}} \Delta(A_i^+)$ and the traffic flows $w \in \mathcal{W}$ is Lipschitz continuous.
4. For each $a \in A$, the restriction of the continuous dynamics transition map $\rho_a : \mathbb{R}^{|A|} \times \mathbb{R}^{|A_o|} \rightarrow \mathbb{R}^{|A|}$, defined recursively by:

$$\rho_a(\xi, p) := K_{i_a} \left(-\xi_a + \frac{\exp(-\beta z_a(w, p))}{\sum_{a' \in A_{i_a}^+} \exp(-\beta z_{a'}(w, p))} \right).$$

to the set of realizable flows and tolls, i.e., $\mathcal{W}' \times [0, C_p]^{|A_o|}$, is Lipschitz continuous.

5. For each $a \in A$, the map $r_{[a]} : \mathbb{R}^{|A_o|} \times \mathbb{R}^{|A_o|}$, defined by:

$$r_{[a]}(p) := -p_{[a]} + \bar{w}_{[a]}^\beta(p) \cdot \frac{ds_{[a]}}{dw}(\bar{w}_{[a]}^\beta(p)), \quad \forall a \in A,$$

is Lipschitz continuous.

Proof. Please see [20], Appendix B.3. \square

Table 6.2: Parameters for simulation.

Notation	Default value
$\theta_{a,0}$	0, 1, 0, 1, 1, 0, 1, 1, 1 (ordered by edge index)
$\theta_{a,1}$	2, 1, 1, 1, 1, 1, 2, 2, 2 (ordered by edge index)
g_1	1
β	10
γ	0.02
$\eta_{i_a}[n]$	Uniform(0, 0.1), $\forall a \in A, i \in I \setminus \{d\}$

6.4 Experiment Results

This section presents experiments that validate the theoretical convergence results of Section 6.3. We present simulation results illustrating that, under (6.14)-(6.15), the traffic flows and tolls converge to a neighborhood of the socially optimal values, as claimed by Theorem 6.3.1.

Consider the network presented in Figure 6.1, following affine latency functions $s_a(w_a) = \theta_{a,1}w_a + \theta_{a,0}$ with parameters given in Table 6.2. To validate Theorem 6.3.1, we evaluate and plot the traffic flow values $W_a[n]$ and toll values $P_a[n]$ on each arc $a \in A$ with respect to discrete time index $n \geq 0$. Figure 6.2 presents traffic flow values at the condensed DAG equilibrium (i.e., w^β) for the original network before and after tolls. Meanwhile, Figure 6.3 and 6.4 illustrate that w and p converge to the condensed DAG equilibrium in approximately 300 iterations. While the original traffic distribution is more concentrated on a few routes, tolls can distribute the traffic more evenly. This shows that tolls can improve overall social welfare by reducing congestion in over-utilized routes.

6.5 Discussion

This work introduces a discrete-time adaptive tolling scheme to minimize the total travel latency in a general traffic network with bidirectional edges. Our model assumes that, at each time, players near-instantaneously react via perturbed best response to the announced tolls. Accordingly, we formulate a two-timescale stochastic dynamical system that describes the joint evolution of traffic flow and tolls. We prove that the fixed point of these dynamics is unique and corresponds to the optimal traffic flow allocation from the perspective of minimizing the total travel time. Moreover, we prove that the stochastic dynamics converges to a neighborhood of the unique fixed point with high probability. Finally, we present simulation results that corroborate our theoretical findings.

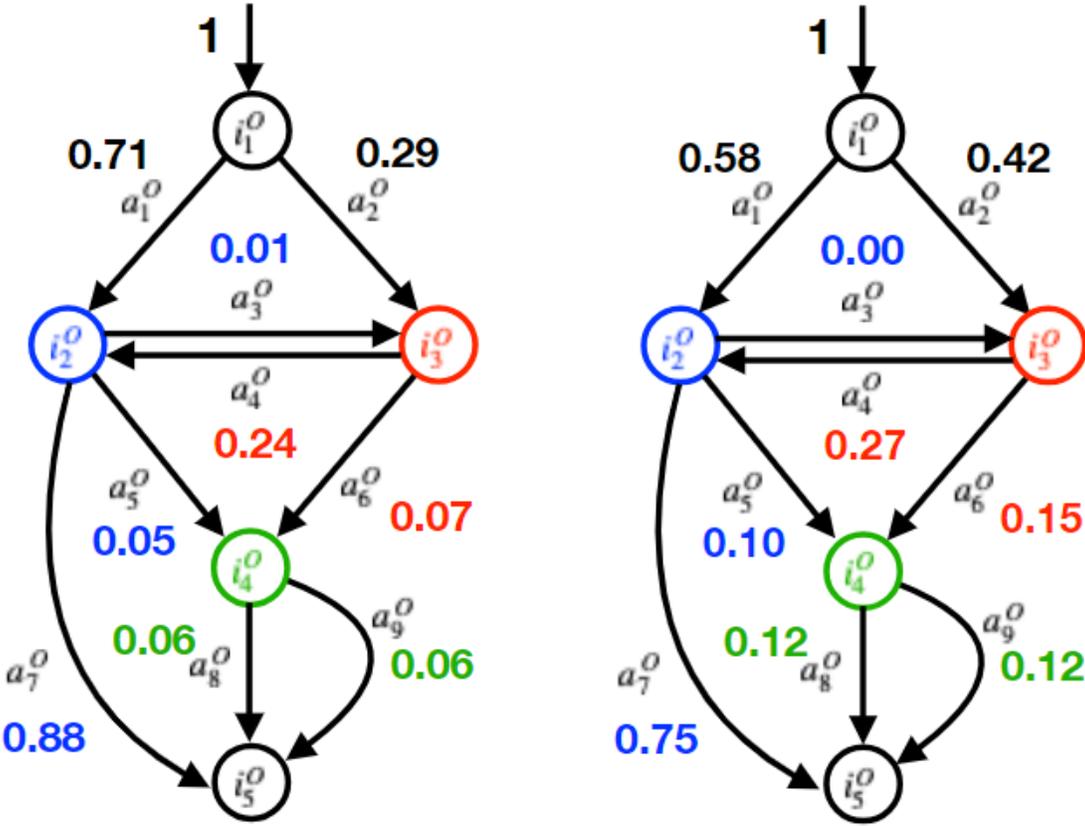


Figure 6.2: Steady state traffic flow on each arc for the original network before (left) and after (right) tolls. Flows on arcs emerging from the same node are identically colored.

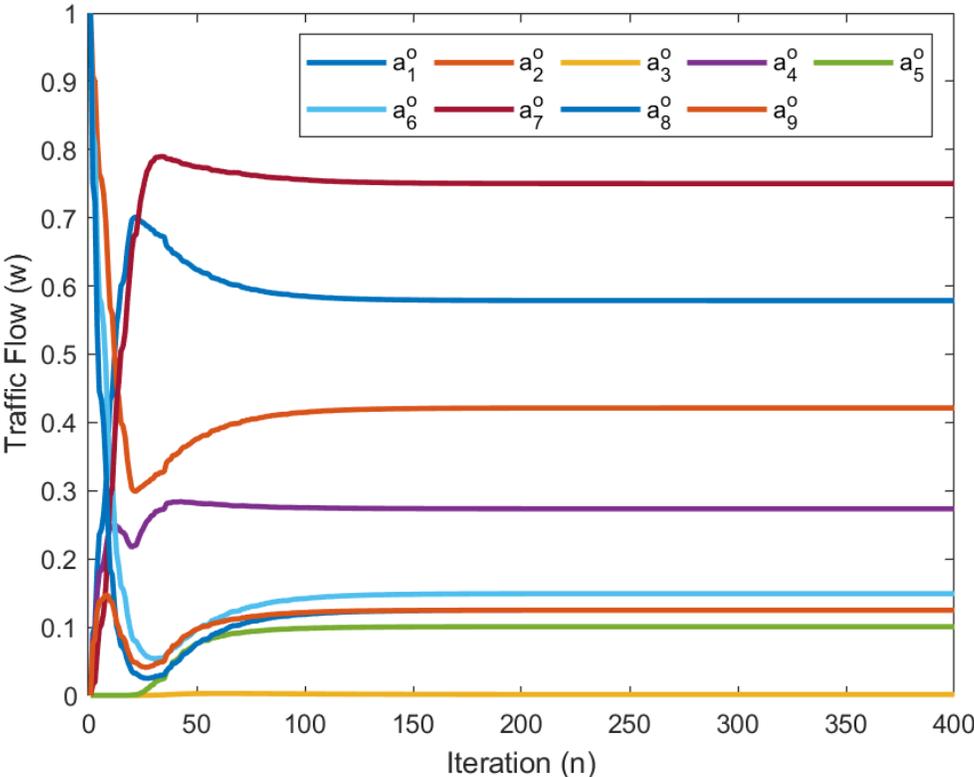


Figure 6.3: Traffic flow W vs. time index n for the condensed DAG in Figure 6.1.

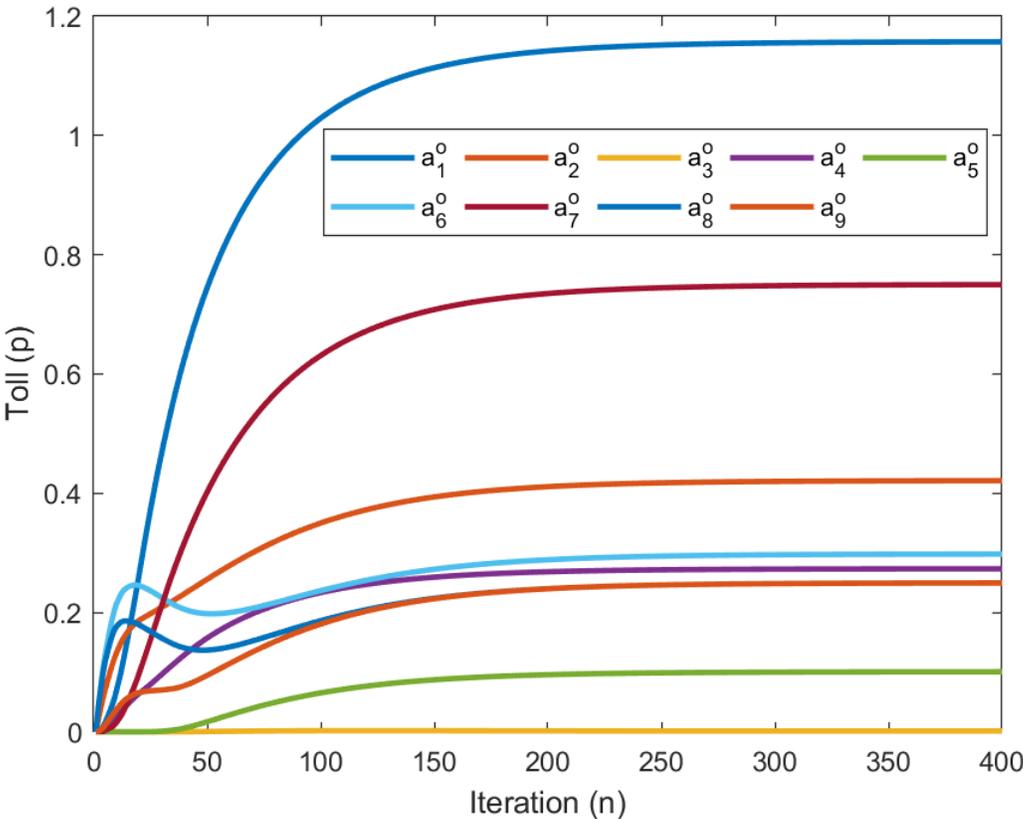


Figure 6.4: Toll P vs. time index n for the condensed DAG in Figure 6.1.

Chapter 7

Online Learning for Adaptive Tolling

Chapter 6 presented an adaptive tolling mechanism over a stochastic *arc-based* traffic assignment model (TAM), in which commuters sequentially select among outgoing arcs at each intermediate node from source to destination. In particular, an entropy parameter $\beta > 0$ is used to characterize stochasticity in the commuters' arc selections. Unfortunately, the deployment of the tolling scheme required the unrealistic assumption that the central tolling authority possesses perfect knowledge of β the network latency functions.

To address this shortcomings, this chapter presents an online learning algorithm in the framework of a stochastic, arc-based TAM, to simultaneously learn the latency function and the entropy parameter, while implementing tolls that become increasingly effective at reducing overall congestion in subsequent iterations. At each iteration, we first implement tolls, constructed during the most recent iteration, on each arc in the network. We then collect the resulting equilibrium traffic flow and latency data from each arc, and apply a regularized least-squares method to update our estimates of the latency function parameters, based on the collected data. In turn, the flow data and latency function estimates can then be used to update our estimate of the entropy parameter β , using the Principle of Optimism in the Face of Uncertainty. Finally, these improved estimates of the latency function and entropy parameters are used to design an improved tolling strategy for the next iteration.

We define the stage-wise regret of our algorithm at each iteration t to be the difference between the following two quantities: (a) The overall latency in the network induced by equilibrium flows corresponding to the toll implemented at iteration t , and (b) The minimum overall latency attainable by the tolling mechanism if it possessed perfect knowledge of the entropy parameter and each arc latency function. The cumulative regret is then computed by summing the stage-wise regret across all iterations. Our algorithm incurs regret of order $O(\sqrt{T} \ln(T)|A| \cdot \max\{|I| \ln(|A|/|I|), B\})$, where T denotes the total iteration count, $|A|$ and $|I|$ denote the number of arcs and nodes in the network, respectively, and B denotes the number of arcs in the network used to construct the estimate of the entropy parameter β at each iteration.

On a technical level, our algorithm utilizes concepts familiar to the bandits community, such as the regularized least-squares method for latency function estimation [48, 64, 1],

and the Principle of Optimism in the Face of Uncertainty for entropy parameter estimation and toll design. However, the problem formulation and proof methodologies considered in this work differ significantly from the above literature. First, in our problem setup, the decision maker’s actions are tolls, which induce equilibrium flows through a non-convex map; in turn, the regret is defined from the overall network congestion generated by these equilibrium flows. Similarly, the unknown entropy parameter estimated in our work affects the cumulative regret in a complicated manner. These complex dependencies between the actions, unknown parameters, and regret preclude the direct use of analysis techniques in the bandit literature. Moreover, whereas the decision-maker in [48] estimates latency functions in the context of a route-based TAM and implements optimal flow assignments directly, our work estimates both the latency functions and entropy parameter β of an underlying arc-based TAM, and implements tolls, which in turn induce an equilibrium flow from which the regret is computed. In particular, to estimate the entropy parameter β , we use a novel approximation scheme beyond the methods in [48].

Likewise, various works have investigated the problem of estimating the entropy parameter of softmax models in the context of traffic assignment models or other machine learning problems [84, 82, 91]. However, these approaches usually use heuristic models to approximate the unknown parameter [84, 82], or assume that the overall objective can be written as a convex function of the entropy parameter [91]. These assumptions separate the above methods from our work, since our formulation involves cost and equilibrium models that are highly non-convex in the action variables (tolls) and in the unknown entropy parameter β .

The remainder of the paper is structured as follows. Section 7.1 introduces the traffic network studied throughout the remaining sections, as well as the incentive structures faced by the commuters traversing the network. Section 7.2 presents our online algorithm. An upper bound for the overall regret incurred by this algorithm is given in Section 7.4. Finally, Section 7.5 presents empirical evidence for the theoretical regret bounds on our algorithm, while Section 7.6 summarizes our work and presents avenues for future research. More details can be found in Chiu and Sastry, “Parameter Estimation in Optimal Tolling for Traffic Networks Under the Markovian Traffic Equilibrium” [19].

7.1 Preliminaries

Setup

Let $G = (I, A)$ be a directed acyclic graph that describes a single-origin single-destination traffic network, with I and A denoting the set of nodes and the set of arcs, respectively. For each arc $a \in A$, we denote the start and end nodes of a by i_a and j_a , respectively. For each node $i \in I$, let $A_i^-, A_i^+ \subset A$ denote the set of incoming and outgoing arcs. Let $g_o \geq 0$ denote the traffic flow entering the network G at each iteration.

To traverse the network, commuters sequentially select from outgoing arcs at each intermediate node, from the origin o to the destination d . Each arc $a \in A$ is associated with a

positive, strictly increasing *latency function* $s_a : [0, \infty) \rightarrow [0, \infty)$, which captures the time required to travel through arc a due to congestion produced by the traffic load $w_a \geq 0$, and a *toll* $p_a \geq 0$, the monetary value each traveler must pay to access the arc. Throughout the rest of the paper, we adopt a linear latency model, formally stated in the assumption below¹.

Assumption 7.1.1 (Linear Latency Functions). *For each arc $a \in A$, there exists some $\theta_a \in \mathbb{R}$ such that $s_a(w_a) = \theta_a w_a$.*

The *cost* $c_a : [0, \infty)^3 \rightarrow [0, \infty)$ on each arc is then obtained by summing the travel time and toll:

$$c_a(\theta_a, w_a, p_a) = s_a(w_a) + p_a = \theta_a w_a + p_a,$$

while the *perceived cost* \tilde{c}_a additionally includes a zero-mean stochastic error term $\delta_a \in \mathbb{R}$ that encapsulates variations in commuters' perception of travel time:

$$\tilde{c}_a(\theta_a, w_a, p_a) = s_a(w_a) + p_a + \delta_a = \theta_a w_a + p_a + \delta_a,$$

At every non-destination node $i \in I \setminus \{d\}$, commuters select among outgoing nodes $a \in A_i^+$ by computing their perceived minimum cost-to-go $\{\tilde{z}_a \in \mathbb{R} : a \in A_i^+\}$ on arc a :

$$\tilde{z}_a(\theta, w, p) := \tilde{c}_a(\theta, w_a, p_a) + \mathbb{E}_\delta \left[\min_{a' \in A_{j_a}^+} \tilde{z}_{a'}(\theta, w, p) \right], \quad j_a \neq d, \quad (7.1)$$

$$\tilde{z}_a(\theta, w, p) := \tilde{c}_a(\theta, w_a, p_a), \quad j_a = d.$$

In this work, we adopt the *logit Markovian Model* [2, 3], under which the noise terms δ_a are described by the Gumbel distribution with scale parameter $\beta > 0$. (We also call $\beta > 0$ the entropy parameter.) As a result, the expected cost-to-go z_a for each arc $a \in A$ admits the following closed-form expression:

$$z_a(\theta, w, p) = c_a(\theta_a, w_a, p_a) - \frac{1}{\beta} \ln \left(\sum_{a' \in A_{j_a}^+} e^{-\beta z_{a'}(\theta, w, p)} \right). \quad (7.2)$$

As a result, the corresponding equilibrium flow, called the *Markovian Traffic Equilibrium (MTE)* $\bar{w}^{\theta, \beta}(p) \in \mathbb{R}^{|A|}$ corresponding to the latency function parameters $\theta \in \mathbb{R}^{|A|}$, entropy parameter $\beta > 0$, and toll vector $p \in \mathbb{R}^{|A|}$, is the unique flow vector satisfying the following fixed point equation—For each non-destination node $i \in I \setminus \{d\}$ and outgoing arc $a \in A_i^+$:

$$\bar{w}_a^{\theta, \beta}(\theta, p) = \left(g_i + \sum_{a' \in A_i^+} \bar{w}_{a'}^{\theta, \beta}(\theta, p) \right) \cdot \frac{\exp(-\beta z_a(\theta, \bar{w}^{\theta, \beta}(p), p))}{\sum_{a' \in A_i^+} \exp(-\beta z_{a'}(\theta, \bar{w}^{\theta, \beta}(p), p))} \in \mathcal{W},$$

¹For an extension of our least-squares-based latency function estimation method to higher-degree polynomial latency functions, please see [48].

where \mathcal{W} is the constraint set that enforces the conservation of traffic flow:

$$\mathcal{W} := \left\{ w \in \mathbb{R}^{|A|} : \sum_{a \in A_i^+} w_a = \sum_{a \in A_i^-} w_a, \forall i \neq o, d, \sum_{a \in A_o^+} w_a = g_o, w_a \geq 0, \forall a \in A \right\} \quad (7.3)$$

Socially Optimal Tolls

The objective of toll implementation is to realign commuter's incentives and route selection decisions, to induce *perturbed social optimality* with respect to the logit Markovian model detailed in Section 7.1, as defined below.

Definition 7.1.1 (Perturbed Socially Optimal Flow). *Let the perturbed total weighted latency $L : \mathcal{W} \times \mathbb{R}^{|A|} \times \mathbb{R} \rightarrow \mathbb{R}$ be given by:*

$$L(w, \theta, \beta) := \sum_{a \in A} w_a \ell_a^\theta(w_a) + \frac{1}{\beta^*} \sum_{i \in I \setminus \{d\}} \left[\sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right]. \quad (7.4)$$

We call $w^* \in \mathcal{W}$ the perturbed socially optimal flow with latency parameters θ regularization parameter $\beta > 0$ if it solves $\min_{w \in \mathcal{W}} L(w, \theta, \beta)$, with \mathcal{W} given by (7.3).

In the perturbed total latency L defined above, the first component is the total latency on the network weighted by the traffic load on each arc, while the second component is a non-positive entropy term that achieves its minimum when the traffic load at each non-destination node allocates itself equally among all outgoing arc. Thus, the entropy parameter β weights the total network latency against the tendency of commuters with imperfect information to explore among outgoing arcs at each intermediate node.

Since the minimization problem posed by Definition 7.1.1 is strictly convex, the perturbed socially optimal flow exists and is unique. Moreover, [22, 20] establish that, given a traffic network $G = (I, A)$ with latency function parameters $\theta \in \mathbb{R}^{|A|}$ and entropy parameter $\beta > 0$, there exists an *optimal toll* $\bar{p} \in \mathbb{R}^{|A|}$ whose corresponding MTE $\bar{w}^{\theta, \beta}(\bar{p})$ is perturbed socially optimal, and a dynamic tolling scheme that converges to the optimal toll. Those results, in the context of the online tolling problem considered in this work, are as summarized below. For more details, please see [20].

Proposition 7.1.2. *There exists $\tilde{w} \in \mathcal{W}$ and $\bar{p} \in \mathbb{R}^{|A|}$ such that $\tilde{w} = \bar{w}^{\theta, \beta}(\bar{p})$ and $\bar{p}_a^t = \bar{w}_a^t \cdot \theta_a$ for each $a \in A$. Moreover, \bar{w} is perturbed socially optimal, i.e., $\tilde{w} = \arg \min_{w \in \mathcal{W}} L(w, \theta, \beta)$.*

Online Learning Problem

Here, we pose the online learning problem that forms the central focus of this work. Let T denote the total number of iterations for which the algorithm is run. Consider a traffic network G with known node and arc structure (I, A) , but unknown latency function parameters $\{\theta_a^* : a \in A\}$ and entropy parameter $\beta^* > 0$. We assume that θ^* and β^* are bounded, as posed below.

Assumption 7.1.2 (Parameter Bounds). *There exist constants $c_\theta, C_\theta, c_\beta > 0$ such that $\theta_a^* \in [c_\theta, C_\theta]$ for each $a \in A$, and $\beta^* > c_\beta$. The central authority has access to c_β but not necessarily c_θ or C_θ .*

The above assumptions are not overly restrictive, since roads cannot be arbitrarily congestive, and travelers usually have some non-zero proclivity for selecting cost-minimizing arcs and routes. Moreover, as established in Section 7.2, the arc latency parameter estimation errors $\|\theta^t - \theta^*\|_2$ shrinks rapidly as t increases. This allows the true, unknown temperature parameter β^* , and thus a lower bound for β^* , to be estimated with increasing accuracy as more data is collected.

Now, consider ourselves in the position of a central traffic authority that wishes to minimize the perturbed total latency over the iterations $t \in [T]$, despite initially lacking knowledge of the function parameters $\theta \in \mathbb{R}^{|A|}$, and the underlying entropy parameter β . To accomplish this, at each iteration $t \in [T]$, we implement a toll vector $\hat{p}^t \in \mathbb{R}^{|A|}$, and observe the resulting MTE traffic load allocation $w^t := \bar{w}^{\theta^*, \beta^*}(p^t) \in \mathcal{W}$, as well as the random realizations of the travelers' latencies on each arc:

$$\ell_{a,j}^t = s_a(w_a^t) + \epsilon_{a,j}^t.$$

for each $j \in \llbracket w^t \rrbracket$, where $\epsilon_{a,j}^t$ are independent 1-subGaussian random variables. We then use the flow data $\{w_a^t : a \in A\}$ and the latency data $\{\ell_{a,j}^t : a \in A, j \in \llbracket w^t \rrbracket\}$ to update our estimates of the underlying, unknown latency function parameter θ^* and entropy parameter β^* , and correspondingly design our toll to implement at the next iteration $t + 1$. The cumulative regret R over the iterations $t \in [T]$ is thus given by:

$$R := \sum_{t=1}^T [L(w^{\theta^*, \beta^*}(p^t), \theta^*, \beta^*) - L(w^{\theta^*, \beta^*}(p^*), \theta^*, \beta^*)] \quad (7.5)$$

The core tenet of the above framework is that, as we accumulate more data on the traffic flow and realized latencies, we can construct increasingly accurate estimates of θ^* and β^* , and consequently adapt our tolls p^t to reduce congestion in an increasingly effective manner.

7.2 Main Algorithm

In this section, we present the main components of our algorithm. Section 7.2 describes the least-squares estimator used to approximate the arc latency functions from collected flow data. Section 7.2 then discusses our novel approximation scheme for the unknown entropy parameter β . Finally, we present our main algorithm in Section 7.2.

Least-Squares Estimator for Latency Function Parameters

First, we present the regularized least-squares estimator for the arc latency coefficients $\{\theta_a : a \in A\}$. At each iteration $t \in [T]$, for each arc $a \in A$, we observe the traffic flow at the

current iteration, w_a^t , and latency data $\{\ell_{a,j}^k : a \in A, k \in [t], j \in [w_a^t]\}$. We then update the regularized least-squares estimate $\hat{\theta}_a^t > 0$ for the true coefficient θ_a^* , with regularizer $\lambda_a > 0$, as follows ²:

$$\hat{\theta}_a^t := \arg \min_{\theta_a} \left(\sum_{j=1}^{t-1} \sum_{k=1}^{\lfloor w_a^j \rfloor} (\ell_{a,k}^j - \theta_a w_a^j)^2 + \lambda_a \|\theta_a\|_2^2 \right).$$

The following lemma states that these estimates, across iterations $t \in [T]$, lie within a neighborhood of the true parameter θ_a^* .

Lemma 7.2.1. [64] *For each iteration $t \in [T]$ and arc $a \in A$, define:*

$$\gamma_a^t := \sqrt{\lambda_a} C_\theta + \sqrt{2 \ln T + 2 \ln \left(\frac{V_a^{t-1}}{\lambda_a} \right)}, \quad (7.6)$$

and let the “good event” E be defined by:

$$E := \left\{ \forall t \in [T], \forall a \in A : |\hat{\theta}_a^{t-1} - \theta_a^*| \leq \frac{\gamma_a^t}{\sqrt{V_a^{t-1}}} \right\}.$$

Then $\mathbb{P}(E) \geq 1 - \frac{|A|}{T}$.

Proof. We first introduce some notation. At each iteration $t \in [T]$, for each arc $a \in A$, the regularized least-squares estimate $\hat{\theta}_a^t > 0$ for the true coefficient θ_a^* , with regularizer $\lambda_a > 0$, is given by:

$$\hat{\theta}_a^t := \arg \min_{\theta_a} \left(\sum_{j=1}^{t-1} \sum_{k=1}^{\lfloor w_a^j \rfloor} (\ell_{a,k}^j - \theta_a w_a^j)^2 + \lambda_a \|\theta_a\|_2^2 \right).$$

Note that the cost objective in the above argmin expression is convex and quadratic. Thus, by setting the gradient to 0, we can compute the optimal parameter estimate as follows (for more details, please see Gollapudi et al. [48], Lemma 2):

$$\hat{\theta}_a^t = \left(\lambda_a + \sum_{j=1}^{t-1} (w_a^j)^3 \right)^{-1} \left(\sum_{j=1}^{t-1} w_a^j \cdot \sum_{k=1}^{\lfloor w_a^j \rfloor} \ell_{a,k}^j \right) \quad (7.7)$$

For convenience, we define:

$$V_a^t := \lambda_a + \sum_{j=1}^{t-1} (w_a^j)^3, \quad (7.8)$$

²We assume $\lfloor w \rfloor \geq 1$, i.e., each arc is traversed upon by at least one commuter per iteration.

$$W_a^t := \sum_{j=1}^{t-1} (w_a^j)^3, \quad (7.9)$$

$$U_a^t := \sum_{j=1}^{t-1} w_a^j \cdot \sum_{k=1}^{\lfloor w_a^j \rfloor} \ell_{a,k}^j, \quad (7.10)$$

$$S_a^t := \sum_{j=1}^{t-1} w_a^j \cdot \sum_{k=1}^{\lfloor w_a^j \rfloor} \epsilon_{a,k}^j. \quad (7.11)$$

Thus, we can write (7.7) as:

$$\hat{\theta}_a^t = (V_a^t)^{-1} U_a^t = (V_a^t)^{-1} (W_a^t \theta_a + S_a^t). \quad (7.12)$$

For each arc $a \in A$, the above process generates regularized least-squares estimates $\{\hat{\theta}_a^t\}$, across iterations $t \in [T]$, for the true underlying parameter θ_a^* . The following lemma demonstrates that these estimates, across iterations $t \in [T]$, lie within a neighborhood of the true parameter θ_a^* .

The remainder of the proof parallels that of Gollapudi et al. [48], Lemma 3, and is included for completeness. From (7.12), we have:

$$\begin{aligned} & \sqrt{V_a^t} |\hat{\theta}_a^t - \theta_a^*| \\ &= \sqrt{V_a^t} |(V_a^t)^{-1} (W_a^t \theta_a + S_a^t) - \theta_a| \\ &= \sqrt{V_a^t} |(V_a^t)^{-1} S_a^t + ((V_a^t)^{-1} W_a^t - 1) \theta_a| \\ &= \sqrt{V_a^t} |(V_a^t)^{-1} S_a^t + ((V_a^t)^{-1} (V_a^t - \lambda_a) - 1) \theta_a| \\ &= \sqrt{V_a^t} |(V_a^t)^{-1} S_a^t - \lambda_a (V_a^t)^{-1} \theta_a| \\ &= (V_a^t)^{-1/2} |S_a^t| + \sqrt{\lambda_a} \theta_a. \end{aligned} \quad (7.13)$$

To bound $(V_a^t)^{-1/2} |S_a^t|$, define $M_a^t(z) := \exp(z S_a^t - \frac{1}{2} V_a^t z^2)$ for each $z \in \mathbb{R}$. Then, for any fixed $z \in \mathbb{R}$:

$$\begin{aligned} \mathbb{E}[M_a^t(z) | \mathcal{F}_a^t] &= M_a^{t-1}(z) \cdot \mathbb{E} \left[\exp \left(w_a^t \cdot \sum_{k=1}^{\lfloor w_a^t \rfloor} \epsilon_{a,k}^t z - \frac{1}{2} \lfloor w_a^t \rfloor (w_a^t)^2 z^2 \right) \middle| \mathcal{F}_a^t \right] \\ &= M_a^{t-1}(z) \cdot \prod_{k=1}^{\lfloor w_a^t \rfloor} \mathbb{E} \left[\exp \left(w_a^t \cdot \epsilon_{a,k}^t z - \frac{1}{2} \lfloor w_a^t \rfloor (w_a^t)^2 z^2 \right) \middle| \mathcal{F}_a^t \right] \\ &\leq M_a^{t-1}(z). \end{aligned}$$

so $M_a^t(z)$ is a supermartingale adapted to the filtration $\mathcal{F}_a^t := \sigma(w_a^1, \tilde{s}_a^1)$. Thus, so is $\tilde{M}_a^t := \mathbb{E}_{z \sim \mathcal{N}(0,1)}[M_a^t(z)]$. It thus follows from Lattimore and Szepesvari [64], Theorem 20.4, that:

$$(V_a^t)^{-1/2} |S_a^t| \leq \sqrt{2 \ln t + \ln \left(\frac{V_a^t}{\lambda_a} \right)}. \quad (7.14)$$

The proof now follows from (7.13) and (7.14). \square

In other words, with probability at least $1 - \frac{|A|}{T}$, for each arc $a \in A$ at each iteration $t \in [T]$, the estimate $\hat{\theta}_a^t$ falls within the confidence interval $\left[\hat{\theta}_a^t - \frac{\gamma_a^t}{\sqrt{V_a^{t-1}}}, \hat{\theta}_a^t + \frac{\gamma_a^t}{\sqrt{V_a^{t-1}}} \right]$. Below, for convenience, we set:

$$\begin{aligned}\hat{\theta}_a^{t,-} &:= \hat{\theta}_a^t - \frac{\gamma_a^t}{\sqrt{V_a^t}}, \\ \hat{\theta}_a^{t,+} &:= \hat{\theta}_a^t + \frac{\gamma_a^t}{\sqrt{V_a^t}}\end{aligned}$$

Entropy Parameter Estimation

Intuitively, the entropy parameter governs the degree to which travelers at an intermediate node prefer to select an outgoing arc that minimizes the cost-to-go. Specifically, when $\beta \rightarrow \infty$, travelers at node i select with probability 1 an outgoing arc $a \in A_i^+$ that minimizes the cost-to-go; when $\beta \rightarrow 0$, travelers at node i select from all outgoing arcs with equal probability. As such, a natural approach for estimate β would begin by fixing a node i^* , whose outgoing routes to the destination are relatively straightforward to describe. Then, we can analyze data that characterize the traffic flows and costs among its outgoing arcs at each iteration $t \in [T]$, to gain insight into the strength of the commuters' preference to minimize their cost-to-go, i.e., to estimate β .

We thus begin with the following lemma, which states that regardless of the precise structure of the traffic network G , there must exist a node $i^* \in I$ with properties desirable for estimating $\beta > 0$. For every node i^* satisfying the conditions of Lemma 7.2.2, each outgoing arc $a' \in A_{i^*}^+$ yields exactly one route from i^* to d . Thus, the route segments from i^* to d have structure akin to a parallel-link network, allowing the estimation of the entropy parameter β^* from i^* to be straightforward. Examples are furnished in Figure 7.1.

Lemma 7.2.2. *There exists a node $i^* \in I \setminus \{d\}$ such that $|A_{i^*}^+| \geq 2$, and for each $j \in A_{i^*}^+$, either $j = d$, or there exists only one route from j to d .*

Proof. By assumption, the graph G contains more than one route from the origin o to the destination d . Thus, there exists some $a \in A$ such that $|A_{i_a}^+| \geq 2$, so the quantity:

$$m^* := \min\{m_a : a \in A, |A_{i_a}^+| \geq 2\}$$

is well-defined. Now, fix any $a \in A$ such that $m_a = m^*$, and $|A_{i_a}^+| \geq 2$. It suffices to show that, for each $j \in A_{i_a}^+$, there exists only one route connecting j to the destination d . Suppose by contradiction that there exists some $j' \in A_{i_a}^+$ such that at least two distinct routes connect j' to d . Let $\bar{j} \in I \setminus \{d\}$ denote any node at which these routes diverge. Then for any $\bar{a} \in A_{\bar{j}}^+$, we have $|A_{i_{\bar{a}}}^+| = |A_{\bar{j}}^+| \geq 2$, and:

$$m_{\bar{a}} < m_a = m^*,$$

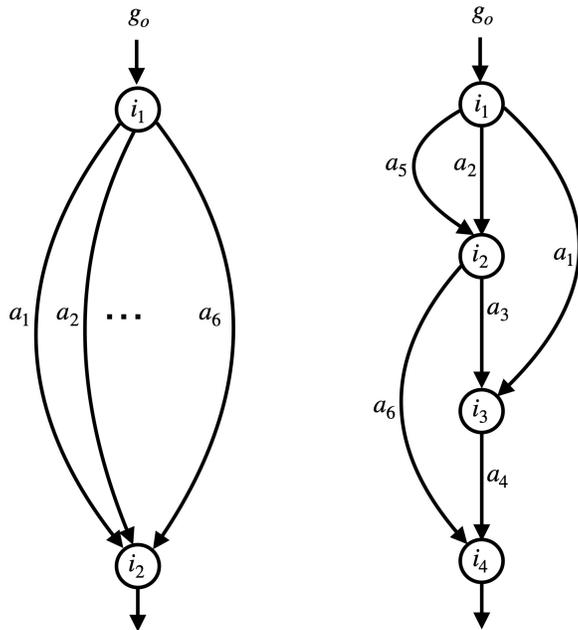


Figure 7.1: (Left) A parallel arc-network with 6 arcs; here, $i^* = i_1$. (Right) A more general network with 6 arcs; here, $i^* = i_2$ since there exists two routes from i_2 to the destination i_4 which do not share an arc.

a contradiction to the definition of m^* . This concludes the proof. □

Below, we present assumptions that facilitate the estimation of the true, unknown temperature parameter β^* . First, for each node $i^* \in I \setminus \{d\}$, and any arc latency parameter estimate $\theta \in \mathbb{R}^{|A|}$ and temperature parameter $\beta > 0$ within a range of reasonable estimates for the true parameters $\theta^* \in \mathbb{R}^{|A|}$ and $\beta^* > 0$, we assume that the MTE costs of the outgoing edges $A_{i^*}^+$ are not identical. In particular, for each such node i^* , among the outgoing arcs A , there must be sufficiently differentiation, in the form of a strictly positive gap $\Delta_z > 0$, between the minimum and maximum costs-to-go. This facilitates the estimation of the temperature parameter in β , and emphasizes its role in the stochastic route choices made on the part of the travelers. Indeed, the temperature parameter β is not meaningful in networks with route segments that are virtually indistinguishable in cost.

Assumption 7.2.1. *Let $\bar{p}(\hat{\theta}, \hat{\beta})$ denote the optimal toll corresponding to an arc-based TAM with entropy parameter $\hat{\beta}$, over a network with latency function parameters θ . There exists $\Delta_z > 0$, such that, for any node $i^* \in I$ satisfying the conditions of Lemma 7.2.2, and any parameter estimates within known bounds, $\hat{\theta} \in [c_\theta, C_\theta]$ and $\hat{\beta} \in [c_\beta, \infty)$, we have:*

$$\max_{a' \in A_{i^*}^+} z_{a'}(\hat{\theta}, \bar{w}^{\theta^*, \beta^*}(\bar{p}(\hat{\theta}, \hat{\beta})), \bar{p}(\hat{\theta}, \hat{\beta}))$$

$$- \min_{a' \in A_{i^*}^+} z_{a'}(\theta, \bar{w}^{\theta^*, \beta^*}(\bar{p}(\hat{\theta}, \hat{\beta})), \bar{p}(\hat{\theta}, \hat{\beta})) \geq \Delta_z.$$

In the following lemma, we establish an estimator β^t for the temperature parameter β at each iteration t whose proximity to the true temperature parameter β^* is directly proportional to the gap between the under- and over-estimators $\theta^{t,-} \in \mathbb{R}^{|A|}$ and $\theta^{t,+} \in \mathbb{R}^{|A|}$ of the true arc latency parameter θ^* . The key intuition behind the estimator is that, if the true latency function parameters θ^* on each arc were known, the underlying entropy parameter β^* can be perfectly recovered by comparing the flows of outgoing arcs at a non-destination node, and the ratios between the costs-to-go of these arcs. However, since the central authority lacks access to θ^* , we instead use the upper and lower bounds of the confidence interval at each iteration t , i.e., $\{\theta_a^{t,+}, \theta_a^{t,-} : a \in A\}$, to construct an estimate β^t of the underlying, unknown entropy parameter β^* . Moreover, we construct the estimate β^t to provably under-approximate β^* , i.e., to guarantee that $\beta^t \leq \beta^*$. This can be viewed as an extension of the Principle of Optimism in the Face of Uncertainty, since the total latency (7.4) is non-decreasing in the entropy parameter β (Recall that the entropy term, to which the $1/\beta^*$ factor is multiplied, is always non-positive).

Lemma 7.2.3. *Let $i^* \in I \setminus \{d\}$ be any node satisfying the conditions in Lemma 7.2.2, and let:*

$$a^* \in \arg \min_{a' \in A_{i^*}^+} z_{a'}(\theta^*, w^t, p^t). \quad (7.15)$$

Then there exists $\beta^t \in [c_\beta, \beta^*]$ such that:

$$\frac{\exp(-\beta^t \cdot z_{a^*}(\theta^{t,-}, w^t, p^t))}{\sum_{a' \in A_{i^*}^+} \exp(-\beta^t \cdot z_{a'}(\theta^{t,+}, w^t, p^t))} = \frac{\exp(-\beta^* \cdot z_{a^*}(\theta^*, w^t, p^t))}{\sum_{a' \in A_{i^*}^+} \exp(-\beta^* \cdot z_{a'}(\theta^*, w^t, p^t))} \quad (7.16)$$

Moreover, let $A(i^*)$ denote the set of all arcs contained in a route from i^* to d . Then:

$$|\beta^t - \beta^*| \leq \frac{\beta^* g_o}{\Delta_z} \cdot \sum_{a \in A(i^*)} (\theta_a^{t,+} - \theta_a^{t,-}) w_a^t. \quad (7.17)$$

Proof. Fix $t \in [T]$. Define $\kappa_{a^*}^t \in \mathbb{R}$ by:

$$\kappa_{a^*}^t := \frac{\exp(-\beta^* \cdot z_{a^*}(\theta^*, w^t, p^t))}{\sum_{a' \in A_{i^*}^+} \exp(-\beta^* \cdot z_{a'}(\theta^*, w^t, p^t))} = \frac{w_{a^*}^t}{\sum_{a' \in A_{i^*}^+} w_{a'}^t},$$

and let $f^t, g^t : \mathbb{R} \times \mathbb{R}^{|A|} \times \mathbb{R}^{|A|} \rightarrow \mathbb{R}$ be given as follows:

$$f^t(\beta, \theta^+, \theta^-) := \frac{\exp(-\beta^t \cdot z_{a^*}(\theta^{t,-}, w^t, p^t))}{\sum_{a' \in A_{i^*}^+} \exp(-\beta^t \cdot z_{a'}(\theta^{t,+}, w^t, p^t))},$$

$$g^t(\beta, \theta^+, \theta^-) := \ln f^t(\beta, \theta^+, \theta^-) - \ln \kappa_{a^*}^t$$

$$= -\beta \cdot z_{a^*}(\theta^-, w^t, p^t) - \ln \left(\sum_{a' \in A_{i^*}^+} \exp(-\beta \cdot z_{a'}(\theta^+, w^t, p^t)) \right) - \ln \kappa_{a^*}^t$$

Note that $g^t(\beta^t, \theta^+, \theta^-) = 0$ holds if and only if $f^t(\beta^t, \theta^+, \theta^-) = \kappa_{a^*}^t$. If one takes $\theta^+ = \theta^{t,+}$ and $\theta^- = \theta^{t,-}$ this becomes a restatement of (7.16). We note that $z_a(\theta, w, p)$ is continuously differentiable for each $a \in A$, $\theta \in \mathbb{R}^{|A|}$, $w \in \mathcal{W}$, and $p \in \mathbb{R}^{|A|}$, and the log-sum-exp function is continuously differentiable in the entropy parameter β . Thus, f^t and g^t are likewise continuously differentiable at each $\beta > 0$ and each $\theta^+, \theta^- \in \mathbb{R}^{|A|}$.

The remainder of the proof proceeds in two parts. We first prove that, given any fixed values $\theta_a^+ \geq \theta_a^*$, $\theta_a^- \leq \theta_a^*$ for each $a \in A$, there exists a unique fixed point solution β to the function $g^t(\beta^t, \theta^+, \theta^-) = 0$. In particular, given $\theta_a^{t,+} \geq \theta_a^*$, $\theta_a^{t,-} \leq \theta_a^*$ for each $a \in A$, there exists a unique entropy parameter estimate $\beta^t > 0$ that solves $g^t(\beta^t, \theta^{t,+}, \theta^{t,-}) = 0$, i.e., that satisfies (7.16), and β^* is the unique entropy parameter value that satisfies $g^t(\beta^*, \theta^*, \theta^*) = 0$. We then bound the gap between β^* and β by bounding the difference between $\theta^{t,+}$ and θ^* , and between $\theta^{t,-}$ and θ^* .

1. Claim—Given any fixed $\theta_a^+ \geq \theta_a^*$, $\theta_a^- \leq \theta_a^*$ for each $a \in A$, there exists a unique fixed point solution β to the function $g^t(\beta^t, \theta^+, \theta^-) = 0$:

Proof: To show that, for any $\theta^{t,+}, \theta^{t,-} \in \mathbb{R}^{|A|}$, the fixed-point equation $g^t(\beta^*, \theta^{t,+}, \theta^{t,-}) = 0$, has a unique solution (or equivalently that $f^t(\beta, \theta^+, \theta^-) = \kappa_{a^*}^t$ has a unique solution), we first note that:

$$\begin{aligned} \frac{1}{|A_{i^*}^+|} &\leq \kappa_{a^*}^t \\ &= \frac{\exp(-\beta^* \cdot z_{a^*}(\theta^*, w^t, p^t))}{\sum_{a' \in A_{i^*}^+} \exp(-\beta^* \cdot z_{a'}(\theta^*, w^t, p^t))} \\ &< 1. \end{aligned}$$

and that $f^t(0, \theta^+, \theta^-) = 1/|A_{i^*}^+|$. Below, we establish that $\lim_{\beta \rightarrow \infty} f^t(\beta, \theta^+, \theta^-) = 1/|A_{i^*}^+|$, by lower bounding $\frac{\partial g^t}{\partial \beta}$. The existence and uniqueness of a solution β to the fixed-point equation $f^t(\beta, \theta^+, \theta^-) = \kappa_{a^*}^t$ then follows from the Intermediate Value Theorem.

To compute derivatives of g^t , we observe that, since $i_{a^*} = i^*$ satisfies the conditions of Lemma 7.2.2, for each $a' \in A_{i^*}^+$, there exists exactly one route that connects $j_{a'}$ and d . As a result, $z_{a'}(\theta^+, w^t, p^t)$ equals the sum of latencies on a' and on arcs comprising that route, and therefore does not depend on the entropy parameter β . Thus:

$$\frac{\partial g^t}{\partial \beta}(\beta, \theta^+, \theta^-) = -z_{a^*}(\theta^-, w^t, p^t) + \frac{\sum_{a' \in A_{i^*}^+} e^{-\beta \cdot z_{a'}(\theta^{t,-}, w^t, p^t)} \cdot z_{a'}(\theta^{t,-}, w^t, p^t)}{\sum_{a' \in A_{i^*}^+} e^{-\beta \cdot z_{a'}(\theta^{t,-}, w^t, p^t)}}$$

$$\begin{aligned}
&= -z_{a^*}(\theta^-, w^t, p^t) + \sum_{\bar{a} \in A_{i^*}^+} \frac{e^{-\beta \cdot z_{\bar{a}}(\theta^{t,-}, w^t, p^t)}}{\sum_{a' \in A_{i^*}^+} e^{-\beta \cdot z_{a'}(\theta^{t,-}, w^t, p^t)}} \cdot z_{\bar{a}}(\theta^{t,-}, w^t, p^t) \\
&= \sum_{\bar{a} \in A_{i^*}^+} \frac{e^{-\beta \cdot z_{\bar{a}}(\theta^{t,-}, w^t, p^t)}}{\sum_{a' \in A_{i^*}^+} e^{-\beta \cdot z_{a'}(\theta^{t,-}, w^t, p^t)}} \cdot \left[z_{\bar{a}}(\theta^{t,-}, w^t, p^t) - z_{a^*}(\theta^{t,-}, w^t, p^t) \right] \\
&= \sum_{\bar{a} \in A_{i^*}^+} \frac{w_{\bar{a}}^t}{\sum_{a' \in A_{i^*}^+} w_{a'}^t} \cdot \left[z_{\bar{a}}(\theta^{t,-}, w^t, p^t) - z_{a^*}(\theta^{t,-}, w^t, p^t) \right].
\end{aligned}$$

The flow continuity equations imply that $\sum_{a' \in A_{i^*}^+} w_{a'}^t \leq g_o$; together with the assumption that $w_a^t \geq 1$ for each $a \in A$, we have:

$$\frac{w_{\bar{a}}^t}{\sum_{a' \in A_{i^*}^+} w_{a'}^t} \geq \frac{1}{g_o}.$$

Combining this with the definition of Δ_z , we obtain:

$$\frac{\partial g^t}{\partial \beta}(\beta, \theta^+, \theta^-) \geq \frac{\Delta_z}{g_o}. \quad (7.18)$$

Thus, $g^t(\beta, \theta^+, \theta^-)$ increases to $+\infty$ as $\beta \rightarrow \infty$, and therefore so does f^t .

To reiterate for emphasis, this claim establishes the unique existence of a entropy parameter estimate $\beta^t > 0$ that satisfies $g^t(\cdot, \theta^{t,+}, \theta^{t,-}) = 0$, or equivalently, (7.16). This claim also establishes that $\beta = \beta^*$ is the unique solution to $g^t(\cdot, \theta^*, \theta^*) = 0$.

2. Claim—We have:

$$|\beta^t - \beta^*| = \frac{\beta^* g_o}{\Delta_z} \cdot \sum_{a \in A(i^*)} (\theta_a^{t,+} - \theta_a^{t,-}) w_a^t.$$

Proof: For convenience, we denote $\theta^\pm := (\theta^+, \theta^-) \in \mathbb{R}^{2|A|}$. For any $\theta^\pm \in \mathbb{R}^{2|A|}$ such that $\theta_a^+ > \theta_a^*$ and $\theta_a^- < \theta_a^*$ for each $a \in A$, let $\beta = \hat{\beta}(\theta^+, \theta^-)$ denote the unique solution to $g^t(\beta, \theta^+, \theta^-)$. Note that for any fixed $w \in \mathcal{W}$ and $p \in \mathbb{R}^{|A|}$, since $z_{a^*}(\theta, w, p)$ is component-wise increasing in θ , we have $f^t(0, \theta^+, \theta^-) \leq \kappa_a^t \leq f^t(\beta^*, \theta^+, \theta^-)$. It thus follows from the Intermediate Value Theorem that $\hat{\beta}(\theta^+, \theta^-) \in [0, \beta^*]$.

By (7.18), we have $\frac{\partial g^t}{\partial \beta}(\beta, \theta^+, \theta^-) \neq 0$ at each $\beta > 0$. This allows us to apply the Implicit Function Theorem, which yields that $\hat{\beta}$ is continuously differentiable in θ^\pm , with:

$$\frac{\partial \hat{\beta}}{\partial \theta^\pm}(\theta^\pm) = \left[\frac{\partial g}{\partial \beta}(\beta, \theta^+, \theta^-) \right]^{-1} \left[\frac{\partial g}{\partial \theta^\pm}(\beta, \theta^+, \theta^-) \right]$$

Now, define $u_+ := \theta^{t,+} - \theta^*$ and $u_- := \theta^{t,-} - \theta^*$. We then have:

$$\begin{aligned}
|\beta^t - \beta^*| &= \left| \int_0^1 \frac{\partial \hat{\beta}}{\partial \theta^\pm} (\theta^+ + \sigma u_+, \theta^- + \sigma u_-)^\top dt \right| \\
&= \left| \int_0^1 \left[\frac{\partial g}{\partial \beta} (\beta, \theta^* - \sigma u_+, \theta^* + \sigma u_-) \right]^{-1} \cdot \frac{\partial g}{\partial \theta^\pm} (\beta, \theta^+ + \sigma u_-, \theta^- + \sigma u_+) \right. \\
&\quad \left. \cdot (\theta^+ - \theta^*, \theta^- - \theta^*) dt \right| \\
&\leq \frac{g_o}{\Delta_z} \cdot \int_0^1 \left| \frac{\partial g}{\partial \theta^\pm} (\beta, \theta^+ + \sigma u_-, \theta^- + \sigma u_+) \cdot (\theta^+ - \theta^*, \theta^- - \theta^*) \right| dt \\
&= \frac{g_o}{\Delta_z} \cdot \int_0^1 \left| \sum_{a \in A} \frac{\partial g}{\partial \theta_a^+} (\beta, \theta^+ + \sigma u_-, \theta^- + \sigma u_+) \cdot (\theta_a^{t,+} - \theta^*) \right. \\
&\quad \left. + \sum_{a \in A} \frac{\partial g}{\partial \theta_a^-} (\beta, \theta^+ + \sigma u_-, \theta^- + \sigma u_+) \cdot (\theta_a^{t,-} - \theta^*) \right| dt,
\end{aligned}$$

where the inequality follows from (7.18). Next, let $A(i^*)$ denote the set of all arcs along routes from the node i^* to the destination node d . Now, observe that, for any $a \in A$, $\beta > 0$ and $\theta^+, \theta^- \in \mathbb{R}^{|A|}$:

$$\begin{aligned}
\frac{\partial g}{\partial \theta_a^+} (\beta, \theta^+, \theta^-) &= -\beta w_a^t \cdot \mathbf{1}\{a \in A(i^*)\}, \\
\frac{\partial g}{\partial \theta_a^-} (\beta, \theta^+, \theta^-) &= \frac{\exp(-\beta \cdot z_a(\theta^{t,-}, w^t, p^t))}{\sum_{a' \in A_{i^*}^+} \exp(-\beta \cdot z_{a'}(\theta^{t,-}, w^t, p^t))} \cdot \beta w_a^t \cdot \mathbf{1}\{a \in A(i^*)\}.
\end{aligned}$$

Substituting into the above upper bound for $|\beta^t - \beta^*|$, we obtain:

$$\begin{aligned}
|\beta^t - \beta^*| &\leq \frac{g_o}{\Delta_z} \cdot \int_0^1 \sum_{a \in A(i^*)} \left| \frac{\partial g}{\partial \theta_a^+} (\beta, \theta^+ + \sigma u_-, \theta^- + \sigma u_+) \right| \cdot (\theta_a^{t,+} - \theta^*) \\
&\quad + \sum_{a \in A(i^*)} \left| \frac{\partial g}{\partial \theta_a^-} (\beta, \theta^+ + \sigma u_-, \theta^- + \sigma u_+) \right| \cdot (\theta^* - \theta_a^{t,-}) dt \\
&\leq \frac{\beta^* g_o}{\Delta_z} \cdot \sum_{a \in A(i^*)} (\theta_a^{t,+} - \theta_a^{t,-}) w_a^t,
\end{aligned}$$

as desired. □

The upper bound (7.17) demonstrates that, by applying the least-squares estimator described in Section 7.2, which ensures that $\|\theta^{t,+} - \theta^{t,-}\|_2 < O(1/\sqrt{t})$ as $t \rightarrow \infty$, we can likewise ensure that $|\beta^t - \beta^*| < O(1/\sqrt{t})$ as $t \rightarrow \infty$.

Algorithm Overview

Armed with the estimation schemes for θ^* and β^* presented in Sections 7.2 and 7.2, we proceed to present our online learning algorithm (Algorithm 14). At each iteration t , the central authority uses latency function and entropy parameter estimates obtained in the previous round to compute the corresponding optimal toll p^t (Line 2). Observe that, for the latency function parameter, we use the lower bound $\theta^{t,-}$ of the confidence interval $(\theta^{t,-}, \theta^{t,+})$, in accordance with the Principle of Optimism in the Face of Uncertainty. Commuters then sequentially select arcs in the traffic network to minimize their average cost-to-go, resulting in the MTE traffic allocation $w^t := \bar{w}^{\theta^*, \beta^*}(p^t)$ (Line 3). The central authority then collects this data, and uses the regularized least-squares method in Section 7.2 to construct an updated estimate θ^t of the underlying latency function parameters θ^* (Lines 5-11). Finally, we construct an update estimate β^t of the underlying entropy parameter β^* using the approach in Section 7.2 (Lines 13-14).

7.3 Lemmas for Regret Analysis

Throughout this section, the notation $x \lesssim y$ denotes that there exists some constant $K(\lambda, \Delta_z, c_\theta, C_\theta, c_\beta, \beta^*)$, such that $x \leq Ky$.

This subsection presents preliminary lemmas that will facilitate the proof of Theorem 7.4.1, which upper bounds the regret of Algorithm 14. We begin with the following lemma, which facilitates the decomposition of the regret into tractable terms.

Lemma 7.3.1. *Suppose $\theta_a^2 \geq \theta_a^1$ for each $a \in A$, and $\beta^2 \geq \beta^1$. Then, for each $w \in \mathcal{W}$:*

$$L(w, \theta^1, \beta^1) \leq L(w, \theta^2, \beta^2).$$

Proof. This follows by noting that $w \geq 0$, and that the entropy term in L is non-positive. \square

Next, we present a result derived from the Fundamental Theorem of Calculus.

Lemma 7.3.2. *If $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is continuously differentiable, then, for each $x_1, x_2 \in \mathbb{R}^n$:*

$$\|f(x_2) - f(x_1)\|_2 \leq \max_{t \in [0,1]} \left\| \frac{\partial f}{\partial x}(x_1 + t(x_2 - x_1)) \right\|_2 \cdot \|x_2 - x_1\|_2.$$

Below, we establish a collection of upper bounds that will be used repeatedly throughout the remainder of the proofs (Lemmas 7.3.3 and 7.3.4).

Algorithm 14: Simultaneous Tolling and Parameter Estimation

Data: $i^* \in I$, $\beta^0 := c_\beta > 0$, $\lambda_a, V_a^0 = \lambda_a, Q_a^0 = 0$, and $p_a^0, \theta_a^{0,-} > 0, \theta_a^{0,+} > 0$ ($\forall a \in A$)

1 **for** $t = 1, \dots, T$ **do**

2 $p^t \leftarrow$ Solution to $p^t = \theta^{t-1,-} \cdot \bar{w}^{\theta^{t-1,-}, \beta^t}(p^t)$.

3 $w^t \leftarrow \bar{w}^{\theta^*, \beta^*}(p^t)$ (Commuters' flow allocation)

4 **for** $a \in A$ **do**

5 $\ell_{a,1}^t, \dots, \ell_{a, \lfloor w_a^t \rfloor}^t \leftarrow$ Costs collected from arc a at iteration t

6 $\gamma_a^t \leftarrow \sqrt{\lambda_a} C_\theta + \sqrt{2 \ln T + \ln \left(\frac{V_a^{t-1}}{\lambda_a} \right)}$

7 $\theta_a^{t,-} \leftarrow \max \left\{ \hat{\theta}_a^{t-1} - \frac{\gamma_a^t}{\sqrt{V_a^{t-1}}}, 0 \right\}$

8 $\theta_a^{t,+} \leftarrow \hat{\theta}_a^{t-1} + \frac{\gamma_a^t}{\sqrt{V_a^{t-1}}}$

9 $V_a^t \leftarrow V_a^{t-1} + \lfloor w_a^t \rfloor (w_a^t)^2$

10 $Q_a^t \leftarrow Q_a^{t-1} + w_a^t \cdot \sum_{k=1}^{\lfloor w_a^t \rfloor} \ell_{a,k}^t$

11 $\hat{\theta}_a^t \leftarrow Q_a^t / V_a^t$

12 **end**

13 $\tilde{\beta}^t \leftarrow$ Solution to $\forall a \in A_{i^*}^+$:

$$\frac{w_a^t}{\sum_{a' \in A_{i^*}^+} w_{a'}^t} = \frac{\exp(-\tilde{\beta}^t \cdot z_a(\theta^{t,-}, w^t, p^t))}{\sum_{a' \in A_{i^*}^+} \exp(-\tilde{\beta}^t \cdot z_{a'}(\theta^{t,+}, w^t, p^t))}.$$

14 $\beta^t \leftarrow \max\{c_\beta, \tilde{\beta}^t\}$.

15 **end**

Lemma 7.3.3. For any $a \in A$ and $t \in [T]$:

$$\gamma_a^t \lesssim \sqrt{\ln(Tg_o)}.$$

Proof. Recall the definition of γ_a^t in (7.6). After taking $\lambda_a = 1$, we have, for any $t \geq 2$:

$$\begin{aligned} \gamma_a^t &= \sqrt{\lambda_a} C_\theta + \sqrt{2 \ln T + 2 \ln \left(\frac{V_a^{t-1}}{\lambda_a} \right)} \\ &= C_\theta + \sqrt{2 \ln T + 2 \ln \left(1 + \sum_{t=1}^{t-1} \lfloor w_a^t \rfloor (w_a^t)^2 \right)} \\ &\leq C_\theta + \sqrt{2 \ln T + 2 \ln (1 + (t-1)g_o^3)} \\ &\lesssim \sqrt{\ln(Tg_o)}. \end{aligned}$$

This result can be straightforwardly extended to the $t = 1$ case by ensuring that the constant encapsulated in the “ \lesssim ” is selected to be large enough. \square

Lemma 7.3.4. *For any $a \in A$:*

$$\sum_{t=1}^T \min \left\{ 1, \frac{\lfloor w_a^t \rfloor (w_a^t)^2}{V_a^{t-1}} \right\} \lesssim \ln(Tg_o).$$

Proof. First, observe that $\min\{1, x\} \leq \frac{1}{\ln 2} \cdot \ln(1+x)$ for each $x \geq 0$. Thus:

$$\begin{aligned} \sum_{t=1}^T \min \left\{ 1, \frac{\lfloor w_a^t \rfloor (w_a^t)^2}{V_a^{t-1}} \right\} &\leq \frac{1}{\ln 2} \cdot \sum_{t=1}^T \ln \left(1 + \frac{\lfloor w_a^t \rfloor (w_a^t)^2}{V_a^{t-1}} \right) \\ &= \frac{1}{\ln 2} \cdot \sum_{t=1}^T \ln \left(\frac{V_a^{t-1} + \lfloor w_a^t \rfloor (w_a^t)^2}{V_a^{t-1}} \right) \\ &= \frac{1}{\ln 2} \cdot \sum_{t=1}^T \ln \left(\frac{V_a^{t-1} + \lfloor w_a^t \rfloor (w_a^t)^2}{V_a^{t-1}} \right) \\ &\leq \frac{1}{\ln 2} \cdot \sum_{t=1}^T \ln \left(\frac{V_a^t}{V_a^{t-1}} \right) \\ &= \frac{1}{\ln 2} \cdot \ln V_a^T \\ &\leq \frac{1}{\ln 2} \cdot \ln(1 + Tg_o^3) \\ &\lesssim \ln(Tg_o), \end{aligned}$$

as desired. \square

Next, we bound the weighted sums of the magnitudes of the latency function parameter errors $\theta^{t,-} - \theta^*$ and entropy parameter $\beta^t - \beta^*$ across iterations $t \in [T]$. First, we require the following lemma.

Lemma 7.3.5. *Under the good event E , for any $p > 0$:*

$$\sum_{t=1}^T \sum_{a \in A} |\theta_a^{t,-} - \theta_a^*| (w_a^t)^p \lesssim g_o^p |A| \sqrt{T} \ln(Tg_o). \quad (7.19)$$

Proof. The desired result follows by taking $p = 2$ in Lemma 7.3.5. \square

Lemma 7.3.6. *Recall that B denotes the number of arcs along routes from i^* to d , which are used to construct an estimate of β^* at each iteration t . Under the good event E :*

$$\sum_{t=1}^T |\beta^t - \beta^*| \lesssim g_o B \sqrt{T} \ln(Tg_o).$$

Proof. Let $A(i^*)$ denote the set of all arcs on routes from i^* to d . By Lemma 7.2.3, under the good event E , we have $\beta^t \in [c_\beta, \beta^*]$, so $|\beta^t - \beta^*| \leq \beta^* - c_\beta$. Moreover, from (7.17), we have:

$$|\beta^t - \beta^*| \lesssim g_o \cdot \sum_{a \in A(i^*)} (\theta_a^{t,+} - \theta_a^{t,-}) w_a^t$$

We then have:

$$\begin{aligned} \sum_{t=1}^T |\beta^t - \beta^*| &\lesssim g_o \cdot \sum_{t=1}^T \left| \min \left\{ \beta^* - c_\beta, \sum_{a \in A(i^*)} (\theta_a^{t,+} - \theta_a^{t,-}) w_a^t \right\} \right| \\ &\lesssim g_o \cdot \sum_{t=1}^T \min \left\{ 1, \sum_{a \in A(i^*)} (\theta_a^{t,+} - \theta_a^{t,-}) w_a^t \right\}. \end{aligned}$$

Take $\tilde{a} \in \max_{a \in A(i^*)} \left\{ \sum_{t=1}^T (\theta_a^{t,+} - \theta_a^{t,-}) w_a^t \right\}$. Then:

$$\begin{aligned} \sum_{t=1}^T |\beta^t - \beta^*| &\lesssim g_o \cdot \sum_{t=1}^T \min \left\{ 1, B \cdot \frac{2\gamma_{\tilde{a}}^t}{\sqrt{V_{\tilde{a}}^{t-1}}} w_{\tilde{a}}^t \right\} \\ &\leq 4g_o B \gamma_{\tilde{a}}^T \cdot \sqrt{T} \cdot \sqrt{\sum_{t=1}^T \min \left\{ 1, \frac{1}{V_{\tilde{a}}^{t-1}} (w_{\tilde{a}}^t)^2 \right\}} \\ &\leq 4g_o B \gamma_{\tilde{a}}^T \cdot \sqrt{T} \cdot \sqrt{\sum_{t=1}^T \min \left\{ 1, \frac{\lfloor w_{\tilde{a}}^t \rfloor (w_{\tilde{a}}^t)^2}{V_{\tilde{a}}^{t-1}} \right\}} \\ &\lesssim g_o B \sqrt{T} \ln(Tg_o) \end{aligned}$$

where we have used the fact that $\lfloor w_{\tilde{a}}^t \rfloor \geq 1$. □

7.4 Regret Analysis

In the main theorem below, we upper bound the regret incurred by Algorithm 14.

Theorem 7.4.1. *There exists some constant $K(\lambda, \Delta_z, c_\theta, C_\theta, c_\beta, \beta^*) > 0$, dependent only on the parameter bounds c_θ, C_θ and c_β , such that for any $T \in \mathbb{N}$:*

$$R \lesssim g_o^2 \ln^2(g_o) |A| \sqrt{T} \ln(Tg_o) \cdot \max \left\{ |I| \ln \left(\frac{|A|}{|I|} \right), B \right\},$$

where $B := |A(i^*)|$ denotes the set of all arcs used to construct the estimates β^t .

Proof. As in Algorithm 14, set $p^t \in \mathbb{R}^{|A|}$ and $p^* \in \mathbb{R}^{|A|}$ to be the unique solutions to the following fixed-point equations:

$$\begin{aligned} p^t &= \theta^{t-1,-} \cdot \bar{w}^{\theta^{t-1,-}, \beta^{t-1}}(p^t), \\ p^* &= \theta^* \cdot \bar{w}^{\theta^*, \beta^*}(p^*). \end{aligned}$$

Under the good event E described in Lemma 7.2.1:

$$L(\bar{w}^{\theta^{t,-}, \beta^t}(p^t), \theta^{t,-}, \beta^t) \leq L(\bar{w}^{\theta^*, \beta^*}(p^*), \theta^{t,-}, \beta^t) \leq L(\bar{w}^{\theta^*, \beta^*}(p^*), \theta^*, \beta^*),$$

where the first inequality follows since Definition 7.1.1, Proposition 7.1.2, and the definition of p^t (Algorithm 14, Line 2) together imply that $\bar{w}^{\theta^{t,-}, \beta^t}(p^t) = \arg \min_{w \in \mathcal{W}} L(w, \theta^{t,-}, \beta^*, \beta^t)$, while the second inequality follows from Lemmas 7.2.1 and 7.3.1.

For convenience, we denote by $\chi : \mathcal{W} \rightarrow \mathbb{R}$ the entropy term in C , restated below:

$$\chi(w) := \sum_{i \in I \setminus \{d\}} \left[\sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right] \quad (7.20)$$

Thus, the regret R can be upper bounded and decomposed as follows:

$$\begin{aligned} R &= \sum_{t=1}^T [L(\bar{w}^{\theta^*, \beta^*}(p^t), \theta^*, \beta^*) - L(\bar{w}^{\theta^*, \beta^*}(p^*), \theta^*, \beta^*)] \\ &\leq \sum_{t=1}^T [L(\bar{w}^{\theta^*, \beta^*}(p^t), \theta^*, \beta^*) - L(\bar{w}^{\theta^{t,-}, \beta^t}(p^t), \theta^{t,-}, \beta^t)] \\ &= \sum_{t=1}^T [L(\bar{w}^{\theta^*, \beta^*}(p^t), \theta^*, \beta^*) - L(\bar{w}^{\theta^*, \beta^*}(p^t), \theta^{t,-}, \beta^t)] \\ &\quad + \sum_{t=1}^T [L(\bar{w}^{\theta^*, \beta^*}(p^t), \theta^{t,-}, \beta^t) - L(\bar{w}^{\theta^{t,-}, \beta^t}(p^t), \theta^{t,-}, \beta^t)] \\ &= \sum_{t=1}^T \sum_{a \in A} (\theta_a^* - \theta_a^{t,-}) \bar{w}_a^{\theta^*, \beta^*}(p^t)^2 \end{aligned} \quad (7.21)$$

$$+ \sum_{t=1}^T \left(\frac{1}{\beta^*} - \frac{1}{\beta^t} \right) \cdot \chi(\bar{w}_a^{\theta^*, \beta^*}(p^t)) \quad (7.22)$$

$$+ \sum_{t=1}^T [L(\bar{w}^{\theta^*, \beta^*}(p^t), \theta^{t,-}, \beta^t) - L(\bar{w}^{\theta^{t,-}, \beta^t}(p^t), \theta^{t,-}, \beta^t)], \quad (7.23)$$

where, in accordance with the notation in Algorithm 14, we set $w^t := \bar{w}^{\theta^*, \beta^*}(p^t)$. Define the three summands (7.21), (7.22), (7.23) by R_1 , R_2 , and R_3 respectively. The convergence rate

of $\theta^{t,-} \rightarrow \theta^*$ and $\beta^t \rightarrow \beta^*$ can then be analyzed to yield non-asymptotic bounds for R_1 and R_2 , respectively. In turn, these bounds are then employed to bound R_3 . Details of these steps are presented in Lemmas 7.4.2, 7.4.4, and 7.4.5 below. \square

Remark 7.4.1. *Compared to [48], our regret upper bound contains an extra term of the form $\max\{|I| \ln(|A|/|I|), B\}$, due to the following unique features of our problem formulation that are not present in [48]: (1) The entropy parameter estimation process, which contributes the network structure-dependent constant B , (2) The tolling authority affects the equilibrium flow allocation indirectly, through tolls, instead of directly dictating commuters' route selections, (3) Mismatch between the latency function and entropy parameter estimates used by the tolling authority to compute tolls $(\theta^{t,-}, \beta^t)$, and the true parameters used by the commuters to best-respond to the implemented toll (θ^*, β^*) .*

Lemma 7.4.2. *Under the good event E :*

$$R_1 := \sum_{t=1}^T \sum_{a \in A} (\theta_a^* - \theta_a^{t,-})(w_a^t)^2 \lesssim g_o^2 |A| \sqrt{T} \ln(Tg_o). \quad (7.24)$$

Proof. Take $\tilde{a} \in \arg \max_{a \in A} \left\{ \sum_{t=1}^T (\theta_a^* - \theta_a^{t,-})(w_a^t)^2 \right\}$. Then, under the good event E :

$$\begin{aligned} R_1 &\leq |A| \cdot \sum_{t=1}^T (\theta_{\tilde{a}}^* - \theta_{\tilde{a}}^{t,-})(w_{\tilde{a}}^t)^2 \\ &\leq |A| \sqrt{g_o} \cdot \sum_{t=1}^T (\theta_{\tilde{a}}^* - \theta_{\tilde{a}}^{t,-})(w_{\tilde{a}}^t)^{3/2} \\ &\leq |A| \sqrt{g_o} \cdot \sum_{t=1}^T \min \left\{ C_\theta g_o^{3/2}, \frac{2\gamma_{\tilde{a}}^t}{\sqrt{V_{\tilde{a}}^{t-1}}} (w_{\tilde{a}}^t)^{3/2} \right\} \\ &\leq 2\sqrt{2} |A| \sqrt{g_o} \cdot \sum_{t=1}^T \min \left\{ C_\theta g_o^{3/2}, \frac{\gamma_{\tilde{a}}^t}{\sqrt{V_{\tilde{a}}^{t-1}}} \cdot \sqrt{[w_{\tilde{a}}^t] \cdot w_{\tilde{a}}^t} \right\}. \end{aligned}$$

where in the final inequality, we have used the fact that, since $w_a^t \geq 1$ by assumption, we have $w_a^t \leq 2[w_a^t]$. Thus, the Cauchy-Schwarz inequality gives:

$$\begin{aligned} R_1 &\leq 2\sqrt{2} C_\theta |A| g_o^2 \gamma_{\tilde{a}}^T \cdot \sum_{t=1}^T \min \left\{ 1, \frac{1}{\sqrt{V_{\tilde{a}}^{t-1}}} \cdot \sqrt{[w_{\tilde{a}}^t] \cdot w_{\tilde{a}}^t} \right\} \\ &\lesssim |A| g_o^2 \sqrt{\ln(Tg_o)} \cdot \sqrt{T} \cdot \sqrt{\sum_{t=1}^T \min \left\{ 1, \frac{[w_{\tilde{a}}^t] (w_{\tilde{a}}^t)^2}{V_{\tilde{a}}^{t-1}} \right\}} \\ &\lesssim g_o^2 |A| \sqrt{T} \ln(Tg_o), \end{aligned}$$

where the final inequality follows from (7.3.4). \square

Next, recall that in (7.20), we defined the entropy term $\chi : \mathcal{W} \rightarrow \mathbb{R}$ as follows:

$$\chi(w) := \sum_{i \in I \setminus \{d\}} \left[\sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right]$$

Lemma 7.4.3. *For any $w \in \mathcal{W}$, we have:*

$$|\chi(w)| \leq g_o \cdot (|I| - 1) \ln \left(\frac{|A|}{|I| - 1} \right)$$

Proof. First, fix $D > 0$ arbitrarily, and consider the following constrained optimization problem on \mathbb{R}^d :

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & \sum_{i=1}^d x_i \ln x_i - \left(\sum_{i=1}^d x_i \right) \ln \left(\sum_{i=1}^d x_i \right) \\ \text{s.t.} \quad & \sum_{i=1}^d x_i = D. \end{aligned}$$

The Lagrangian of the above problem is given by:

$$\mathcal{L}(x, \lambda, \mu) = \sum_{i=1}^d x_i \ln x_i - \left(\sum_{i=1}^d x_i \right) \ln \left(\sum_{i=1}^d x_i \right) + \lambda \left(\sum_{i=1}^d x_i - D \right) + \sum_{i=1}^d \mu_i x_i.$$

The corresponding KKT conditions are therefore:

$$\begin{aligned} 0 = \frac{\partial \mathcal{L}}{\partial x_i} &= \ln x_i + 1 - \ln \left(\sum_{j=1}^d x_j \right) - 1 + \lambda + \mu_i = \ln \left(\frac{x_i}{\sum_{j=1}^d x_j} \right) + \lambda + \mu_i, \quad \forall i \in [d], \\ 0 &= \mu_i x_i, \quad \forall i \in [d], \end{aligned}$$

and $\sum_{i=1}^d x_i = D$. The optimal solution is thus $x^* = \frac{D}{d}(1, \dots, 1)$, with corresponding minimum value:

$$\sum_{i=1}^d x_i^* \ln x_i^* - \left(\sum_{i=1}^d x_i^* \right) \ln \left(\sum_{i=1}^d x_i^* \right) = d \cdot \frac{D}{d} \ln \left(\frac{D}{d} \right) - D \ln D = -D \ln d.$$

This implies that:

$$\left| \sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right| \leq \sum_{a \in A_i^+} w_i \cdot \ln |A_i^+|.$$

Summing over all non-destination nodes, we obtain:

$$\begin{aligned}
& \left| \sum_{i \in I \setminus \{d\}} \left[\sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right] \right| \\
& \leq \sum_{i \in I \setminus \{d\}} \left(\sum_{a \in A_i^+} w_a \right) \ln |A_i^+| \\
& \leq g_o \cdot \sum_{i \in I \setminus \{d\}} \ln |A_i^+| \\
& \leq g_o \cdot |I \setminus \{d\}| \ln \left(\prod_{i \in I \setminus \{d\}} |A_i^+|^{1/|I \setminus \{d\}|} \right) \\
& \leq g_o \cdot |I \setminus \{d\}| \ln \left(\frac{1}{|I \setminus \{d\}|} \sum_{i \in I \setminus \{d\}} |A_i^+| \right) \\
& = g_o \cdot (|I| - 1) \ln \left(\frac{|A|}{|I| - 1} \right),
\end{aligned}$$

where the final inequality follows from the arithmetic-geometric inequality. \square

Lemma 7.4.4. *Under the good event E :*

$$R_2 := \sum_{t=1}^T \left(\frac{1}{\beta^t} - \frac{1}{\beta^*} \right) \cdot \chi(w^t) \lesssim g_o^2 \cdot B(|I| - 1) \ln \left(\frac{|A|}{|I| - 1} \right) \cdot \sqrt{T} \ln(Tg_o). \quad (7.25)$$

Proof. From Lemma 7.3.6:

$$\sum_{t=1}^T |\beta^t - \beta^*| \lesssim g_o B \sqrt{T} \ln(Tg_o).$$

This bound, together with the upper bound on χ provided by Lemma 7.4.3, completes the proof. \square

Next, we bound R_3 , as presented below.

Lemma 7.4.5. *Under the good event E :*

$$\begin{aligned}
R_3 & := \sum_{t=1}^T \left| L(\bar{w}^{\theta^*, \beta^*}(p^t), \theta^{t,-}, \beta^t) - L(\bar{w}^{\theta^{t,-}, \beta^t}(p^t), \theta^{t,-}, \beta^t) \right| \\
& \lesssim g_o^2 \ln^2(g_o) |A| \sqrt{T} \ln(Tg_o) \cdot \max \left\{ |I| \ln \left(\frac{|A|}{|I|} \right), B \right\}.
\end{aligned} \quad (7.26)$$

Proof. Define the map $\tilde{w} : \mathbb{R}^{|A|} \times \mathbb{R} \times \mathbb{R}^{|A|} \rightarrow \mathbb{R}^{|A|}$ by $\tilde{w}(\theta, \beta, p) := \bar{w}^{\theta, \beta}(p)$. Observe that $L(\cdot, \theta, \beta)$ is continuously differentiable on \mathcal{W} , for any fixed $\theta \in \mathbb{R}^{|A|}$, $\beta > 0$; later, we will establish that \tilde{w} is continuously differentiable as well. Then, from the Fundamental Theorem of Calculus to the maps L and \tilde{w} , we obtain:

$$\begin{aligned}
& L(\bar{w}^{\theta^*, \beta^*}(p^t), \theta^{t,-}, \beta^t) - L(\bar{w}^{\theta^{t,-}, \beta^t}(p^t), \theta^{t,-}, \beta^t) \\
&= [L(\bar{w}^{\theta^*, \beta^*}(p^t), \theta^{t,-}, \beta^t) - L(\bar{w}^{\theta^{t,-}, \beta^*}(p^t), \theta^{t,-}, \beta^t)] \\
&\quad + [L(\bar{w}^{\theta^{t,-}, \beta^*}(p^t), \theta^{t,-}, \beta^t) - L(\bar{w}^{\theta^{t,-}, \beta^t}(p^t), \theta^{t,-}, \beta^t)] \\
&= \int_0^1 \frac{\partial L}{\partial w} \left(\bar{w}^{\theta^{t,-} + u(\theta^* - \theta^{t,-}), \beta^t}(p^t), \theta^{t,-}, \beta^t \right) \\
&\quad \cdot \frac{\partial \tilde{w}}{\partial \theta} (\theta^{t,-} + u(\theta^* - \theta^{t,-}), \beta^t, p^t) du \cdot (\theta^* - \theta^{t,-}) \\
&\quad + \int_0^1 \frac{\partial L}{\partial w} \left(\bar{w}^{\theta^{t,-}, \beta^t + u(\beta^* - \beta^t)}(p^t), \theta^{t,-}, \beta^t \right) \\
&\quad \cdot \frac{\partial \tilde{w}}{\partial \theta} (\theta^{t,-}, \beta^t + u(\beta^* - \beta^t), p^t) du \cdot (\beta^* - \beta^t).
\end{aligned} \tag{7.27}$$

For convenience, define:

$$\begin{aligned}
S_{w, \theta} &:= \left\{ \bar{w}^{\theta^{t,-} + u(\theta^* - \theta^{t,-}), \beta^t}(p^t) : u \in [0, 1] \right\}, \\
S_{w, \beta} &:= \left\{ \bar{w}^{\theta^{t,-}, \beta^t + u(\beta^* - \beta^t)}(p^t) : u \in [0, 1] \right\}, \\
S &:= S_{w, \theta} \cup S_{w, \beta}, \\
S_\theta &:= \left\{ \theta^{t,-} + u(\theta^* - \theta^{t,-}) : u \in [0, 1] \right\} \\
S_\beta &:= \left\{ \beta^t + u(\beta^* - \beta^t) : u \in [0, 1] \right\}.
\end{aligned}$$

Then, by applying the Cauchy-Schwarz inequality to (7.3.2), we obtain:

$$\begin{aligned}
& L(\tilde{w}^{\theta^*, \beta^*}(p^t), \theta^{t,-}, \beta^t) - L(\tilde{w}^{\theta^{t,-}, \beta^t}(p^t), \theta^{t,-}, \beta^t) \\
&\leq \max_{w \in S_w} \left\| \frac{\partial L}{\partial w}(w, \theta^{t,-}, \beta^t) \right\|_2 \\
&\quad \cdot \left[\max_{\theta \in S_\theta} \left\| \frac{\partial \tilde{w}}{\partial \theta}(\theta, \beta^*, p^t) \cdot (\theta^* - \theta^{t,-}) \right\|_2 + \max_{\beta \in S_\beta} \left\| \frac{\partial \tilde{w}}{\partial \beta}(\theta^*, \beta, p^t) \right\|_2 \cdot |\beta^* - \beta^t| \right]
\end{aligned} \tag{7.28}$$

We bound each of the max terms in (7.28) below.

1. Bounding $\max_{w \in S_w} \left\| \frac{\partial L}{\partial w}(w, \theta^{t,-}, \beta^t) \right\|_2$:

For each $a \in A$, and any $w \in \mathcal{W}$, $\theta \in \mathbb{R}^{|A|}$, and $\beta > 0$:

$$\frac{\partial L}{\partial w_a}(w, \theta, \beta) = 2\theta_a w_a + \frac{1}{\beta} \ln \left(\frac{w_a}{\sum_{a' \in A_a^+} w_{a'}} \right).$$

Note that $|\theta_a^{t,-}| \leq C_\theta$ for each $a \in A$, and that for any $w \in \mathcal{W}$, we have $\|w\|_2 \leq \sum_{a \in A} w_a \leq m(G)g_o$. Moreover, by Lemma 7.4.3, and the assumption that $w_a \geq 1$ for each $a \in A$ (note that the set $\{w \in \mathbb{R}^{|A|} : w_a \geq 1, \forall a \in A\}$ is convex), we have for each $w \in \mathcal{W}$:

$$\begin{aligned} \sum_{a \in A} \left| \ln \left(\frac{w_a}{\sum_{a' \in A_{i_a}^+} w_{a'}} \right) \right| &= - \sum_{a \in A} \ln \left(\frac{w_a}{\sum_{a' \in A_{i_a}^+} w_{a'}} \right) \\ &\leq - \sum_{a \in A} w_a \ln \left(\frac{w_a}{\sum_{a' \in A_{i_a}^+} w_{a'}} \right) \\ &= |\chi(w)| \\ &\leq g_o \cdot (|I| - 1) \ln \left(\frac{|A|}{|I| - 1} \right) \end{aligned} \quad (7.29)$$

Meanwhile:

$$\sum_{a \in A} \left| \ln \left(\frac{w_a}{\sum_{a' \in A_{i_a}^+} w_{a'}} \right) \right|^2 \leq \ln^2(g_o) |A| \quad (7.30)$$

Thus, we obtain that, for any $w \in S_w$:

$$\begin{aligned} &\left\| \frac{\partial L}{\partial w_a}(w, \theta^{t,-}, \beta^t) \right\|_2 \\ &\leq 2C_\theta m(G)g_o + \frac{1}{c_\beta} \min \left\{ \ln(g_o) \sqrt{|A|}, g_o (|I| - 1) \ln \left(\frac{|A|}{|I| - 1} \right) \right\}. \end{aligned} \quad (7.31)$$

2. Bounding $\max_{\theta \in S_\theta} \left\| \frac{\partial \tilde{w}}{\partial \theta}(\theta, \beta^*, p^t) \cdot (\theta^* - \theta^{t,-}) \right\|_2$:

First, we verify that \tilde{w} is indeed continuously differentiable, and compute the Jacobians $\frac{\partial L}{\partial w}$, $\frac{\partial \tilde{w}}{\partial \theta}$, and $\frac{\partial \tilde{w}}{\partial \beta}$. This requires the results of [20], Lemma 1, which we summarize below. Define $F : \mathcal{W} \times \mathbb{R}^{|A|} \times \mathbb{R}^{|A|} \times \mathbb{R} \times \mathbb{R}^{|A|} \rightarrow \mathbb{R}$ as follows—For each:

$$\begin{aligned} &F(w, \theta, \beta, p) \\ &= \sum_{[a] \in A_o} \int_0^{w_a} [\theta_a z + p_a] dz + \frac{1}{\beta} \sum_{i \neq d} \left[\sum_{a \in A_i^+} w_a \ln w_a - \left(\sum_{a \in A_i^+} w_a \right) \ln \left(\sum_{a \in A_i^+} w_a \right) \right] \end{aligned}$$

Note that $F(\cdot, \theta, \beta, p)$ is strongly convex, with parameter at least c_θ .

Next, observe that \mathcal{W} is a compact subset of a strict affine subspace in $\mathbb{R}^{|A|}$. Let d be the dimension of the smallest affine subspace containing \mathcal{W} . Then, there exist $M \in \mathbb{R}^{|A| \times |I \setminus \{d\}|}$ with orthonormal columns, and $b \in \mathbb{R}^{|I \setminus \{d\}|}$ such that:

$$\mathcal{W} = \{w \in \mathbb{R}^{|A|} : M^\top w + b = 0, w_a \geq 0, \forall a \in A\}.$$

Let $B \in \mathbb{R}^{|A| \times (|A| - |I \setminus \{d\}|)}$ consist of orthonormal columns orthogonal to the columns of M . We then use the theory of constrained optimization to completely characterize $\tilde{w}(\theta, \beta, p) = \bar{w}^{\theta, \beta}(p)$. In particular, $w = \tilde{w}(\theta, \beta, p)$ if and only if the following implicit equation, characterized by the map $J : \mathbb{R}^{|A|} \times \mathbb{R}^{|A|} \rightarrow \mathbb{R}^{|A|}$ defined below, is satisfied:

$$J(w, \theta, \beta, p) := \begin{bmatrix} M^\top w + b \\ B^\top \nabla_w F(w, \theta, \beta, p) \end{bmatrix} = 0.$$

Moreover, the proof of [20], Lemma 1 establishes that, for any fixed $\theta \in \mathbb{R}^{|A|}$, $\beta > 0$, $p \in \mathbb{R}^{|A|}$:

$$\frac{\partial J}{\partial w}(\theta, \beta, p) = \begin{bmatrix} M^\top \\ B^\top \nabla_w^2 F(w, \theta, \beta, p) \end{bmatrix} \in \mathbb{R}^{|A| \times |A|}$$

is non-singular. By the Implicit Function Theorem, this establishes the continuous differentiability of \tilde{w} . We can then compute $\frac{\partial \tilde{w}}{\partial \theta} \in \mathbb{R}^{|A| \times |A|}$ at any $(\theta, \beta, p) \in \mathbb{R}^{|A|} \times \mathbb{R} \times \mathbb{R}^{|A|}$ as:

$$\begin{aligned} \frac{\partial \tilde{w}}{\partial \theta}(\theta, \beta, p) &= \left[\frac{\partial J}{\partial w}(\theta, \beta, p) \right]^{-1} \frac{\partial J}{\partial \theta}(\theta, \beta, p) \\ &= \begin{bmatrix} M^\top \\ B^\top \nabla_w^2 F(w, \theta, \beta, p) \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ B^\top \frac{\partial}{\partial \theta} \nabla_w F(w, \theta, \beta, p) \end{bmatrix} \\ &= B(B^\top \nabla_w^2 F(w, \theta, \beta, p)B)^{-1} B^\top \cdot \frac{\partial}{\partial \theta} \nabla_w F(w, \theta, \beta, p), \end{aligned}$$

where we have used the fact that by construction, $\begin{bmatrix} M & B \end{bmatrix}$ is an orthogonal matrix (see [20], Appendix A).

Now, observe that the (a, a') -entry of $\frac{\partial}{\partial \beta} \nabla_w F(w, \theta, \beta, p) \in \mathbb{R}^{|A| \times |A|}$ is given by:

$$\frac{\partial^2}{\partial \theta_{a'} \partial w_a} F(w, \theta, \beta, p) = 2w_a \cdot \mathbf{1}\{a' = a\}, \quad \forall a \in A,$$

Substituting back into (7.28) and applying the Cauchy-Schwarz inequality, we obtain that, for each $\theta \in S_\theta$:

$$\frac{\partial \tilde{w}}{\partial \theta}(\theta, \beta^*, p^t) \cdot (\theta^* - \theta^{t,-}) = B(B^\top \nabla_w^2 F(w, \theta, \beta, p)B)^{-1} B^\top \cdot ((\theta^* - \theta^{t,-})w_a^t)_{a \in A}.$$

Applying the Cauchy-Schwarz inequality, we obtain:

$$\begin{aligned} &\max_{\theta \in S_\theta} \left\| \frac{\partial \tilde{w}}{\partial \theta}(\theta, \beta^*, p^t) \cdot (\theta^* - \theta^{t,-}) \right\|_2 \\ &\leq \|B(B^\top \nabla_w^2 F(w, \theta, \beta, p)B)^{-1} B^\top\|_2 \cdot \|((\theta^* - \theta^{t,-})w_a^t)_{a \in A}\|_2. \end{aligned}$$

Since the columns of B are orthonormal, we have $\|B(B^\top \nabla_w^2 F(w, \theta, \beta, p)B)^{-1}B^\top\|_2 \leq \|\nabla_w^2 F(w, \theta, \beta, p)\|_2 \leq 1/c_\theta$. Moreover, we can upper bound $\|((\theta^* - \theta^{t,-})w_a^t)_{a \in A}\|_2 \leq \|((\theta^* - \theta^{t,-})w_a^t)_{a \in A}\|_1 = \sum_{a \in A} (\theta^* - \theta^{t,-})w_a^t$. We thus obtain:

$$\max_{\theta \in S_\theta} \left\| \frac{\partial \tilde{w}}{\partial \theta}(\theta, \beta^*, p^t) \cdot (\theta^* - \theta^{t,-}) \right\|_2 \leq \frac{1}{c_\theta} \cdot \sum_{a \in A} (\theta^* - \theta^{t,-})w_a^t. \quad (7.32)$$

3. Bounding $\max_{\beta \in S_\beta} \left\| \frac{\partial \tilde{w}}{\partial \beta}(\theta^*, \beta, p^t) \right\|_2 \cdot |\beta^* - \beta^t|$:

In the same manner that we used to compute $\frac{\partial \tilde{w}}{\partial \theta}$ above, we can compute $\frac{\partial \tilde{w}}{\partial \beta} \in \mathbb{R}^{|A|}$ at any $(\theta, \beta, p) \in \mathbb{R}^{|A|} \times \mathbb{R} \times \mathbb{R}^{|A|}$ as:

$$\begin{aligned} \frac{\partial \tilde{w}}{\partial \beta}(\theta, \beta, p) &= \left[\frac{\partial J}{\partial w}(\theta, \beta, p) \right]^{-1} \frac{\partial J}{\partial \beta}(\theta, \beta, p), \\ &= \begin{bmatrix} M^\top \\ B^\top \nabla_w^2 F(w, \theta, \beta, p) \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ B^\top \frac{\partial}{\partial \beta} \nabla_w F(w, \theta, \beta, p) \end{bmatrix} \\ &= B(B^\top \nabla_w^2 F(w, \theta, \beta, p)B)^{-1}B^\top \cdot \frac{\partial}{\partial \beta} \nabla_w F(w, \theta, \beta, p), \end{aligned}$$

Now, observe that the a -th entry of $\frac{\partial}{\partial \beta} \nabla_w F(w, \theta, \beta, p) \in \mathbb{R}^{|A|}$ is:

$$\frac{\partial^2}{\partial \beta \partial w_a} F(w, \theta, \beta, p) = -\frac{1}{\beta^2} \ln \left(\frac{w_a}{\sum_{a' \in A_{i_a}^+} w_{a'}} \right).$$

Using (7.29) and (7.30), we obtain:

$$\left\| \frac{\partial}{\partial \beta} \nabla_w F(w, \theta, \beta, p) \right\|_2 \leq \frac{1}{\beta^2} \cdot \min \left\{ \ln(g_o) \sqrt{|A|}, g_o(|I| - 1) \ln \left(\frac{|A|}{|I| - 1} \right) \right\}$$

Finally, we conclude that:

$$\begin{aligned} &\max_{\beta \in S_\beta} \left\| \frac{\partial \tilde{w}}{\partial \beta}(\theta^*, \beta, p^t) \right\|_2 \cdot |\beta^* - \beta^t| \\ &\leq \|B(B^\top \nabla_w^2 F(w, \theta, \beta, p)B)^{-1}B^\top\|_2 \cdot \left\| \frac{\partial}{\partial \beta} \nabla_w F(w, \theta, \beta, p) \right\|_2 \cdot |\beta^* - \beta^t| \\ &\leq \frac{1}{c_\theta \beta^2} \cdot \min \left\{ \ln(g_o) \sqrt{|A|}, g_o(|I| - 1) \ln \left(\frac{|A|}{|I| - 1} \right) \right\} \cdot |\beta^* - \beta^t|. \end{aligned} \quad (7.33)$$

Substituting (7.31), (7.32), (7.33) back into (7.28), we obtain that:

$$R_3 = \sum_{t=1}^T |L(\bar{w}^{\theta^*, \beta^*}(p^t), \theta^{t,-}, \beta^t) - L(\bar{w}^{\theta^{t,-}, \beta^t}(p^t), \theta^{t,-}, \beta^t)|$$

$$\begin{aligned}
&\lesssim \left(m(G)g_o + \frac{1}{c_\beta} \cdot g_o \cdot (|I| - 1) \ln \left(\frac{|A|}{|I| - 1} \right) \right) \\
&\quad \cdot \left[\sum_{t=1}^T \sum_{a \in A} (\theta^* - \theta^{t,-}) w_a^t + \min \left\{ \ln^2(g_o) \cdot |A|, g_o |I| \ln \left(\frac{|A|}{|I|} \right) \right\} \cdot \sum_{t=1}^T |\beta^* - \beta^t| \right]
\end{aligned} \tag{7.34}$$

Applying Lemmas 7.3.5 (with $p = 1$) and 7.3.6, we obtain:

$$\begin{aligned}
R_3 &\lesssim \left(m(G)g_o + \min \left\{ \ln(g_o) \sqrt{|A|}, g_o |I| \ln \left(\frac{|A|}{|I|} \right) \right\} \right) \\
&\quad \cdot \left[g_o |A| \sqrt{T} \ln(Tg_o) + \min \left\{ \ln(g_o) \sqrt{|A|}, g_o |I| \ln \left(\frac{|A|}{|I|} \right) \right\} \cdot g_o B \sqrt{T} \ln(Tg_o) \right] \\
&\lesssim g_o^2 m(G) |A| \sqrt{T} \ln(Tg_o) + g_o^2 \ln(g_o) m(G) \sqrt{|A|} B \sqrt{T} \ln(Tg_o) \\
&\quad + g_o^2 |A| |I| \ln \left(\frac{|A|}{|I|} \right) \sqrt{T} \ln(Tg_o) + g_o \ln^2(g_o) |A| B \sqrt{T} \ln(Tg_o) \\
&\lesssim g_o^2 \ln^2(g_o) |A| \sqrt{T} \ln(Tg_o) \cdot \max \left\{ |I| \ln \left(\frac{|A|}{|I|} \right), B \right\}.
\end{aligned}$$

Note that we have used the fact that $m(G) \leq |I|$. □

Finally, we combine the results of Lemmas 7.4.2, 7.4.4, and 7.4.5 in the above sections to conclude our proof of Theorem 7.4.1.

Proof of Theorem 7.4.1, Continued. Lemmas 7.4.2, 7.4.4, and 7.4.5, we have:

$$\begin{aligned}
R_1 &\lesssim g_o^2 |A| \sqrt{T} \ln(Tg_o), \\
R_2 &\lesssim g_o^2 \cdot B |I| \ln \left(\frac{|A|}{|I|} \right) \cdot \sqrt{T} \ln(Tg_o), \\
R_3 &\lesssim g_o^2 \ln^2(g_o) |A| \sqrt{T} \ln(Tg_o) \cdot \max \left\{ |I| \ln \left(\frac{|A|}{|I|} \right), B \right\}.
\end{aligned}$$

Note that $R_1 \lesssim R_3$ and $R_2 \lesssim R_3$. We thus conclude that:

$$\begin{aligned}
R &= R_1 + R_2 + R_3 \\
&\lesssim g_o^2 \ln^2(g_o) |A| \sqrt{T} \ln(Tg_o) \cdot \max \left\{ |I| \ln \left(\frac{|A|}{|I|} \right), B \right\}.
\end{aligned}$$

□

7.5 Experiments

Here, we present numerical results on simulated traffic networks that validate the regret bounds presented in Theorem 7.4.1. In particular, we run Algorithm 14 for $T = 2500$ iterations, with $g_o = 100$, on the parallel-arc network in Figure 7.1 (left), with underlying parameters $\theta^* := (1.5, 2.5, 3.5, 4.5, 5.5, 6.5) \in \mathbb{R}^6$, and $\beta^* = 0.25$, and on the more general network in Figure 7.1 (right), with underlying parameters $\theta^* := (0.6, 0.4, 0.4, 0.4, 0.6, 0.6) \in \mathbb{R}^6$, and $\beta^* = 0.25$. To suppress constants in the cumulative regret, we selected $\lambda_a = 0.01$ for each $a \in [6]$.

For convenience, for each iteration $t \in [T]$, let $L^t := L(w^{\theta^*, \beta^*}(p^t), \theta^*, \beta^*)$ denote the cost incurred at iteration t , let $L^* := L(w^{\theta^*, \beta^*}(p^*), \theta^*, \beta^*)$ denote the minimum possible cost, and let $R^t := \sum_{\tau=1}^t [L(w^{\theta^*, \beta^*}(p^\tau), \theta^*, \beta^*) - L(w^{\theta^*, \beta^*}(p^*), \theta^*, \beta^*)]$ denote the cumulative regret up to iteration t . In Figure 7.2, we detail the growth of the cumulative regret $R^t - L^*t$ as a function of t . We also provide logarithmic plots that describe the decay of the stage-wise regret $L^t - L^*$, the magnitude of the latency function parameter estimation error $\|\theta^t\|_2$, and the magnitude of the entropy parameter estimation error $|\beta^t - \beta|$. For both networks, the cumulative regret increases as a sub-linear function of the iteration count, while the cumulative regret, θ estimation error, and β estimation error decrease gracefully to 0.

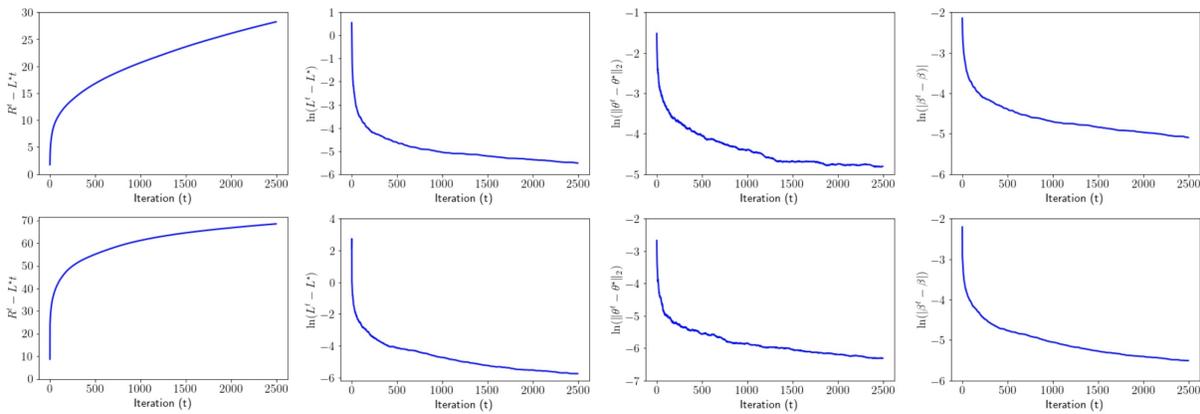


Figure 7.2: (Left to right) The cumulative regret $R^t - L^*t$, logarithm of stage-wise regret $\ln(L^t - L^*)$, logarithm of θ -estimation error $\ln(\|\theta^t - \theta^*\|_2)$, and logarithm of stage-wise regret $\ln(|\beta^t - \beta|)$ for the parallel-arc network in Figure 7.1 (top) and the more general network in Figure 7.1 (bottom). Note the sub-linear growth of the cumulative regret with respect to the iteration count, and the rapid decay of the stage-wise regret, θ -estimation error, and β -estimation error to 0, as the iteration count t increases.

7.6 Discussion

This work presents a novel online learning algorithm to learn the latency function and entropy parameters that characterize commuters' arc-selection decisions on a single source-single destination traffic network, while simultaneously implementing tolls to minimize the overall network congestion. We characterize a notion of regret using the accumulation across iterations of the gap between the incurred and minimum costs, and prove that our cumulative regret metric increases sub-linearly in the number of iterations t . Finally, we present numerical results illustrating the performance of our regret algorithm on simulated traffic networks.

Part IV
Future Work

Chapter 8

Conclusion

The previous chapters of this thesis presented contributions towards safe and efficient societal-scale autonomous navigation along three crucial dimensions—A fast and reliable optimization template for localization and mapping (Part 1), dynamic game theory-based algorithms for joint prediction and motion planning (Part 2), and adaptive tolling schemes to reduce congestion on large transportation networks (Part 3). Below, we discuss promising directions of future work aimed at bringing the promise of autonomous navigation closer to reality.

8.1 Part 1

Interesting avenues of ongoing work along the direction of localization and mapping include establishing a rapprochement of learning-based and model-based state estimation methods for downstream path planning. Another interesting research direction involves deploying SLAM algorithms in challenging, high-speed settings, such as autonomous racing, to stress-test their computational efficiency and robustness in extreme instances of autonomous navigation, while ensuring safe operation at all times.

- **Rapprochement of learning-based perception and model-based estimation.** An important open problem in my ongoing research is to enable robust state estimation across a wide range of different environments, which necessitates a principled fusion of learning- and model-based perception methods in a unified framework. Learning-based perception can provide semantic interpretations of the environment, but often struggles to perform accurate metric reconstruction when necessary, such as for the generation of safe trajectories for an agent in cramped or cluttered surroundings. Meanwhile, visual SLAM algorithms can generate consistent metric maps, but are often slow, and do not provide semantic information relevant to the downstream navigation task. I plan to investigate how these two approaches, learning-based and model-based, should be combined, to support an autonomous navigation framework that is computationally efficient, enjoys a high goal-reaching success rate, and is robust with respect to challenging and unpredictable scenes.



Figure 8.1: Indy Autonomous racecars in motion at the Las Vegas Motor Speedway racetrack.

- **Autonomous Racing.** In the Indy Autonomous Challenge (IAC), self-driving race cars must complete laps around a professional racetrack at high speed (> 150 mph), without incurring collisions or violating lane-following regulations (Figure 8.1). However, state estimation during racing poses an intimidating challenge, since motion blur at such high speeds precludes the use of cameras for tracking lateral (perpendicular to the race tracks) displacement. Moreover, the irregularly shaped tracks often lack easily discernible features. To meet these challenges, my collaborators and I have proposed a SLAM architecture that uses LiDAR sensor data for lateral tracking, and stereo cameras for longitudinal tracking. We are excited to perform extensive experiments across a wide range of autonomous racetracks, such as the Las Vegas Motor Speedway and the Monza Racetrack in Italy.

8.2 Part 2

In the realm of dynamic game-theory-based motion planning algorithm design, many promising directions of future work await investigation. The first involves extending the defensive driving and GTP-SLAM frameworks (Chapters 4 and 5) to capture more nuanced multi-agent interactions in real-world traffic scenarios. The second direction involves applying iterative dynamic game algorithms to analyze control theory and mechanism design problems in societal-scale, non-atomic settings, such as adaptive tolling (Chapters 6 and 7).



Figure 8.2: The Monza Motor Speedway racetrack is irregularly shaped, and lacks features commonly used for localization in real-life traffic, such as traffic signals.

- **Defensive Driving.** I aim to explore more flexible approaches for encoding defensive behavior in games. For example, in many real-life scenarios, vehicle dynamics should be modeled with stochastic, partial observations, and should account for occlusions and dynamic disturbances. Also, the ego agent may wish to select the adversarial time horizon more flexibly. Of particular interest are cases where the ego agent may choose to vary T_{adv} from one non-ego agent to another. For example, the ego may observe that some non-ego agents are behaving more adversarially than others, and respond accordingly by associating such players with higher values of T_{adv} , compared to other non-ego agents. In addition, the ego may wish to allocate different parts of the overall time horizon to be adversarial, rather than simply the first T_{adv} seconds. For example, choosing the adversarial time horizon to be the final T_{adv} seconds of the overall time horizon, rather than the first T_{adv} seconds, would transform the game from an adversarial-to-cooperative type to a cooperative-to-adversarial type. This would be useful in situations where the ego agent predicts that the surrounding non-ego agents are currently cooperative, but may become momentarily distracted in the near future. For example, such a formulation may motivate the ego agent to gradually approach an intersection at which other agents might run a red light.
- **SLAM with Game-Theoretic Priors.** Regarding my work on SLAM algorithms that use game-theoretic priors, our current experiments do not yet consider loop closures, which are essential for the long-term recovery of static scenes in SLAM tasks. It is thus critical to study how to best incorporate game-theoretic priors when detect-

ing and enforcing loop closures. Moreover, our current method was implemented in full-graph optimization problems; in practice, however, SLAM graphs are often optimized incrementally, as measurements are acquired in real-time. Our approach readily extends to this setting. Finally, our method only computes open-loop game strategies, corresponding to feedforward, rather than feedback, controls. Future work will investigate game-theoretic SLAM priors in more complicated strategy spaces.

- **Dynamic feedback large-scale games.** Another promising direction for future research is to design dynamic game theory-based algorithms to model non-cooperative interactions between hundreds or thousands of self-interested dynamic agents. In the literature, non-atomic games are generally solved under a static setting or an open-loop information pattern, which generally struggle to capture more nuanced interactions between dynamic agents in atomic games. I aim to address this gap by developing algorithms for solving dynamic feedback non-atomic games, to characterize the aggregate behavior of non-cooperative, self-interested autonomous agents at large scales.

8.3 Part 3

Future work for adaptive toll design will target the removal of certain assumptions placed on the formulation presented in Chapters 6 and 7. These assumptions may be overly stringent in practice, or fail to account for aspects of social welfare, such as fairness.

1. **Studying Transient Effects in Traffic.** Chapters 6 and 7 focus on a static and asymptotic model of social optimality. In other words, any tolling mechanism or route assignment scheme that asymptotically steers the aggregate traffic allocation towards latency-minimizing flow levels is declared to be socially optimal. This paradigm does not capture certain complex, non-asymptotic phenomena that often cause significant disruptions in traffic flow in real life. A prominent example is the occurrence of traffic accidents that block several lanes on a busy highway, causing vehicular pileup, excessive congestion, and possibly even additional accidents beyond the first. It is thus of interest to develop arc assignment methods that can swiftly disperse vehicles towards alternate routes when such an emergency event occurs at a crowded traffic scene.
2. **Heterogeneous Tolling for Fairness and Robustness.** The tolling mechanism proposed in Chapter 6 assumes that in a transportation network with fixed tolls, all travelers on the same arc perceive the same overall cost, obtained by summing the toll and a monetary value associated with the travel time. However, in real-life traffic, travelers may select routes in different, nuanced ways that are not well-modeled by this simplified framework. For example, some commuters may take other factors of the traffic network into account, such as paving quality or neighborhood safety, when performing route selection. Moreover, individuals with lower income may respond more drastically to the deployment of a toll, while individuals with higher income may

completely ignore tolls imposed on the network. This example additionally raises issues of fairness, since it is undesirable to considerably increase the commute costs faced by low-income communities. Future work will aim to address the above concerns, to implement an adaptive incentive design scheme on a transportation network that more accurately captures nuances in the route selection decisions of different populations of travelers, while ensuring that travelers from all communities are treated fairly and humanely.

- 3. Online Learning for Adaptive Tolling.** A natural avenue of future work, for the online learning and adaptive tolling project (Chapter 7), is to extend the online learning algorithm of Chapter 7 to traffic networks with multiple origin-destination pairs, and possibly bi-directional edges. Such settings pose particular challenges to the estimation of the entropy parameters, since each arc in the network could be shared among commuters representing a diverse range of travel histories and destinations. It would also be interesting to explore whether one could relax the assumption that the central authority possesses knowledge of a lower bound $c_\beta > 0$ for β^* . Yet another direction of future inquiry would be to study the effect of a non-constant input flow g_o on the growth of the cumulative regret. Finally, it would be interesting to run Algorithm 14 on larger simulated or real-life traffic networks, to fully investigate the empirical dependence of the cumulative regret growth on the network parameters.

Bibliography

- [1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. “Improved Algorithms for Linear Stochastic Bandits”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor et al. Vol. 24. Curran Associates, Inc., 2011.
- [2] Takashi Akamatsu. “Decomposition of Path Choice Entropy in General Transport Networks”. In: *Transportation Science* 31.4 (Nov. 1997), pp. 349–362. DOI: 10.1287/trsc.31.4.349.
- [3] Jean-Bernard Baillon and Roberto Cominetti. “Markovian Traffic Equilibrium”. In: *Mathematical Programming* (Feb. 2008). DOI: 10.1007/s10107-006-0076-2.
- [4] Somil Bansal et al. “Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances”. In: (2017), pp. 2242–2253.
- [5] Tamer Basar and Geert Jan Olsder. *Dynamic Noncooperative Game Theory*. Vol. 23. SIAM, 1999.
- [6] B. Bell. “The Iterated Kalman Smoother as a Gauss-Newton Method”. In: *SIAM J. Optim.* 4 (1994), pp. 626–636.
- [7] Richard Bellman. “Dynamic programming”. In: *Science* 153.3731 (1966), pp. 34–37.
- [8] M. E. Ben-Akiva. *Discrete Choice Analysis: Theory and Application to Travel Demand*. Cambridge: MIT Press, 1985.
- [9] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control (Two Volume Set)*. 2nd. Athena Scientific, 2001. ISBN: 1886529086.
- [10] Vivek Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.
- [11] Vivek S Borkar. “Cooperative Dynamics and Wardrop Equilibria”. In: *Systems & control letters* 58.2 (2009), pp. 91–93.
- [12] Neal E. Boudette and Bill Vlasic. “Tesla Self-Driving System Faulted by Safety Agency in Crash”. In: *The New York Times* (Sept. 12, 2017). URL: <https://www.nytimes.com/2017/09/12/business/self-driving-cars.html>.

- [13] Michael Burri et al. “The EuRoC Micro Aerial Vehicle Datasets”. In: *The International Journal of Robotics Research* (2016). DOI: 10.1177/0278364915620033. eprint: <http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.full.pdf+html>. URL: <http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract>.
- [14] C. Cadena et al. “Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age”. In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1309–1332.
- [15] Haipeng Chen et al. “DyETC: Dynamic electronic toll collection for traffic congestion alleviation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.
- [16] Chih-Yuan Chiu. *Simultaneous Localization and Mapping: A Rapprochement of Filtering and Optimization-Based Approaches*. 2021. URL: <https://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-76.pdf> (visited on 05/14/2021).
- [17] Chih-Yuan Chiu. “SLAM Backends with Objects in Motion: A Unifying Framework and Tutorial”. In: *2023 American Control Conference (ACC)*. 2023, pp. 1635–1642. DOI: 10.23919/ACC55779.2023.10155957.
- [18] Chih-Yuan Chiu and David Fridovich-Keil. “GTP-SLAM: Game-Theoretic Priors for Simultaneous Localization and Mapping in Multi-Agent Scenarios”. In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. 2022, pp. 247–252. DOI: 10.1109/CDC51059.2022.9992656.
- [19] Chih-Yuan Chiu and Shankar Sastry. “Parameter Estimation in Optimal Tolling for Traffic Networks Under the Markovian Traffic Equilibrium”. In: (Submitted to): *American Control Conference* (2024).
- [20] Chih-Yuan Chiu et al. “Dynamic Tolling in Arc-based Traffic Assignment Models”. In: (Accepted at): *59th Annual Allerton Conference on Communication, Control, and Computing* (2023).
- [21] Chih-Yuan Chiu*, David Fridovich-Keil*, and Claire Tomlin. “Encoding Defensive Driving as a Dynamic Nash Game”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 10749–10756. DOI: 10.1109/ICRA48506.2021.9560788.
- [22] Chih-Yuan Chiu* et al. “Arc-based Traffic Assignment: Equilibrium Characterization and Learning”. In: (Accepted at): *62nd IEEE Conference on Decision and Control (CDC)* (2023).
- [23] Simon Le Cleac’h, Mac Schwager, and Zachary Manchester. “ALGAMES: A Fast Solver for Constrained Dynamic Games”. In: *arXiv preprint arXiv:1910.09713* (2019).
- [24] Anna Dai et al. “Fast Frontier-based Information-driven Autonomous Exploration with an MAV”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 9570–9576. DOI: 10.1109/ICRA40945.2020.9196707.

- [25] Andrew J. Davison. *FutureMapping: The Computational Structure of Spatial AI Systems*. 2018. eprint: 1803.11288 (cs.AI).
- [26] Andrew J. Davison and Joseph Ortiz. *FutureMapping 2: Gaussian Belief Propagation for Spatial AI*. 2019. eprint: 1910.14139 (cs.AI).
- [27] F. Dellaert. “Factor Graphs for Robot Perception”. In: *Foundations and Trends® in Robotics* 6.1-2 (2017), pp. 1–139.
- [28] Frank Dellaert et al. “Gtsam”. In: *URL: <https://borg.cc.gatech.edu>* (2012).
- [29] Frank Dellaert, Michael Kaess, et al. “Factor Graphs for Robot Perception”. In: *Foundations and Trends® in Robotics* 6.1-2 (2017), pp. 1–139.
- [30] Frank Dellaert and Michael Kaess. “Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing”. In: *The International Journal of Robotics Research* 25.12 (2006), pp. 1181–1203. DOI: 10.1177/0278364906072768. eprint: <https://doi.org/10.1177/0278364906072768>. URL: <https://doi.org/10.1177/0278364906072768>.
- [31] Bolei Di and Andrew Lamperski. “Local First-Order Algorithms for Constrained Non-linear Dynamic Games”. In: *2020 American Control Conference (ACC)*. IEEE. 2020, pp. 5358–5363.
- [32] Bolei Di and Andrew Lamperski. “Newton’s Method and Differential Dynamic Programming for Unconstrained Nonlinear Dynamic Games”. In: (2019), pp. 4073–4078.
- [33] Noel E Du Toit and Joel W Burdick. “Robot Motion Planning in Dynamic, Uncertain Environments”. In: *IEEE Transactions on Robotics* 28.1 (2012), pp. 101–115.
- [34] Kevin Eickenhoff, Patrick Geneva, and Guoquan Huang. “Closed-Form Preintegration Methods for Graph-based Visual-Inertial Navigation”. In: *The International Journal of Robotics Research* 38 (2019), pp. 563–586.
- [35] L. C. Evans and P. E. Souganidis. “Differential Games and Representation Formulas for Solutions of Hamilton-Jacobi-Isaacs Equations”. In: *Indiana University mathematics journal* 33.5 (1984), pp. 773–797.
- [36] Farbod Farshidian and J. Buchli. “Chance-Constrained Optimal Covariance Steering with Iterative Risk Allocation”. In: *ArXiv abs/1512.07173* (2015).
- [37] Farbod Farshidian and J. Buchli. “Risk Sensitive, Nonlinear Optimal Control: Iterative Linear Exponential-Quadratic Optimal Control with Gaussian Noise”. In: *ArXiv abs/1512.07173* (2015).
- [38] Jaime Fisac et al. “Reach-Avoid Problems with Time-Varying Dynamics, Targets and Constraints”. In: *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*. 2015, pp. 11–20.
- [39] Jaime F Fisac and S Shankar Sastry. “The Pursuit-evasion-defense Differential Game in Dynamic Constrained Environments”. In: *54th Conference on Decision and Control (CDC)*. IEEE. 2015, pp. 4549–4556.

- [40] Jaime F Fisac et al. “Hierarchical Game-Theoretic Planning for Autonomous Vehicles”. In: *arXiv preprint arXiv:1810.05766* (2018).
- [41] Michael Florian and Donald Hearn. *Network Equilibrium and Pricing*. Vol. 23. Jan. 2003, pp. 373–411. DOI: 10.1007/0-306-48058-1_11.
- [42] Alejandra Fonseca-Morales and Onésimo Hernández-Lerma. “Potential Differential Games”. In: *Dynamic Games and Applications* 8.2 (2018), pp. 254–279.
- [43] Christian Forster et al. “IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation”. In: Georgia Institute of Technology. 2015.
- [44] F. Fraundorfer and D. Scaramuzza. “Visual Odometry, Part II: Matching, Robustness, Optimization, and Applications”. In: *IEEE RAM* 19.2 (2012), pp. 78–90.
- [45] U. Frese, R. Wagner, and T. Röfer. “A SLAM Overview from a User’s Perspective”. In: *Künstliche Intelligenz* 24 (2010), pp. 191–198.
- [46] David Fridovich-Keil et al. “Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games”. In: *arXiv preprint arXiv:1909.04694* (2019).
- [47] Drew Fudenberg and David K Levine. *The Theory of Learning in Games*. Vol. 2. MIT press, 1998.
- [48] Sreenivas Gollapudi et al. “Online Learning for Traffic Navigation in Congested Networks”. In: *International Conference on Algorithmic Learning Theory* (2023).
- [49] David González-Sánchez and Onésimo Hernández-Lerma. “Dynamic Potential Games: The Discrete-Time Stochastic Case”. In: *Dynamic Games and Applications* 4 (2014), pp. 309–328.
- [50] G. Grisetti et al. “A Tutorial on Graph-Based SLAM”. In: *IEEE Intelligent Transportation Systems Magazine* 2.4 (2010), pp. 31–43. DOI: 10.1109/MITS.2010.939925.
- [51] C. Hua and L. Dou. “A New Algorithm Merging Static Game with Complete Information Into EKF For Multi-Robot Cooperative Localization”. In: *Zhongnan Daxue Xuebao (Ziran Kexue Ban)/Journal of Central South University (Science and Technology)* 44 (Nov. 2013), pp. 4534–4541.
- [52] S. Huang and G. Dissanayake. “A Critique of Current Developments in Simultaneous Localization and Mapping”. In: *IJARS* 13.5 (2016), p. 1729881416669482. DOI: 10.1177/17298814166669482.
- [53] R Isaacs. “Differential games, parts 1-4”. In: *The Rand Corporation, Research Memorandums Nos. RM-1391, RM-1411, RM-1486* 55 (1954).
- [54] Rufus Isaacs. *Differential Games: a Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. Courier Corporation, 1999.
- [55] Rufus Isaacs. *Games of Pursuit*. Tech. rep. Rand Corporation, 1951.

- [56] M. Kaess et al. “iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree”. In: *IJRR* 31 (2 Feb. 2012), pp. 217–236.
- [57] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. “iSAM: Incremental Smoothing and Mapping”. In: *Robotics, IEEE Transactions on* 24 (Jan. 2009), pp. 1365–1378. DOI: 10.1109/TR0.2008.2006706.
- [58] Noriko Kanekoa, Daisuke Fukudab, and Qian Gec. “Optimal Congestion Tolling Problem under the Markovian Traffic Equilibrium”. In: *Sustainability* (2021).
- [59] Talha Kavuncu, Ayberk Yaraneri, and Negar Mehr. “Potential iLQR: A Potential-Minimizing Controller for Planning Multi-Agent Interactive Trajectories”. In: *Robotics: Science and Systems* (July 2021).
- [60] Mingxing Ke et al. “An EKF based Overlapping Coalition Formation Game for Cooperative Wireless Network Navigation”. In: *IET Communications* 15.19 (2021), pp. 2407–2424. DOI: <https://doi.org/10.1049/cmu2.12279>.
- [61] Kasra Khosoussi, Shoudong Huang, and Gamini Dissanayake. “A Sparse Separable SLAM Back-End”. In: *IEEE Transactions on Robotics* 32.6 (2016).
- [62] Markus Koschi et al. “Set-Based Prediction of Pedestrians in Urban Environments Considering Formalized Traffic Rules”. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (2018), pp. 2704–2711.
- [63] Forrest Laine et al. “The Computation of Approximate Generalized Feedback Nash Equilibria”. In: *arXiv preprint arXiv:2101.02900* (2021).
- [64] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. DOI: 10.1017/9781108571401.
- [65] Simon Le Cleac’h, Mac Schwager, and Zachary Manchester. “LUCIDGames: Online Unscented Inverse Dynamic Games for Adaptive Trajectory Prediction and Planning”. In: *Robotics and Automation Letters* 6.3 (2021), pp. 5485–5492.
- [66] J.J. Leonard and H.F. Durrant-Whyte. “Simultaneous Map Building and Localization for an Autonomous Mobile Robot”. In: *IEEE IROS*. Vol. 3. 1991, pp. 1442–7.
- [67] Stefan Leutenegger et al. “Keyframe-based Visual-Inertial Odometry using Nonlinear Optimization”. In: *The International Journal of Robotics Research* 34 (2015), pp. 314–334.
- [68] Ana Ley. “How Might Congestion Pricing Actually Work in New York?” In: (2023).
- [69] Mingyang Li and Anastasios I. Mourikis. “Improving the Accuracy of EKF-based Visual-Inertial Odometry”. In: *2012 IEEE International Conference on Robotics and Automation* (2012), pp. 828–835.
- [70] Mingyang Li and Anastasios I. Mourikis. “Optimization-based Estimator Design for Vision-aided Inertial Navigation: Supplemental Materials”. In: *Robotics: Science and Systems* (2012).

- [71] MLA Lourakis and Antonis A Argyros. “Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment?” In: *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*. Vol. 2. IEEE. 2005, pp. 1526–1531.
- [72] Chinmay Maheshwari et al. “Dynamic Tolling for Inducing Socially Optimal Traffic Loads”. In: *2022 American Control Conference (ACC)*. 2022, pp. 4601–4607. DOI: 10.23919/ACC53348.2022.9867193.
- [73] Michael Manville. “How and Why Would Congestion Pricing Work?” In: (2021).
- [74] Kostas Margellos and John Lygeros. “Hamilton-Jacobi Formulation for Reach-avoid Differential Games”. In: *Transactions on Automatic Control* 56.8 (2011), pp. 1849–1861.
- [75] Donald W Marquardt. “An algorithm for least-squares estimation of nonlinear parameters”. In: *Journal of the society for Industrial and Applied Mathematics* 11.2 (1963), pp. 431–441.
- [76] Peter S. Maybeck et al. *Stochastics Models, Estimation, and Control: Introduction*. 1979.
- [77] Hamid Mirzaei et al. “Enhanced Delta-tolling: Traffic Optimization via Policy Gradient Reinforcement Learning”. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 47–52. DOI: 10.1109/ITSC.2018.8569737.
- [78] Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. “A Time-dependent Hamilton-Jacobi Formulation of Reachable Sets for Continuous Dynamic Games”. In: *IEEE Transactions on automatic control* 50.7 (2005), pp. 947–957.
- [79] Anastasios I. Mourikis and Stergios I. Roumeliotis. “A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation”. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation* (2007), pp. 3565–3572.
- [80] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
- [81] M. Ono, L. Blackmore, and B. C. Williams. “Chance Constrained Finite Horizon Optimal Control with Nonconvex Constraints”. In: *Proceedings of the 2010 American Control Conference*. 2010, pp. 1145–1152.
- [82] Yuki Oyama, Yusuke Hara, and Takashi Akamatsu. “Markovian Traffic Equilibrium Assignment Based on Network Generalized Extreme Value Model”. In: *Transportation Research Part B: Methodological* 155 (2022), pp. 135–159.
- [83] Yuki Oyama and Eiji Hato. “A Discounted Recursive Logit Model for Dynamic Gridlock Network Analysis”. In: *Transportation Research Part C: Emerging Technologies* 85 (2017), pp. 509–527.

- [84] Yuki Oyama and Eiji Hato. “Prism-based Path Set Restriction for Solving Markovian Traffic Assignment Problem”. In: *Transportation Research Part B: Methodological* 122 (2019), pp. 528–546.
- [85] Jong-Shi Pang and Gesualdo Scutari. “Nonconvex Games with Side Constraints”. In: *SIAM Journal on Optimization* (2021).
- [86] C. Pek and M. Althoff. “Computationally Efficient Fail-safe Trajectory Planning for Self-driving Vehicles Using Convex Optimization”. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (2018), pp. 1447–1454.
- [87] Lasse Peters et al. “Inferring Objectives in Continuous Dynamic Games from Noise-Corrupted Partial State Observations”. In: *Robotics: Science and Systems*. 2021.
- [88] Michael JD Powell. “A new algorithm for unconstrained optimization”. In: *Nonlinear programming*. Elsevier, 1970, pp. 31–65.
- [89] Wei Qiu, Haipeng Chen, and Bo An. “Dynamic Electronic Toll Collection via Multi-Agent Deep Reinforcement Learning with Edge-Based Graph Convolutional Networks”. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 4568–4574. DOI: 10.24963/ijcai.2019/635. URL: <https://doi.org/10.24963/ijcai.2019/635>.
- [90] Benjamin Recht and Stephen Wright. *Optimization for Modern Data Analysis*. Cambridge University Press, 2021. ISBN: 1316518981.
- [91] Paul B. Reberdy and Naomi Ehrich Leonard. “Parameter Estimation in Softmax Decision-Making Models With Linear Objective Functions”. In: *IEEE Transactions on Automation Science and Engineering* 13 (2015), pp. 54–67.
- [92] Tim Roughgarden. “Algorithmic Game Theory”. In: *Communications of the ACM* 53.7 (2010), pp. 78–86.
- [93] William H. Sandholm. *Population Games And Evolutionary Dynamics*. Economic Learning and Social Evolution, 2010.
- [94] Shankar Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer, 1999.
- [95] Amay Saxena* et al. “Simultaneous Localization and Mapping: Through the Lens of Nonlinear Optimization”. In: *IEEE Robotics and Automation Letters* 7.3 (2022), pp. 7148–7155. DOI: 10.1109/LRA.2022.3181409.
- [96] D. Scaramuzza and F. Fraundorfer. “Visual Odometry, Part I: The First 30 Years and Fundamentals”. In: *IEEE RAM* 18.4 (2011), pp. 80–92.
- [97] Gabe Sibley, Larry H. Matthies, and Gaurav S. Sukhatme. “Sliding window filter with application to planetary landing.” In: *J. Field Robotics* 27.5 (2010), pp. 587–608. URL: <http://dblp.uni-trier.de/db/journals/jfr/jfr27.html#SibleyMS10>.
- [98] Joan Solà. “Simultaneous localization and mapping with the extended Kalman filter”. In: *arXiv* (2014).

- [99] Alan Wilbor Starr and Yu-Chi Ho. “Further Properties of Nonzero-sum Differential Games”. In: *Journal of Optimization Theory and Applications* 3.4 (1969), pp. 207–219.
- [100] Alan Wilbor Starr and Yu-Chi Ho. “Nonzero-sum Differential Games”. In: *Journal of optimization theory and applications* 3.3 (1969), pp. 184–206.
- [101] H. Strasdat, J.M.M. Montiel, and A. J. Davison. “Visual SLAM: Why Filter?” In: *IVC* 30.2 (2012), pp. 65–77. ISSN: 0262-8856.
- [102] Duy-Nguyen Ta, Marin Kobilarov, and Frank Dellaert. “A Factor Graph Approach to Estimation and Model Predictive Control on Unmanned Aerial Vehicles”. In: *International Conference on Unmanned Aircraft Systems (ICUAS)* (2014).
- [103] Chao Tang and Lihua Dou. “An Improved Game Theory-Based Cooperative Localization Algorithm for Eliminating the Conflicting Information of Multi-Sensors”. In: *Sensors* 20.19 (2020). ISSN: 1424-8220. DOI: 10.3390/s20195579. URL: <https://www.mdpi.com/1424-8220/20/19/5579>.
- [104] Akio Tanikawa, Hiro Mukai, and Min Xu. “Local Convergence of the Sequential Quadratic Method for Differential Games”. In: *Transactions of the Institute of Systems, Control and Information Engineers* 25.12 (2012), pp. 349–357.
- [105] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN: 0262201623.
- [106] S. Tully et al. “Iterated Filters for Bearing-only SLAM”. In: *2008 IEEE International Conference on Robotics and Automation*. 2008, pp. 1442–1448. DOI: 10.1109/ROBOT.2008.4543405.
- [107] S. Umeyama. “Least-Squares Estimation of Transformation Parameters Between Two Point Patterns”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.4 (1991), pp. 376–380. DOI: 10.1109/34.88573.
- [108] S. Vaskov et al. “Not-at-Fault Driving in Traffic: A Reachability-Based Approach”. In: 2019, pp. 2785–2790.
- [109] Yiheng Wang, Hexi Jin, and Guanjie Zheng. “CTRL: Cooperative Traffic Tolling via Reinforcement Learning”. In: *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. 2022, pp. 3545–3554.
- [110] Zijian Wang, Riccardo Spica, and Mac Schwager. “Game Theoretic Motion Planning for Multi-Robot Racing”. In: (2018).
- [111] Zijian Wang, Riccardo Spica, and Mac Schwager. “Game Theoretic Motion Planning for Multi-robot Racing”. In: *Distributed Autonomous Robotic Systems*. Springer, 2019, pp. 225–238.
- [112] Hai Yang and Hai-Jun Huang. “Principle of Marginal-Cost Pricing: How Does it Work in a General Road Network?” In: *Transportation Research Part A: Policy and Practice* 32.1 (1998), pp. 45–54.

- [113] Ming-Yuan Yu, R. Vasudevan, and Matthew Johnson-Roberson. “Risk Assessment and Planning with Bidirectional Reachability for Autonomous Driving”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020), pp. 5363–5369.
- [114] Alessandro Zanardi et al. “Urban Driving Games With Lexicographic Preferences and Socially Efficient Nash Equilibria”. In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 4978–4985.
- [115] J. Zhang et al. “VDO-SLAM: A Visual Dynamic Object-aware SLAM System”. In: *arXiv*. 2020. arXiv: 2005.11052 [cs.R0].
- [116] Yetong Zhang et al. “MR-iSAM2: Incremental Smoothing and Mapping with Multi-Root Bayes Tree for Multi-Robot SLAM”. In: *International Conference on Intelligent Robots and Systems (IROS)* (2021), pp. 8671–8678.
- [117] Zhengyou Zhang. “Parameter Estimation Techniques: a Tutorial with Application to Conic Fitting”. In: *Image and Vision Computing* 15.1 (1997), pp. 59–76. ISSN: 0262-8856. DOI: [https://doi.org/10.1016/S0262-8856\(96\)01112-2](https://doi.org/10.1016/S0262-8856(96)01112-2). URL: <http://www.sciencedirect.com/science/article/pii/S0262885696011122>.