Distributed Extremum-Seeking Control with Applications to High-Altitude Balloons

by

Isaac Vandermeulen

A thesis submitted to the Department of Chemical Engineering
in conformity with the requirements for
the degree of Master of Applied Science

Queen's University
Kingston, Ontario, Canada
August 2016

Abstract

The real-time optimization of large-scale systems is a difficult problem due to the need for complex models involving uncertain parameters and the high computational cost of solving such problems by a decentralized approach. Extremum-seeking control (ESC) is a model-free real-time optimization technique which can estimate unknown parameters and can optimize nonlinear time-varying systems using only a measurement of the cost function to be minimized. In this thesis, we develop a distributed version of extremum-seeking control which allows large-scale systems to be optimized without models and with minimal computing power.

First, we develop a continuous-time distributed extremum-seeking controller. It has three main components: consensus, parameter estimation, and optimization. The consensus provides each local controller with an estimate of the cost to be minimized, allowing them to coordinate their actions. Using this cost estimate, parameters for a local input-output model are estimated, and the cost is minimized by following a gradient descent based on the estimate of the gradient. Next, a similar distributed extremum-seeking controller is developed in discrete-time.

Finally, we consider an interesting application of distributed ESC: formation control of high-altitude balloons for high-speed wireless internet. These balloons must be steered into a favourable formation where they are spread out over the Earth and provide coverage to the entire planet. Distributed ESC is applied to this problem, and is shown to be effective for a system of 1200 ballons subjected to realistic wind currents. The approach does not require a wind model and uses a cost function based on a Voronoi partition of the sphere. Distributed ESC is able to steer balloons from a few initial launch sites into a formation which provides coverage to the entire Earth and can maintain a similar formation as the balloons move with the wind around the Earth.

Acknowledgments

First, I would like to thank my supervisors, Dr. Martin Guay and Dr. Jim McLellan, for their support, guidance, and advice. Thanks to Martin for introducing me to extremum-seeking control, nonlinear control, and countless other interesting topics and the daily helpful discussions over the past two years. Thanks to Jim for introducing me to the wonderful world of research and for all the encouragement over the years.

I would also like to thank all of my friends who have made my time at Queen's enjoyable. In particular, I'd like to thank the people who I've had the pleasure of sharing an office with in Dupuis G37. Thanks to Ehsan, Hussain, Judith, Sam, Dan, Seb, Nic, Sean, Walter, Ryan, and Falco for keeping me sane while doing research and making the office such a great place to work.

Finally, I would like to thank my parents for their love and support during my childhood. Their encouragement helped me develop as a young mathematician, and gave a great foundation to be able to do the work that went into this thesis.

Contents

Abstract . . . . **i**

Acknowledgments . . . . **iii**

Contents . . . . **iv**

List of Tables . . . . **vii**

List of Figures . . . . **viii**

List of Abbreviations . . . . **x**

List of Symbols . . . . **xi**

Chapter 1: Introduction **1**
1.1 Motivation 1
1.2 Statement of contributions 4
1.3 Organization of thesis 5

Chapter 2: Literature review **8**
2.1 Extremum-seeking control 8
2.2 Distributed control & consensus 14
2.3 Formation & coverage control 19

Chapter 3: Continuous-time distributed ESC **23**
3.1 Introduction 23
3.2 Problem description 25
3.3 Distributed extremum-seeking controller 28
3.3.1 Consensus algorithm 28
3.3.2 Input-output dynamics of agent i 31
3.3.3 Time-varying parameter estimation 32
3.3.4 Proportional-integral extremum-seeking control 36

3.3.5 Convergence analysis 37
3.4 Simulation examples 40
3.4.1 Stable 25 agent linear system 41
3.4.2 Unstable 5 agent linear system 43
3.4.3 Unstable 5 agent nonlinear system 47
3.5 Conclusion 49

Chapter 4: Discrete-time distributed ESC 51
4.1 Introduction 51
4.2 Problem description 53
4.3 Discrete-time controller design 57
4.3.1 Dynamic average consensus 58
4.3.2 Time-varying parameter estimation 59
4.3.3 Proportional-integral extremum-seeking control 61
4.4 Convergence analysis 63
4.5 Simulation examples 67
4.5.1 100 agent LQ system 68
4.5.2 5 agent nonlinear system 71
4.6 Conclusion 75

Chapter 5: Formation control of high-altitude balloons 77
5.1 Introduction 77
5.2 Problem definition 83
5.3 Balloon dynamics model 86
5.3.1 Coordinate charts 87
5.3.2 General nonlinear time-varying model 88
5.3.3 Data-based time-invariant model 90
5.3.4 Control-affine model 90
5.4 Network coverage cost model 91
5.4.1 Voronoi partitions 93
5.4.2 Distance-based cost function 96
5.4.3 Voronoi centroid-based cost function 97
5.4.4 Voronoi area-based cost function 98
5.4.5 Internet bandwidth-based cost function 99
5.4.6 Population-based cost function 101
5.4.7 Summary of cost functions 102
5.5 Distributed ESC for balloons 103
5.5.1 Convergence analysis 107
5.6 Simulation results 109
5.6.1 Continuous-time simulation results 109

5.6.2 Discrete-time simulation results 112
5.6.3 Maintaining a formation 116
5.6.4 Launching the balloons 120
5.7 Conclusion 124

Chapter 6: Conclusion 126
6.1 Summary 126
6.2 Future work 129

Bibliography 131

Appendix A: Proof of Theorem 3.1 145

Appendix B: Proof of Lemma 4.1 158

Appendix C: Proof of Lemma 4.2 162

Appendix D: Proof of Lemma 4.3 168

Appendix E: Proof of Lemma 4.4 171

Appendix F: Proof of Theorem 4.1 177

List of Tables

3.1 Continuous-time tuning parameters for the 25-agent system 42
3.2 Initial conditions for the 25-agent continuous-time system 42
3.3 Continuous-time tuning parameters for the unstable system 44
3.4 Initial conditions for the unstable continuous-time system 45
3.5 Continuous-time tuning parameters for the nonlinear system 48
3.6 Initial conditions for the continuous-time nonlinear system 48

4.1 Discrete-time tuning parameters for the 100-agent system 70
4.2 Initial conditions for the 100-agent discrete-time system 71
4.3 Discrete-time tuning parameters for the nonlinear system 73
4.4 Initial conditions for the nonlinear discrete-time system 73

5.1 Initial conditions of balloons with continuous-time controllers 111
5.2 Tuning parameters for the continuous time balloon controllers 112
5.3 Tuning parameters for the discrete-time balloon formations 116
5.4 Balloon launch site coordinates 121

List of Figures

1.1 Centralized vs. decentralized vs. distributed control 2

2.1 Block diagram of a general form of ESC 10
2.2 Block diagram of the ESC proposed by Krstić and Wang 11
2.3 Block diagram of the ESC proposed by Choi *et al.* 12
2.4 Block diagram of the PI ESC proposed by Guay and Dochain 13
2.5 General scheme of a distributed system 15
2.6 Distributed system as seen by one agent 16
2.7 Examples graph and digraph 17

3.1 Block diagram of continuous-time distributed PI ESC 38
3.2 Cyclic graph for 25 agents 42
3.3 25-agent system with distributed ESC in continuous-time 43
3.4 Cyclic graph for 5 agents 45
3.5 Unstable linear system with distributed ESC in continuous-time . . . 46
3.6 Linear graph for 5 agents 48
3.7 Nonlinear system with distributed ESC in continuous-time 50

4.1 Block diagram of discrete-time distributed ESC 58
4.2 Cyclic graph for 100 agents 70
4.3 100-agent system with distributed ESC in discrete-time 72
4.4 Linear graph for 5 agents 74
4.5 Nonlinear system with distributed ESC in discrete-time 75

5.1 Internet penetration by continent 78
5.2 Population and internet user growth 79
5.3 Surface vs. stratospheric wind currents 82
5.4 Components of wind velocity 92
5.5 Good and bad balloon formations 92
5.6 Example of a spherical Voronoi partition 94
5.7 Construction of a spherical Voronoi partion 95
5.8 Population density of Earth 102
5.9 Cost trajectories for balloons with simple wind model 113

5.10 Initial and final balloon configurations with simple wind model 114
5.11 Maps of balloon trajectories of randomly placed balloons 118
5.12 Cost trajectories for randomly placed balloons 119
5.13 Balloon launch site locations 120
5.14 Maps of balloon trajectories when launched from cities 122
5.15 Cost trajectories for balloons launched from cities 123

List of Abbreviations

Broadband over **P**ower**l**ine	BPL
Decision **M**aker	DM
Digital **S**ubscriber **L**ine	DSL
Extremum **S**eeking **C**ontrol(ler)	ESC
Linear **Q**uadratic	LQ
Multi **A**gent **S**ystem	MAS
Proportional-**I**ntegral	PI
Recursive **L**east **S**quares	RLS
Real **T**ime **O**ptimization	RTO

List of Symbols

Arbitrary constants obtained by completing the squares	$a_1, \ldots, a_{19}$
Altitude of the i^{th} balloon	a_i
i, j^{th}-Element of the adjacency matrix	$a_{i,j}$
Adjacency matrix associated with communication network	$\boldsymbol{A}$
Area of i^{th} balloon's Voronoi cell	A_i
Total surface area of Earth	A_t
Bandwidth of data flowing through balloon i	b_i
Nominal estimate of the total bandwidth of the internet	$\widehat{b}_i$
Open geodesic ball of radius r centered at q	$\mathcal{B}_r(q)$
Positive constants defined to simplify Lyapunov expressions	$c_1, \ldots, c_{50}$
Vector of all dither signals	$\boldsymbol{d}$
Dither signals used by agent i	$\boldsymbol{d}_i$
i^{th}-Diagonal element of the degree matrix	$d_{i,i}$
Degree matrix associated with communication network	$\boldsymbol{D}$
Dither signal amplitude	D
Estimation error	e_i
Edge of a graph from vertex v_i to vertex v_j	$e_{i,j}$
Edge set of a graph	$\mathcal{E}$
Vector field describing Earth's wind currents at isobaric pressure u	$f\{u\}$
Drift vector field approximating Earth's wind currents	f_0
Control vector field approximating Earth's wind currents	f_1
Component of drift vector field	f_i
Drift vector field	$\boldsymbol{f}$
Drift vector field associated with the zero dynamics	$\boldsymbol{f}_z$
Drift vector field associated with the input-output dynamics	$\boldsymbol{f}_\xi$
Vector field consisting of p copies of Earth's wind currents	$F\{\boldsymbol{u}\}$
Drift vector field approximating $F\{\boldsymbol{u}\}$	F_0
Control vector field approximating $F\{\boldsymbol{u}\}$	F_1
Component of control vector fields (row vector)	$\boldsymbol{g}_i$
Control vector fields associated with agent i (column vector)	$\boldsymbol{G}_i$
Matrix of all control vector fields	$\boldsymbol{G}$
Matrix of control vector fields associated with input-output dynamics	$\boldsymbol{G}_\xi$

Graph	$\mathcal{G}$
Riemannian metric	$\mathbb{G}$
Local cost function measured by agent i	h_i
Vector of local cost functions	$\boldsymbol{h}$
Vector of local cost functions associated with input-output dynamics	$\boldsymbol{h}_\xi$
Total cost function	H
Agent or state counter	i
Identity matrix	$\boldsymbol{I}$
Secondary agent or state counter	j
Total cost	J
Agent i's estimate of the total cost	$\widehat{\boldsymbol{J}}_i$
Vector of all agents' total cost estimates	$\widehat{\boldsymbol{J}}$
Consensus estimation error associated with agent i	$\widetilde{\boldsymbol{J}}_i$
Vector of all agents' consensus estimation errors	$\widetilde{\boldsymbol{J}}$
Index set of balloons adjacent to balloon i in the Delaunay triangulation	$\mathbb{J}_i$
Time-step counter	k
Parameter estimation proportional gain. Must be between 0 and 1	K
Weight for the area-based component of a balloon's local cost	K_A
Weight for the centroid-based component of a balloon's local cost	K_C
Exponential decay parameter for the continuous time consensus error	K_c
Proportional gain for ESC gradient descent	K_g
Minimum proportional gain need to stabilize the steady-state manifold	K_g^*
Weight for the population-based component of a balloon's local cost	K_P
Continuous time parameter estimation gain	K_T
Steady-state cost function	ℓ
Desired distance between balloons	ℓ^*
Distance function associated with Riemannian metric $\mathbb{G}$	$\ell_{\mathbb{G}}$
i^{th} row of the Laplacian $\boldsymbol{L}$	$\boldsymbol{\ell}_{i,:}$
$(i,j)^{\text{th}}$ element of the Laplacian $\boldsymbol{L}$	$\ell_{i,j}$
Maximum distance over which balloons can communicate	$\ell_{\max}$
Lipschitz constant for the drift vector field	L_f
Lipschitz constant for the control vector fields	L_G
Lipschitz constant for the total cost function	L_H
Lipschitz constant for the steady-state map	L_π
Laplacian matrix associated with communication network	$\boldsymbol{L}$
Lie derivative of H along vector field $\boldsymbol{f}$	$\mathcal{L}_{\boldsymbol{f}}H$
Lie derivative of H along vector fields $\boldsymbol{G}$	$\mathcal{L}_{\boldsymbol{G}}H$
Number of inputs controlled by agent i	m_i
Number of inputs to the system	m
Dimension of the state space	n

Dimension of the i^{th} state	n_i
Number of launch sites	n_ℓ
Number of vertices in a Voronoi polygon	n_v
Dimension of the subspace associated with the zero dynamics	n_z
Used in an ϵ-δ type argument for the consensus error	N
Set of natural number	$\mathbb{N}$
Landau symbol (Big O notation)	$\mathcal{O}$
Number of agents	p
Population contained the i^{th} balloon's Voronoi cell	p_i
Total population of Earth	p_t
Probability density function of an event being observed on Q	P
System configuration, i.e. point in configuration space	q
Centroid of balloon i' s Voronoi cell	$q_{c,i}$
Configuration of balloon	q_i
Point directly between balloons i and j	$q_{i,j}$
Balloon i's estimate of the centroid of its Voronoi cell	$\widehat{q}_{c,i}$
Configuration space	Q
Overall configuration space of system	Q_{sys}
Number of internet users connected to balloon i	r_i
Estimate of the number of internet users	$\widehat{r}_t$
Earth's radius	$R_\oplus$
Set of real numbers	$\mathbb{R}$
Miscellaneous counter	s
Binary relation defining the steady-state manifold	S
2-sphere	$\mathbb{S}^2$
Time	t
Time horizon used in the persistence of excitation assumption	T
Tangent bundle	T
East-West (u-)component of the wind in m/s	u_{wind}
Inputs controlled by agent i	$\boldsymbol{u}_i$
Vector of all agents' inputs	$\boldsymbol{u}$
Input bias for agent i's inputs	$\widehat{\boldsymbol{u}}_i$
Vector of all agents' input biases	$\widehat{\boldsymbol{u}}$
Deviation of agent i's input bias from its optimal value	$\widetilde{\boldsymbol{u}}_i$
Vector of all agents' input deviations	$\widetilde{\boldsymbol{u}}$
Agent i's input which will minimize the steady-state cost	$\boldsymbol{u}_i^*$
Vector of all agents' optimal inputs	$\boldsymbol{u}^*$
Chart domain for the standard longitude-latitude chart	U
Chart domain for the secondary chart needed to make an atlas on $\S^2$	$\widetilde{\mathsf{U}}$
Set of allowable values for the i^{th} input	$\mathbb{U}_i$

Set of allowable inputs	$\mathbb{U}$
Vertex of a graph	v_i
North-South (v-)component of the wind in m/s	v_{wind}
Vector of the i^{th} parameter estimation deviance and i^{th} true parameter	$\boldsymbol{v}_{\theta,i}$
Vector of the deviance of $\boldsymbol{x}$ from the steady state and $\widehat{u}$ from the optimum	$\boldsymbol{v}_x$
Vector of the deviance of $\boldsymbol{z}$ from the steady state and $\widehat{u}$ from the optimum	$\boldsymbol{v}_z$
Lyapunov function for the entire extremum seeking controller	V_{ESC}
Lyapunov function for parameter estimation and stabilization	V_{Stab}
Lyapunov function for the parameter estimation algorithm	V_{PE}
Lyapunov function for the deviation of the input bias from its optimum	V_u
Lyapunov function for the deviation of the state from steady-state	V_x
Lyapunov function for the zero dynamics	V_z
Lyapunov function for the auxiliary variable estimation errors	V_η
Lyapunov function for the parameter estimation errors	V_θ
Vertex set of a graph	$\mathcal{V}$
Filtered version of agent i's regressor vector	$\boldsymbol{w}_i$
Weight associated with edge $e_{i,j}$ in a weighted graph	$w_{i,j}$
First coordinate of a point in $\mathbb{R}^3$	x
State	$\boldsymbol{x}$
Optimal steady-state value of $\boldsymbol{x}$	$\boldsymbol{x}^*$
Deviation of state from its optimal value	$\widetilde{\boldsymbol{x}}$
Component of state used in mean value theorem	$\overline{x}_i$
Value of state used in mean value theorem	$\overline{\boldsymbol{x}}$
Set of allowable values of the state	$\mathbb{X}$
Set of time-varying vector fields on Q	$\mathfrak{X}(\mathbb{R}, \mathsf{Q})$
Set of (time-invariant) vector fields on Q	$\mathfrak{X}(\mathsf{Q})$
Second coordinate of a point in $\mathbb{R}^3$	y
Area based local cost measured by the i^{th} balloon	$y_{A,i}$
Bandwidth based local cost measured by the i^{th} balloon	$y_{B,i}$
Distance to centroid based local cost measured by the i^{th} balloon	$y_{C,i}$
Distance between balloons based local cost measured by the i^{th} balloon	$y_{D,i}$
Population based local cost measured by the i^{th} balloon	$y_{P,i}$
Users based local cost measured by the i^{th} balloon	$y_{U,i}$
Local cost measured by agent i	y_i
Vector of local costs	$\boldsymbol{y}$
Predicted average cost obtained using the parameter estimate	$\widehat{\boldsymbol{y}}_i$
Third coordinate of a point in $\mathbb{R}^3$	z
States associated with the zero dynamics in normal form	$\boldsymbol{z}$
Set of integers	$\mathbb{Z}$

Recursive least squares forgetting factor	α
i^{th} interior angle of a Voronoi cell	α_i
Lower bound coefficient for the gradient of ℓ	β_1
Lower bound coefficient for $\Omega(\boldsymbol{x}, \boldsymbol{z})$	β_2
$\boldsymbol{x}$-lower bound coefficient for $\Omega(\boldsymbol{x}, \boldsymbol{z})$	$\beta_{2,x}$
$\boldsymbol{z}$-lower bound coefficient for $\Omega(\boldsymbol{x}, \boldsymbol{z})$	$\beta_{2,z}$
Upper bound coefficient for $\Omega(\boldsymbol{x}, \boldsymbol{z})$	β_3
$\boldsymbol{x}$-upper bound coefficient for $\Omega(\boldsymbol{x}, \boldsymbol{z})$	$\beta_{3,x}$
$\boldsymbol{z}$-upper bound coefficient for $\Omega(\boldsymbol{x}, \boldsymbol{z})$	$\beta_{3,z}$
Upper bound coefficient for $\Delta\Omega(\boldsymbol{x}, \boldsymbol{z})$	β_4
$\boldsymbol{x}$-upper bound coefficient for $\Delta\Omega(\boldsymbol{x}, \boldsymbol{z})$	$\beta_{4,x}$
$\boldsymbol{z}$-upper bound coefficient for $\Delta\Omega(\boldsymbol{x}, \boldsymbol{z})$	$\beta_{4,z}$
Geodesic path connecting balloons i and j	$\gamma_{i,j}$
Lower bound for sums of $\boldsymbol{w}_i$ for persistence of excitation	γ_w^-
Upper bound for sums of $\boldsymbol{w}_i$ for persistence of excitation	γ_w^+
Norm of the control vector fields at the steady-state optimum	γ_G
Lower bound for sums of $\boldsymbol{\phi}_i$ or $\boldsymbol{u}_i$ for persistence of excitation	γ_u^-
Upper bound for sums of $\boldsymbol{\phi}_i$ or $\boldsymbol{u}_i$ for persistence of excitation	γ_u^+
Lower bound for eigenvalues of $\boldsymbol{\Lambda}_x$ and Λ_z	γ_v^-
Upper bound for $\Vert\widehat{\boldsymbol{\theta}}\Vert$ used in the projection algorithm	γ_θ
Lower bound for eigenvalues of $\boldsymbol{\Lambda}_\theta$	γ_Λ^-
Lower bound for eigenvalues of $\boldsymbol{\Sigma}_i$	γ_Σ^-
Upper bound for eigenvalues of $\boldsymbol{\Sigma}_i$	γ_Σ^+
Great circle which bounds the hemisphere closer to balloon i than balloon j	$\overline{\gamma}_{i,j}$
Voronoi cell associated with the i^{th} balloon	Γ_i
Balloon i's estimate of its Voronoi cell	$\widehat{\Gamma}_i$
Parameter between 0 and 1 used in mean value theorem	δ_i
Change in variable between adjacent time steps	Δ
Singular perturbation parameter	ϵ
Auxiliary variable	η_i
Estimate of auxiliary variable	$\widehat{\eta}_i$
Auxiliary variable estimation error	$\widetilde{\eta}_i$
Positive constant used in the parameter estimation Lyapunov function	ζ
Parameter describing changes in J which cannot be influenced by Agent i	$\theta_{0,i}$
Parameter describing changes in $\boldsymbol{J}$ which can be influenced by Agent i	$\boldsymbol{\theta}_{1,i}$
Vector of $\theta_{0,i}$ and $\boldsymbol{\theta}_{1,i}$. Parameter vector in RLS	$\boldsymbol{\theta}_i$
Vector of each agent's parameter $\boldsymbol{\theta}_{1,i}$	$\boldsymbol{\theta}_1$
Agent i's estimate of $\boldsymbol{\theta}_{0,i}$ obtained from RLS	$\widehat{\theta}_{0,i}$
Agent i's estimate of $\boldsymbol{\theta}_{1,i}$ obtained from RLS	$\widehat{\boldsymbol{\theta}}_{1,i}$
Parameter estimate obtained from the RLS algorithm by agent i	$\widehat{\boldsymbol{\theta}}_i$

Estimation error associated with agent i's estimate of $\boldsymbol{\theta}_{0,i}$	$\widetilde{\theta}_{0,i}$
Estimation error associated with agent$i$'s estimate of $\boldsymbol{\theta}_{1,i}$	$\widetilde{\boldsymbol{\theta}}_{1,i}$
Parameter estimation error associated with agent i	$\widetilde{\boldsymbol{\theta}}_i$
Used in an ϵ-δ type argument for the estimation error	ι
Input gain in the continuous time consensus algorithm	κ_0
Integral gain constant used in the consensus algorithm	κ_I
Proportional gain constant used in the consensus algorithm	κ_P
Longitude	λ
Initial longitude of i^{th} balloon	$\lambda_{0,i}$
Desired longitude of i^{th} balloon	$\lambda_{d,i}$
Longitude of i^{th} balloon	λ_i
Longitude-like coordinate in the secondary, rotated chart	$\widetilde{\lambda}$
i^{th} smallest eigenvalue of the symmetric Laplacian	$\widehat{\lambda}_i$
Constant matrix for the quadratic term involving $\boldsymbol{v}_x$	$\boldsymbol{\Lambda}_x$
Constant matrix for the quadratic term involving $\boldsymbol{v}_z$	$\boldsymbol{\Lambda}_x$
Constant matrix for the quadratic term involving $\boldsymbol{v}_{\theta,i}$	$\boldsymbol{\Lambda}_\theta$
Variable defined to simplify Lyapunov expressions	$\mu_{1,i}$
Variable defined to simplify Lyapunov expressions	$\mu_{2,i}$
Variable defined to simplify Lyapunov expressions	$\mu_{3,i}$
Variable defined to simplify Lyapunov expressions	$\mu_{4,i}$
Vector of exponentially decaying upper bounds for the consensus error	$\boldsymbol{\nu}$
Initial upper bound for the consensus error	ν_0
Upper bound for the i^{th} consensus error	ν_i
States associated with the input-output dynamics in normal form	$\boldsymbol{\xi}$
Steady-state map	$\boldsymbol{\pi}$
Steady-state map associated with $\boldsymbol{x}$	$\boldsymbol{\pi}_x$
Steady-state map associated with $\boldsymbol{z}$	$\boldsymbol{\pi}_z$
Steady-state map associated with $\boldsymbol{\xi}$	$\boldsymbol{\pi}_\xi$
i^{th} forward kinematic map	Π_i
Integrator variable used by the consensus algorithm	$\boldsymbol{\rho}$
Initialization value for covariance matrices	σ_0
Small parameter used to ensure the invertibility of $\boldsymbol{\Sigma}_i$	σ_1
Covariance matrix used by agent i's RLS	$\boldsymbol{\Sigma}_i$
Integral time constant used for the ESC gradient descent	τ_I
Probability density function of an event occuring on Q	ϕ
Vector of 1 and $\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i$. Regressor vector in RLS	$\boldsymbol{\phi}_i$
Latitude	φ
Initial latitude of i^{th} balloon	$\varphi_{0,i}$
Desired latitude of i^{th} balloon	$\varphi_{d,i}$
Latitude of i^{th} balloon	φ_i

Latitude-like coordinate in the secondary, rotated chart	$\widetilde{\varphi}$
Upper bound for the drift component of $\dot{\theta}$	χ_0
Upper bound for the control component of $\dot{\theta}$	χ_1
Normal form state diffeomorphism	$\boldsymbol{\psi}$
Component of $\boldsymbol{\psi}$ associated with the zero dynamics	$\boldsymbol{\psi}_z$
Component of $\boldsymbol{\psi}$ associated with the input-output dynamics	$\boldsymbol{\psi}_\xi$
Frequency of the i^{th} dither signal	ω_i
Overall Lyapunov function for stabilizability	Ω

Transpose	$^\top$
Gradient	∇
Union	$\cup$
Intersection	$\cap$
Empty set	$\{\}$
Matrix or vector consisting entirely of zeros	$\mathbf{0}$
Vector consisting entirely of ones	$\mathbf{1}$
Euclidean 2-norm	$\|\cdot\|$
Maximum out-degree of a graph with Laplacian $\boldsymbol{L}$	$\deg^+(\boldsymbol{L})$
Square matrix with elements of the vector $\boldsymbol{v}$ on its diagonal	$\operatorname{diag}(\boldsymbol{v})$
Lipschitz projection of $\cdot$ onto ball of radius r	$\operatorname{Proj}_r(\,\cdot\,)$
Spectrum (set of eigenvalues) of matrix $\boldsymbol{A}$	$\operatorname{spec}(\boldsymbol{A})$

Chapter 1

Introduction

1.1 Motivation

The real-time optimization (RTO) of large-scale systems is an important but challenging problem in control engineering. RTO is a closed-loop control technique which improves system performance by minimizing a measurable cost function in real-time. Existing RTO techniques, such as model predictive control [3, 30, 110, 132], stochastic optimization [20, 144], and dynamic programming [6, 7, 32], are computationally intensive. The number of computations required—and thus computation time—typically grows rapidly with the dimension of the system. These techniques are therefore impractical for large-scale systems such as chemical plants, multi-robot systems, and traffic planning which involve high-dimensional state spaces. Without effective RTO techniques, large-scale systems often suffer from suboptimal performance.

Distributed real-time optimization is an alternative to conventional RTO which is better suited to large-scale systems. A distributed system consists of several controllers, typically referred to as agents or decision makers (DMs), which cooperate to achieve an overall control objective [125, 130]. Each agent makes some local measurements, communicates with other DMs, and implements a control action based on its

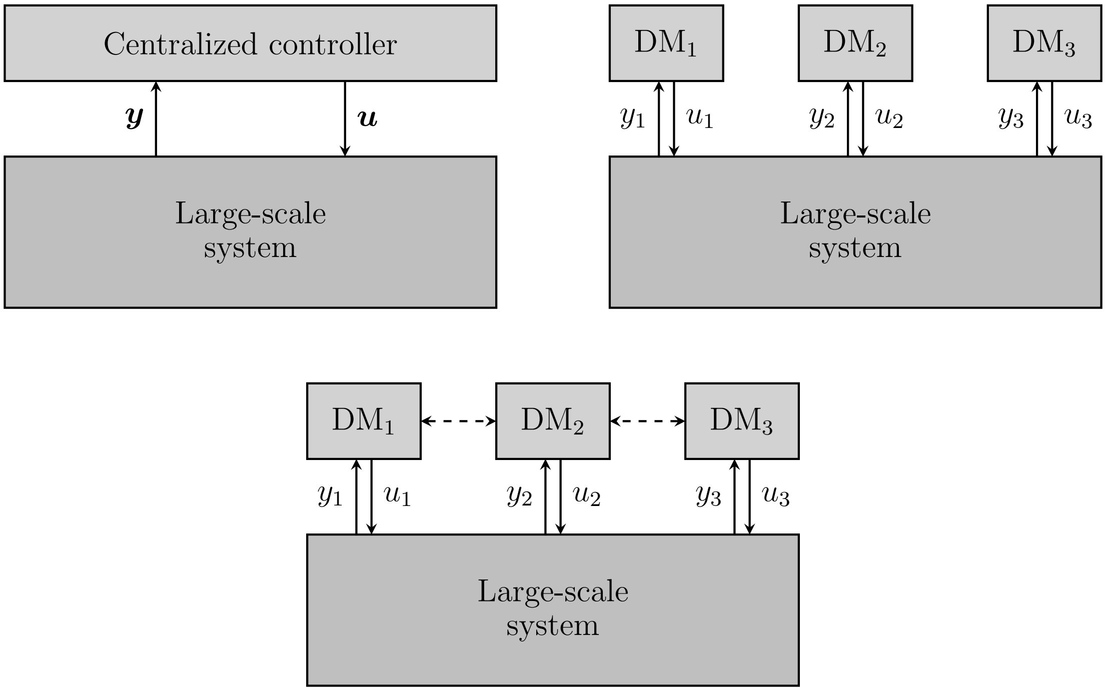


Figure 1.1: Comparison of centralized (top left), decentralized (top right), and distributed (bottom) control architectures.

local information. Distributed control can be contrasted with centralized and decentralized control (Figure 1.1). Centralized control consists of one DM which performs all calculations; Decentralized control consists of several DMs but no communication. The key to a successful distributed control algorithm is the communication network. Through communication, agents can coordinate their actions and achieve overall objectives; without communication, agents only true to achieve their own local objectives, resulting in a competitive system with worse overall performance.

Extremum-seeking control (ESC) is a model-free RTO technique. ESC works by introducing a small perturbation, called a dither signal, to the system to generate data and estimate a simple model which is locally valid. Based on this model, the ESC moves in the direction which minimizes the cost function. As the system changes, the local model is updated so that it remains locally valid. The main advantage of ESC is that it is a model-free technique for nonlinear, time-varying systems, and can therefore be applied to a large class of systems while avoiding the need to model the system, which is usually the most difficult and costly step of RTO. Existing ESC techniques are centralized (see Section 2.1), consisting of a single DM which measures the cost function and controls all of the control inputs.

The main topic of this thesis is distributed ESC. A distributed extremum-seeking control system consists of several agents which each measure a local cost and control one or more control inputs. The objective of the network of ESC agents is to minimize the total cost which is the sum of the local costs. The main difference between distributed and centralized ESC is the use of a consensus algorithm which provides each agent with an estimate of the total cost. Each agent uses ESC to minimize the estimate of the total cost. Collectively, the agents achieve overall objective of minimizing the total cost, while the computational burden is shared between several agents. In this thesis, distributed ESC is developed in both continuous-time and discrete-time.

At the end of this thesis, we apply distributed ESC to one useful real-world application: formation control of high-altitude balloons. These balloons float in the stratosphere and can connect wirelessly to internet users on Earth to provide them with high-speed internet access. One such project is Google's Project Loon which intends to use several thousand balloons to provide high-speed internet to users anywhere on Earth [98]. One of the main challenges (see Chapter 5) for such a fleet of balloons is maintaining a formation where the balloons are spread out enough to provide adequate coverage everywhere on Earth. The balloons ride the Earth's wind currents and can change their velocity by pumping helium in or out of the balloon to move up or down into different wind currents. In this thesis, distributed ESC is used to solve this formation control problem. This approach allows each balloon to control its own motion without needing a model of the wind currents and results in each balloon moving in a direction which is beneficial for the formation of the overall network.

1.2 Statement of contributions

Several of the results from this thesis have expanded the body of knowledge in distributed RTO or formation control, and have been published in other places. These results are:

1. Section 3.3 involves the development of a version continuous-time distributed ESC that can be used to optimize large-scale systems. The initial result was only valid for stable systems, such as the 25-agent stable linear system in Subsection 3.4.1. This approach has the advantage of allowing the controller to optimize an unknown large-scale system in a distributed manner, using only measurements of a local cost function. These results were published in and presented at the 2015 American Control Conference in Chicago, Illinois [50].

2. The results from Section 3.3 can also be applied to the stabilization of unknown, slowly unstable systems. The simulation results in Subsection 3.4.2 and Subsection 3.4.3 show the effectiveness of this technique for unstable and nonlinear systems. These results were published and presented at the 2015 International Symposium on Advanced Control of Chemical Processes in Whistler, British

Columbia [51].

3. Chapter 4 extends the distributed ESC results to discrete-time. The discrete-time controller uses different parameter estimation and consensus algorithms than a discretization of the continuous-time algorithms. These results have been accepted for publication and presentation at the 2016 IFAC Symposium on Nonlinear Control Systems in Monterey, California [126]

4. Subsection 5.6.1 describes the application of continuous-time ESC to the balloon formation control problem. This approach does not require knowledge of wind currents and was shown in Subsection 5.6.1 to be able to steer a system of 20 balloons to the optimal dodecahedral formation. This work has been accepted for publication and presentation at the 2016 American Control Conference in Boston, Massachusetts [127].

5. Subsection 5.6.2 describes the application of discrete-time ESC to the balloon formation control problem. Using a reasonable 6 minute time step and realistic wind models, it is shown in Subsection 5.6.2 that the discrete-time controller is able to maintain a randomized formation and that it can steer balloons from several initial launch sites into a favourable formation. This work has been submitted for publication and presentation at the 2016 IEEE Conference on Decisions and Control in Las Vegas, Nevada [128].

1.3 Organization of thesis

The remainder of this thesis is organized into five chapters and six appendices.

Chapter 2 is a literature review. It contains background information relevant to the three main chapters, as well as a survey of the current state of the art in ESC and formation control. In particular, the literature review covers the topics of extremum-seeking control, distributed control, consensus algorithms, formation control, and coverage control.

Chapter 3 contains the development of distributed ESC in continuous-time. First, the problem is formally defined and the necessary assumptions are stated. Next, the controller is designed, using a dynamic average consensus algorithm, time-varying parameter estimation algorithm, and a proportional-integral (PI) ESC technique. This chapter includes a formal proof of the convergence of the algorithm, and simulation results for large-scale, unstable, and nonlinear systems.

Chapter 4 continues the development of distributed ESC, this time in discrete-time. The techniques are developed by directly considering a discrete-time system, instead of simply discretizing a continuous-time system. The controller resembles the controller developed in Chapter 3, but it uses different consensus and parameter estimation algorithms. Again, a proof of the algorithm's convergence and simulation results for large-scale and nonlinear systems are provided.

Chapter 5 concerns the application of distributed ESC to the formation control of high-altitude balloons. This chapter begins with a discussion on world-wide internet penetration and a survey of existing internet technology to provide justification for the use a fleet of high-speed internet balloons. Next, the formation control problem and its objectives are defined. Since simulation results require a model of Earth's wind currents, a discussion on wind current models follows. Next, several cost functions which measure how well the balloons provide global coverage are introduced. This chapter includes analysis of the application of both the continuous-time distributed

ESC algorithm from Chapter 3 and the discrete-time distributed ESC algorithm from Chapter 4 to the balloon formation control problem. Several simulation results showing the effectiveness of both controllers are presented. Most notably, simulations involving 1200 balloons and using real wind data show that discrete-time distributed ESC is an excellent technique for formation control of high-altitude balloons.

Chapter 6 is the conclusion. It contains a brief summary of the major results discussed in this thesis and provides an outlook on some useful future work that could extend these results.

The six appendices contain the mathematical proofs which are omitted from Chapter 3 and Chapter 4. Appendix A contains the proof of the main theorem from Chapter 3; Appendix F contains the proof of the main theorem from Chapter 4. Appendices B–E contain proofs of some smaller lemmas which are used in the proof found in Appendix F.

Chapter 2

Literature review

This chapter provides an introduction to the various topics covered in this thesis. Section 2.1 discusses the main ideas behind extremum-seeking control (ESC), its history, advantages, and current limitations. Section 2.2 gives an overview of multi-agent systems (MAS) and distributed control with an emphasis on consensus algorithms. Section 2.3 covers current techniques in formation control and coverage control.

2.1 Extremum-seeking control

Extremum-seeking control is a model-free RTO technique which relies only on measurements of a cost function without requiring knowledge of its mathematical description. ESC typically considers systems of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{2.1}$$

$$y = h(\boldsymbol{x}) \tag{2.2}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state, $\boldsymbol{u} \in \mathbb{R}^m$ is the input, $y \in \mathbb{R}$ is the output, $\boldsymbol{f}$ is a smooth vector field on $\mathbb{R}^n$, and h is a smooth convex cost function. The objective of ESC is to find $\boldsymbol{u}^*$ and $\boldsymbol{x}^*$ such that $\boldsymbol{f}(\boldsymbol{x}^*, \boldsymbol{u}^*) = \boldsymbol{0}$ and $h(\boldsymbol{x})$ is minimized by $\boldsymbol{x}^*$. This problem is the steady-state minimization of $h(\boldsymbol{x})$ subject to the dynamics given by $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$.

A similar problem of minimizing a static map—typically representing the steady-state cost function—is often also considered. By letting $\boldsymbol{x} = \boldsymbol{\pi}(\boldsymbol{u})$ be the steady-state of (2.1), the system can be described by

$$y = \ell(\boldsymbol{u}) = h(\boldsymbol{\pi}(\boldsymbol{u})). \tag{2.3}$$

The objective of ESC is to find $\boldsymbol{u}^*$ which minimizes $\ell = h \circ \boldsymbol{\pi}$.

Several variants of ESC have been developed which solve these problems. The common feature of all ESC algorithms is that the controller has no knowledge of $\boldsymbol{f}$, h, $\boldsymbol{\pi}$, or ℓ and must rely solely on measurements of y. A typical ESC (Figure 2.1) consists of a dither signal, parameter estimation, and a gradient descent. The dither signal provides persistence of excitation which is required by the parameter estimation algorithm to estimate $\frac{\partial \ell}{\partial \boldsymbol{u}}$, the gradient of the steady-state cost function with respect to the input. The ESC then follows a gradient descent based on this estimate to minimize the steady-state cost function.

The basic concept behind ESC was first developed in the 1920s by LeBlanc [73] to maximize the power transfer between a tram car and an electrical transmission line. Since this technique was not well understood and there was no guarantee of convergence, few researchers investigated ESC for several decades. Between 1940–1970 there was some research activity on ESC in Russia by researchers such as Kazakevich [62, 63, 64, 65], Morosanov [92], and Meerkov [88]. With the advent of adaptive control in the 1960s, ESC—itself a form of adaptive control—finally became a topic in the English-language control literature, with researchers such as Eykhoff [35, 36] and Blackman [14] investigating. Despite some advances, these techniques still lacked

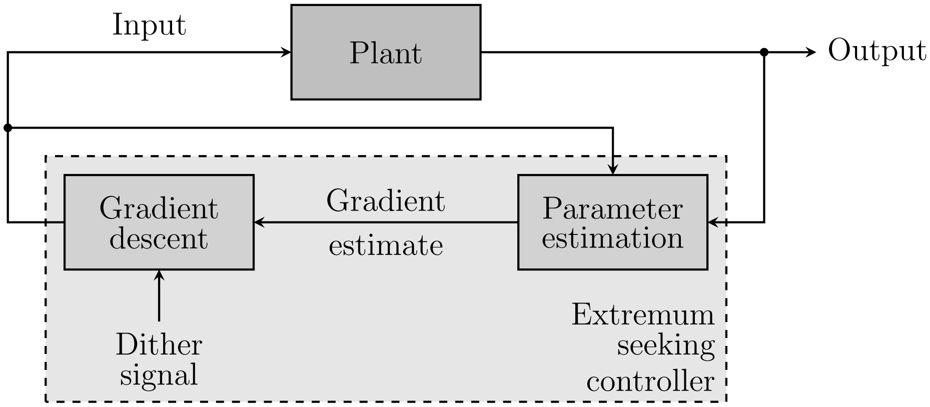


Figure 2.1: Block diagram of a general form of ESC. The controller minimizes the output by a gradient descent. The gradient is estimated by the parameter estimation. The dither signal provides persistence of excitation needed for parameter estimation.

a rigorous theoretical basis [117]. For more information on the history of extremum-seeking control, see the excellent review paper by Tan [119].

The early 2000s saw a renewed interest in ESC when Krstić and Wang [71] provided a rigorous proof of the technique's convergence. Krstić and Wang's original ESC (Figure 2.2) uses a series of high- and low-pass filters to correlate the plant output y with an input signal θ and estimate the optimal parameter $\widehat{\theta}$ which minimizes y. Their technique relies on a time-scale separation between the dynamics of the plant and the frequency of the dither signal $a\sin(\omega t)$, and they used averaging and singular perturbation analyses to prove the convergence of the system.

With ESC finally on a rigorous theoretical foundation, considerably more research was done on the topic in the early 2000s. Rotea extended the basic ESC scheme to multi-input systems by passing the input $\boldsymbol{u} \in \mathbb{R}^m$ through an appropriate dynamic compensator and was also able to show that ESC is still effective in the presence of a noisy output signal [108]. A basic discrete-time ESC (Figure 2.3) was designed by

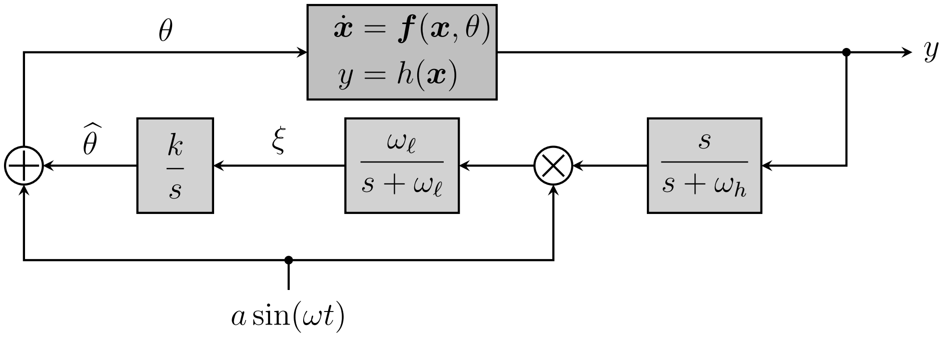


Figure 2.2: Block diagram of the form of ESC proposed by Krstić and Wang in [71]. The output signal is high-pass filtered, modulated by the dither signal, low-pass filtered, and integrated to obtain the input bias.

Choi *et al.* for the optimization of a cost function represented by a static map [22]; however, this technique is limited in application as it cannot be used for systems with unstable or slow dynamics. While the original ESC techniques were only shown to be effective at finding a time-invariant extremum, Ariyur and Krstić were able to extend these results to systems with time-varying extrema [4]. Tan *et al.* showed that ESC can achieve semi-global stability where the region of attraction can be made arbitrarily large by appropriate choice of tuning parameters [120].

Early ESC techniques were perturbation-based, estimating the relevant parameters using a combination of filters and a sinusoidal perturbation. More recently, estimation based ESC techniques have been developed which directly estimate the steady-state gradient. Initial work by Guay and Zhang used an adaptive control approach to guarantee convergence of parameter estimates resulting in convergence of the system to the unknown optimum [52]. This work was extended by DeHaan and Guay to solve constrained optimization problems by augmenting the cost function with a barrier function that prevents violation of the constraint [29]. Adetola and Guay investigated the minimal perturbation required for parameter convergence [1]

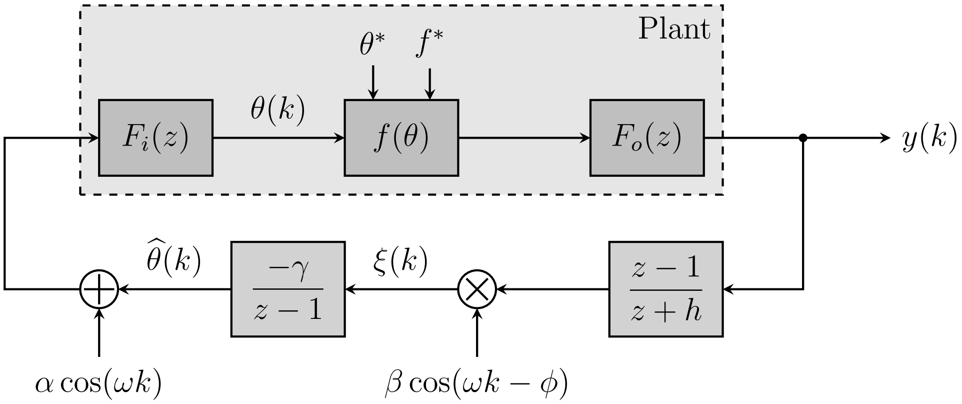


Figure 2.3: Block diagram of the discrete-time ESC proposed by Choi *et al.* in [22]. The linear input and output blocks, $F_i(z)$ and $F_o(z)$ are assumed stable. The ESC consists of a high-pass filter, modulation by a signal similar to the dither, and integration.

and finite-time parameter convergence [2]. Nešić *et al.* showed that ESC is possible using any arbitrary parameter estimation routine combined with any arbitrary optimization routine [97]. A particularly effective estimation technique developed by Moshksar and Guay requires only one tuning parameter and can track time-varying parameters [93]. While perturbation-based ESC is easier to tune, Guay and Burns found that estimation-based ESC routines tend to have better performance [46].

One of the main limitations of ESC is its slow performance due to the need for time-scale between the system dynamics and the controller. One method to improve performance is to estimate the Hessian of the cost function and use a Newton-based descent instead of a gradient-based descent. Newton-based ESC has been developed by Moase *et al.* [90] and Ghaffari *et al.* [41], but this technique still requires time-scale separation. Zhang and Ordóñez were able to remove the need for time-scale separation using a numerical optimization algorithm and a state regulator, however their technique requires a measurement of both the output and the state [142]. By

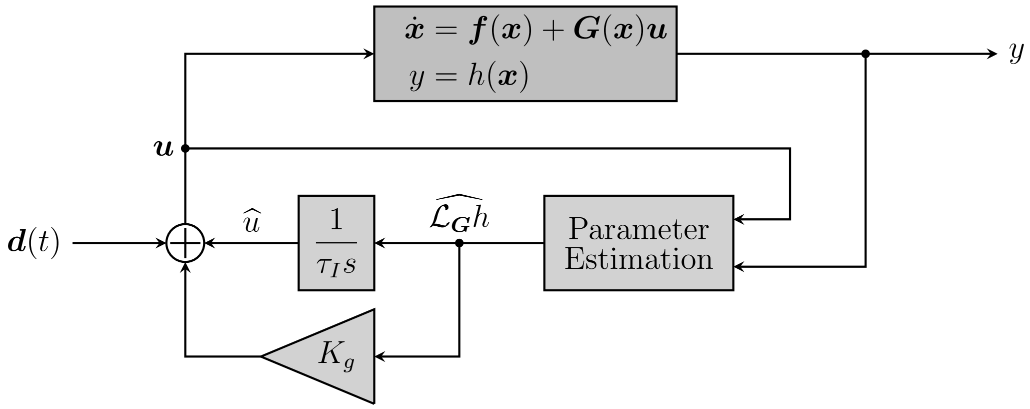

$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}$
$y = h(\boldsymbol{x})$
y
$\boldsymbol{u}$
$\boldsymbol{d}(t)$
$\widehat{u}$
$\frac{1}{\tau_I s}$
$\widehat{\mathcal{L}_{\boldsymbol{G}} h}$
Parameter Estimation
K_g

Figure 2.4: Block diagram of the basic form of the PI ESC algorithm developed by Guay and Dochain [48]. This technique is based on a gradient descent using $\widehat{\mathcal{L}_{\boldsymbol{G}} h}$, an estimate of the gradient of the steady-state cost function with respect to the input. The integral term provides the standard extremum-seeking action. The proportional term increases the speed of optimization, eliminating the need for time-scale separation.

using a proportional-integral (PI) extremum-seeking technique (Figure 2.4), Guay and Dochain were able to remove the need for time-scale separation in output-feedback ESC [48]. This technique has been extended to discrete-time systems by Guay [45].

Over the years, ESC has been applied to many diverse applications such as photovoltaic cells [17, 18, 77, 139], bioreactors [25, 49, 87, 124, 131, 143], heating and cooling systems [79, 82, 90, 133], active braking systems [121, 140, 141], wind turbines [26, 42, 60, 68, 101], fuel cells [13, 19, 27, 146], aerospace engineering [8, 9, 10, 21], and internal combustion engines [31, 67, 103, 104]. Outside of this thesis, ESC has not yet been applied to the formation control of high-altitude balloons.

2.2 Distributed control & consensus

Achieving overall objectives of large-scale dynamical systems is difficult due to the computational complexity associated with centralized approaches. Instead, a distributed approach can be used. A distributed or multi-agent system (MAS) consists of several controllers, known as agents, which each perform one small part of the overall system's task [86]. The agents are often spread out over a large geographic area. Individual agents communicate with each other over a network to share information. The overall distributed system (Figure 2.5) is quite complex, with many controllers measuring different outputs, controlling different inputs, and communicating with other agents.

The distributed system as seen by a single agent (Figure 2.6) is much simpler [86]. It measures only a small number of outputs and receives some information from a few other agents which it can use to coordinate its actions with the rest of the network of agents. The information sent over the network is limited; each agent is not aware of exactly what the other agents are doing. Using its measurements of the system and the information from the network, the agent manipulates a few control actions. The agent must also send information out to other nodes in the network, but it typically sends the same information to all other nodes.

One way to coordinate MASs is through a consensus algorithm [99]. The objective of a consensus algorithm is for agents to share information so that all agents eventually agree on the value of a certain variable. This consensus variable typically relates to the overall state of the system, providing each agent with an overall understanding of the state of the entire system despite not knowing any other agent's control action. For example, in distributed optimization, a consensus algorithm is used on the overall cost

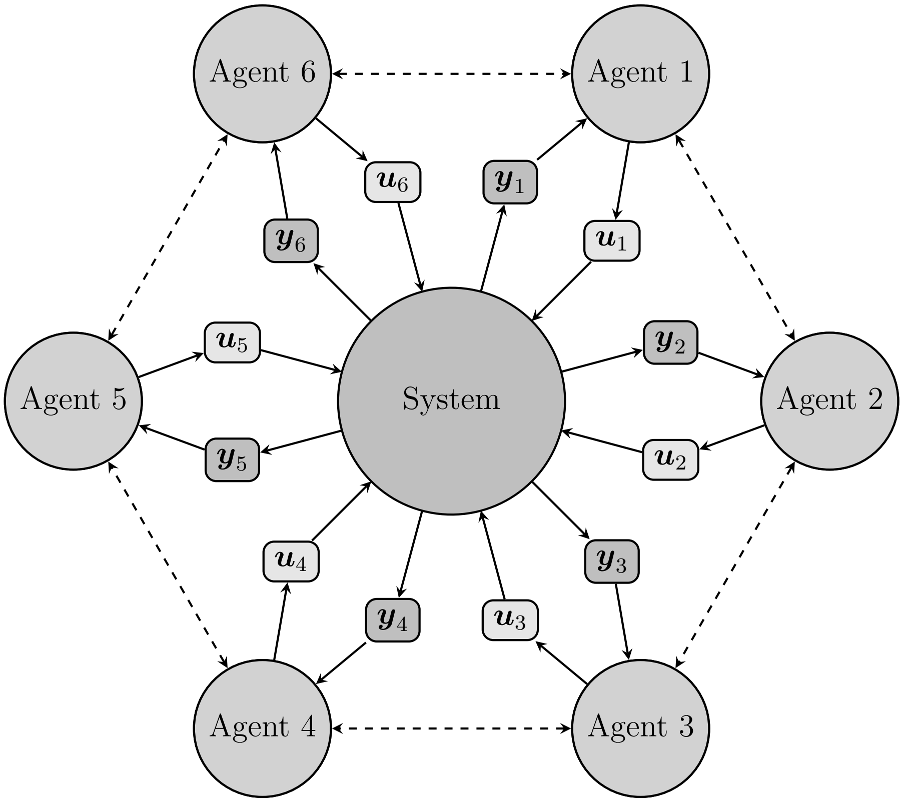


Figure 2.5: General scheme for a distributed control system. Agent i measures some output variables $\boldsymbol{y}_i$ and manipulates some input variables $\boldsymbol{u}_i$. Nearby agents communicate over the network to share information about the over all state of the system.

function [96]. While each agent does not directly measure the total cost—typically they can only measure one component of it—through consensus, all agents know the total cost and can cooperatively work to minimize it. In this way, the distributed system is able to solve a large optimization problem, and it has the advantages of parallel computing, such as decreased computing times, improved robustness, and scalability [118].

The effectiveness of a consensus algorithm depends on the network topology which can be described by a graph theory [15]. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a vertex set

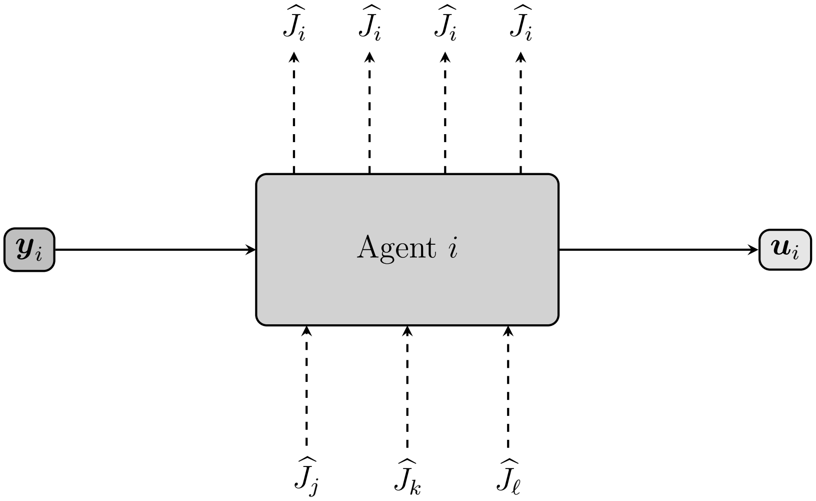


Figure 2.6: View of a distributed system as seen by one agent. This agent measures a few outputs, $\boldsymbol{y}_i$, and manipulates a few inputs, $\boldsymbol{u}_i$. It receives some information $\widehat{J}_j$, $\widehat{J}_k$, and $\widehat{J}_\ell$, which summarize the overall state of the system according to agents j, k, and ℓ. It transmits some information, $\widehat{J}_i$, which summarizes its own view on the overall state of the system.

$\mathcal{V}$ and an edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Each vertex $v_i \in \mathcal{V}$ represents an agent i in the MAS. An edge $e_{i,j} \in \mathcal{E}$ indicates that agent i can send information to agent j. A graph is said to be *undirected* if $e_{j,i} \in \mathcal{E}$ whenever $e_{i,j} \in \mathcal{E}$, and is otherwise said to be *directed* (Figure 2.7). For a *weighted graph*, we also assign a weight $w_{i,j} > 0$ to each edge $e_{i,j} \in \mathcal{E}$. In many cases we simply take $w_{i,j} = 1$ for every $e_{i,j} \in \mathcal{E}$ and refer to the graph as *unweighted.* A communication network can be described by a graph and the consensus algorithm relies on the values of $w_{i,j}$.

Graphs can be described by one of several matrices, allowing graphs to be analyzed using linear algebra. The weighted adjacency matrix $\boldsymbol{A}$ is a square matrix containing all of the weights of the graph. If $e_{i,j} \notin \mathcal{E}$, then $a_{i,j} = 0$ and otherwise $a_{i,j} = w_{i,j}$. For an unweighted graph, $a_{i,j} = 1$ if $e_{i,j} \in \mathcal{E}$ and $a_{i,j} = 0$ if $e_{i,j} \notin \mathcal{E}$. The degree matrix $\boldsymbol{D}$ is a square diagonal matrix where $d_{i,i} = \sum_{j=1}^{p} w_{i,j}$. For an unweighted

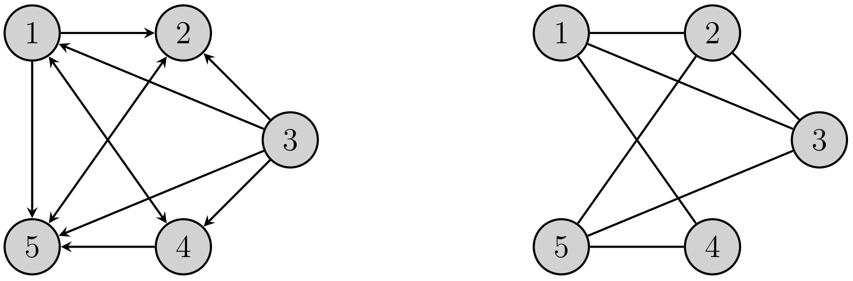


Figure 2.7: Examples of a directed graph (left) and an undirected graph (right). The directed graph has directed edges represented by single-headed arrows and undirected edges represented by double-headed arrows. In the undirected graph, all edges are undirected so the arrowheads are omitted.

graph, $d_{i,i} = \deg^+(v_i)$ is the number of agents which agent i sends information to. The Laplacian matrix is defined by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$. The Laplacians of the unweighted graphs in Figure 2.7 are

$$\boldsymbol{L}_{\text{directed}} = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & 0 & 0 & 2 & -1 \\ 0 & 1 & 0 & 0 & -1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{L}_{\text{undirected}} = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ -1 & -1 & 3 & 0 & -1 \\ -1 & 0 & 0 & 2 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{bmatrix}.$$

Note that $\boldsymbol{L}\mathbf{1} = \mathbf{0}$ since the diagonals of $\boldsymbol{D}$ are the row sums of $\boldsymbol{A}$. A graph is said to be *weight-balanced* if $\boldsymbol{L}^\top\mathbf{1} = \mathbf{0}$. Also note that for an undirected graph $\boldsymbol{L} = \boldsymbol{L}^\top$, while for an directed graph $\boldsymbol{L}$ is not necessarily symmetric.

One main requirement for consensus algorithms is that the graph is connected so that each agent receives information from every agent. An undirected graph is *connected* if there is a sequence of edges connecting every pair of vertices. The necessary notion of connectivity for directed graphs is strong connectivity. A directed graph is *strongly connected* if for any pair of nodes (i, j) there exists a directed path from i to j and a directed path from j to i. Intuitively, connectivity matters because it means that each agent can get information from all the other agents, although this information may have passed through many other agents first.

Several researchers have considered the convergence properties of various consensus algorithms. The simplest consensus algorithms are of the form $\dot{\boldsymbol{x}} = -\boldsymbol{L}\boldsymbol{x}$ or $\boldsymbol{x}[k+1] = (\boldsymbol{I} - \boldsymbol{L}\Delta t)\boldsymbol{x}[k]$. Olfati-Saber and Murray showed that these first-order linear consensus algorithms result in all the states converging to the same value whenever the network is connected and that the convergence value is the average of the initial states whenever $\boldsymbol{L}$ is weight-balanced [100]. Nonlinear consensus algorithms can also achieve convergence, as Moreau showed, whenever the graph is connected and the update law is a convex combination of states at the previous time step [91]. When nonlinear consensus algorithms are used, the states do not necessarily converge to the average of the initial states. First-order consensus algorithms have been applied to vehicle formations, attitude alignment, rendezvous problem, coordinated decision making, flocking, coupled oscillators, and robot position synchronization [106]. The effect of communication delays has been considered by Xiao and Wang [138], Tian *et al.* [123], and Lin *et al.* [83]. Li and Zhang used a distributed stochastic approximation protocol to limit the effect of noise in communication channels and provide necessary and sufficient conditions for minimizing the mean square error and providing almost sure convergence when data are corrupted by stochastic noise [80].

Dynamic average consensus algorithms resemble the aforementioned static average consensus algorithms, except instead of averaging several initial values, they attempt to track the time-varying average of several time-varying reference signals. Freeman *et al.* considered first- and second-order linear consensus algorithms in continuous-time and were able to show that for weight-balanced connected digraphs, the PI algorithm achieves dynamic consensus with slowly time-varying input signals [38]. A class of discrete-time dynamic average consensus algorithms were designed by Zhu and Martínez using n^{th}-order algorithms for averaging signals with bounded n^{th} differences [147]. Kia *et al.* analyzed n^{th}-order consensus algorithms in continuous- and discrete-time and bounded the convergence rate and tracking error in terms of the eigenvalues of $\boldsymbol{L}$ and the bound for the n^{th} difference or derivative [66].

2.3 Formation & coverage control

Formation control and coverage control are two control objectives for MASs consisting of spatially distributed, mobile agents such as satellites, wheeled vehicles, aircraft, and robots. Formation control is primarily concerned with maintaining the positions and orientations of individual agents relative to other agents [28]. In formation control, the entire group of agents may move within its environment, and the control objective is simply to maintain a formation. Coverage control, on the other hand, is primarily concerned with deploying a group of sensor-equipped agents so that their entire environment can be adequately monitored by sensors [56]. In coverage control, the relative positions of the agents do not matter as long as their formation solves the coverage problem. While formation and coverage control have different objectives, the resulting behaviour is often complementary, with a fixed formation providing good coverage [5].

In formation control, the overall system objective is to maintain the relative positions and orientations of all agents. Many formation control strategies also include practical considerations such as collision avoidance, communication constraints, and obstacle avoidance. One approach to formation control is a behavioural approach where each robot implements a control that satisfies a weighted average of control objectives such as maintaining formation, path following, and collision avoidance [5]. Another approach is to designate one agent as the leader and have all agents maintain their position and orientation relative to this leader while the leader follows a path, coordinating the entire group [23, 113, 122]. Alternatively, a virtual leader can be used to ensure that the entire formation follows the path while maintaining formation [34, 76]. The use of artificial potential functions defined based on the positions of a virtual leader and obstacles can be used to maintain formation while avoiding obstacles and is implemented by each agent following a gradient descent based on its potential [75].

In coverage control, several vehicles occupy a space Q and try to spread themselves out to maximize the coverage of this region. Given a probability density $\phi : \mathsf{Q} \to \mathbb{R}_{\geq 0}$, the objective of the MAS is to position agents at $q_1, \ldots, q_p$ to minimize

$$H = \int_{\mathsf{Q}} \min_{q_i}\{\|q - q_i\|\} \mathrm{d}\phi(q). \tag{2.4}$$

When H is minimized, the agents are evenly spread out so that they maximize the possibility of observing an event whose location has probability density ϕ. Since the integrand at q depends only on the position of the agent that minimizes $\|q - q_i\|$, coverage control is closely linked to Voronoi partitions.

A Voronoi partition is a partition of a space into several regions $\Gamma_1, \ldots, \Gamma_p$ based on a set of points $q_1, \ldots, q_n$ with the property that Γ_i is the set of points which are closest to q_i [129]. Formally, a Voronoi cell is defined by

$$\Gamma_i = \{q \in \mathsf{Q} \mid \|q - q_i\| < \|q - q_j\| \, \forall j \neq i\} \,. \tag{2.5}$$

Using the definition of the a Voronoi cell, the coverage control objective function, (2.4), can alternatively be written as

$$H = \sum_{i=1}^{p} \int_{\Gamma_i} \|q - q_i\| \, \mathrm{d}\phi(q). \tag{2.6}$$

This cost function is (locally) minimized when each q_i is at the ϕ-centroid of Γ_i resulting in a centroidal Voronoi formation [33].

A common way to solve coverage control problems is Lloyd's algorithm, where each agent moves towards the centroid of its Voronoi cell. This approach is based on Lloyd's optimal quantization schemes which can be interpreted as a gradient descent [85]. Since Voronoi cells and centroids can be computed from neighbouring agents's position, Lloyd's algorithm can be implemented by a distributed MAS with communication between neighbours [24]. When ϕ is multimodal or there or several local minima to H, a "ladybug"-like variant of Lloyd's algorithm where agents spiral towards their Voronoi centroids can result in a better configuration [111]. If the orientation of the agents affect their sensing ability, a different metric can be used in place of the standard ℓ_2-norm, resulting in an anisotropic Voronoi partition and a cost function which is minimized by an anisotropic centroidal Voronoi formation [53]. Computation times for the Voronoi centroid in Lloyd's algorithm can be improved by using a polynomial approximation of ϕ with an adaptive law to improve the estimate of ϕ [112]. Lloyd's algorithm can also be adapted to non-convex environments by combining it with a path planning algorithm that avoids obstacles [16].

Several other techniques exist which can solve coverage control problems. While these techniques are different from Lloyd's algorithm, and do not rely on Voronoi partitions, they are still based on a gradient descent. One method, when ϕ is constant, is to use a gradient descent of a potential function based on each agent's distance from other agents and the boundary [55]. Another formulation is based on a gradient descent of a cost function $H = \int_{\mathsf{Q}} \phi(q)P(q, q_1, \ldots, q_n)\mathrm{d}q$ where P is the probability that an event at q can be sensed by some sensor $q_1, \ldots, q_n$ [57, 81]. This approach does not assume that an event can only be sensed by the nearest agent, and can be seen as a generalization of Lloyd's algorithm. It reverts to Lloyd's algorithm and can be expressed in terms of a Voronoi partition when it is assumed that the nearest agent always senses an event. This joint probability approach can be used in a non-convex environment by having $P(q, q_1, \ldots, q_n)$ be defined in terms of sensing cones which get blocked by obstacles [145].

Both formation control and coverage control rely on various geometric notions, such as distances, angles, areas, centroids, and Voronoi partitions. While the majority of the results are derived for Euclidean space, the extension to non-Euclidean spaces is often straight-forward by using differential geometry. Delaunay triangulations and Voronoi partitions—which are of particular importance for coverage control—can be defined on arbitrary Riemannian manifolds with uniqueness guaranteed if there set of points is sufficiently dense [74]. Of Riemannian manifolds, the sphere is particularly important, as an MAS surrounding Earth naturally moves on the sphere. Bishop and Basiri have used differential geometric definitions of angles and distances to achieve formation control on the sphere [12].

Chapter 3

Continuous-time distributed extremum-seeking control

In this chapter, a continuous-time version of distributed extremum-seeking control is developed. The most general problem which can be solved by this technique and some necessary assumptions is described in Section 3.2. The distributed ESC is designed and its convergences is proven in Section 3.3. Several simulation examples for large-scale, unstable, and nonlinear systems are shown in Section 3.4.

3.1 Introduction

Current approaches in extremum-seeking control assume that a single controller is used to control the entire system. In many applications this approach is not practical. If the system is very large, a single centralized controller may not be able to perform the necessary calculations fast enough to implement the needed control action. If a system consists of several subsystems which are distributed over a large distance, it is more practical to have each subsystem controlled by a nearby local controller than to use a single controller which is distant from most subsystems. Since current ESC techniques are restricted to a centralized approach, they cannot be applied to potential control applications such as transportation networks, the electricity grid, social media, large manufacturing plants, and distributed robotics.

Multi-agent variants of ESC have already been proposed by several researchers [39, 40, 43, 69, 102, 116]. In these approaches, each agent measures its own local cost or payoff function which it optimizes by extremum-seeking. However, since all the costs are dependent on each other, this approach is competitive, and the overall system can only reach a Nash equilibrium, and cannot achieve overall system objectives. Decentralized ESC techniques which do reach the overall system optimum require carefully designed cost local cost functions such that the total cost is minimized when all the local costs are minimized [78, 105]. Nedić *et al.* showed that a distributed system can achieve overall system objectives by solving local optimization problems and communicating estimates of the overall optimum via a network [96]. A general framework for distributed ESC based on communicating total cost estimates was provided by Kvaternik *et al.* but was not applied with any ESC algorithm [72].

In this chapter, we develop a distributed version of extemum-seeking control in continuous-time. This approach involves a consensus algorithm over a network of agents which each implement a local ESC. It is assumed that each agent measures a local cost using sensor measurements, but does not have a mathematical model describing the cost function. The overall system objective is to measure the sum of the local costs. Since each agent does not actually measure the objective function to be minimized, a consensus algorithm is used to provide each agent of the overall cost. Then the agents use a gradient-based ESC algorithm to minimize the overall cost based on the estimate obtained from the network. A novel proportional-integral ESC technique originally proposed by Guay and Dochain [48] is used to remove the need for time-scale separation, and allow the distributed ESC to stabilize slowly unstable systems. The proposed continuous-time distributed PI ESC technique is valid for nonlinear systems with unknown, unstable dynamics with coupling between agents.

3.2 Problem description

Consider a network of nonlinear control-affine systems of the form

$$\dot{\boldsymbol{x}}_i = \boldsymbol{f}_i(\boldsymbol{x}, \boldsymbol{z}) + \boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{z})\boldsymbol{u} \tag{3.1}$$

$$y_i = h_i(\boldsymbol{x}) \tag{3.2}$$

where $\boldsymbol{x} = [\boldsymbol{x}_1^\top \quad \dots \quad \boldsymbol{x}_p^\top]^\top \in \mathbb{X} \subseteq \mathbb{R}^n$ is the state vector, $\boldsymbol{u} = [\boldsymbol{u}_1^\top \quad \dots \quad \boldsymbol{u}_p^\top] \in \mathbb{U} \subseteq \mathbb{R}^m$ is the input vector for the entire network, and $\boldsymbol{z} \in \mathbb{R}^{n_z}$ is a vector of states associated with the zero dynamics. Each $\boldsymbol{x}_i$ has dimension n_i with $\sum_{i=1}^p n_i = n$. The dynamics of each agent i are described by the dynamics (3.1) with local cost (3.2). Each agent manipulates the local input variables $\boldsymbol{u}_i \in \mathbb{R}^{m_i}$, with $\sum_{i=1}^p m_i = m$. The vector fields $\boldsymbol{f}_i(\boldsymbol{x}, \boldsymbol{z}) \in \mathbb{R}^{n_i}$ and $\boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{z}) \in \mathbb{R}^{n_i \times m}$ are unknown smooth functions of $\boldsymbol{x}$ and $\boldsymbol{z}$ and the cost functions $h_i(\boldsymbol{x})$ are unknown smooth functions of $\boldsymbol{x}$.

The overall network cost function is the sum of all the individual costs. Using the definition of the local costs (3.2), the total cost is

$$J = H(\boldsymbol{x}) = \sum_{i=1}^{p} h_i(\boldsymbol{x}). \tag{3.3}$$

The objective is to steer the system to the equilibrium $\boldsymbol{x}^* \in \mathbb{X}$ and $\boldsymbol{u}^* \in \mathbb{U}$ that achieves the minimum value of $J(\boldsymbol{x})$ using only measurements of the local cost and communication between agents.

By concatenating the agent dynamics from (3.1), we can write the overall dynamics of the network as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{z}) + \boldsymbol{G}(\boldsymbol{x}, \boldsymbol{z})\boldsymbol{u} \tag{3.4}$$

$$\dot{\boldsymbol{z}} = \boldsymbol{f}_z(\boldsymbol{x}, \boldsymbol{z}). \tag{3.5}$$

The drift vector field $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{z}) = [\boldsymbol{f}_1^\top(\boldsymbol{x}, \boldsymbol{z}) \;\; \dots \;\; \boldsymbol{f}_p^\top(\boldsymbol{x}, \boldsymbol{z})]^\top$, (matrix-valued) control vector field $\boldsymbol{G}(\boldsymbol{x}, \boldsymbol{z}) = [\boldsymbol{g}_1^\top(\boldsymbol{x}, \boldsymbol{z}) \;\; \dots \;\; \boldsymbol{g}_p^\top(\boldsymbol{x}, \boldsymbol{z})]^\top$, and zero dynamics vector field $\boldsymbol{f}_z(\boldsymbol{x}, \boldsymbol{z}) : \mathbb{R}^n \times \mathbb{R}^{n_z} \to \mathbb{R}^{n_z}$ are all smooth.

The overall system objective is to minimize J at steady-state. The overall system equilibrium occurs when a triple $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u})$ simultaneously satisfies the equations

$$\boldsymbol{0} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{z}) + \boldsymbol{G}(\boldsymbol{x}, \boldsymbol{z})\boldsymbol{u} \tag{3.6}$$

$$\boldsymbol{0} = \boldsymbol{f}_z(\boldsymbol{x}, \boldsymbol{z}). \tag{3.7}$$

We assume that for each $\boldsymbol{u} \in \mathbb{U}$, there exists a unique pair $(\boldsymbol{x}, \boldsymbol{z}) \in \mathbb{X} \times \mathbb{R}^{n_z}$ which satisfies (3.6) and (3.7). Thus we can define maps $\boldsymbol{\pi}_x : \mathbb{U} \to \mathbb{X}$ and $\boldsymbol{\pi}_z : \mathbb{U} \to \mathbb{R}^{n_z}$ such that for a given $\boldsymbol{u}$, $(\pi_x(\boldsymbol{u}), \pi_z(\boldsymbol{u}), \boldsymbol{u})$ is the unique triple that satisfies (3.6) and (3.7). Using these steady-state maps, we can define a steady-state cost function by

$$J = H(\boldsymbol{\pi}_x(\boldsymbol{u})) = \ell(\boldsymbol{u}) \tag{3.8}$$

where $\ell = H \circ \boldsymbol{\pi}_x$. At equilibrium, the system objective is reduced to finding the minimizer $\boldsymbol{u}^*$ of $\ell(\boldsymbol{u})$. Furthermore, for stabilization we must ensure that $\boldsymbol{x}(t)$ and $\boldsymbol{z}(t)$ converge to $\boldsymbol{x}^* = \boldsymbol{\pi}_x(\boldsymbol{u})$ and $\boldsymbol{z}^* = \boldsymbol{\pi}_z(\boldsymbol{u})$.

For the combined stabilization and optimization problem to be well posed, we must make assumptions about the cost function and dynamics. The main assumption about the cost functions is that the total steady-state cost—but not the local costs—is locally strictly convex. For stabilization to be possible, we must assume that the $\boldsymbol{x}$-dynamics are controllable and that the $\boldsymbol{z}$-dynamics are stable.

Assumption 3.1 (Convexity)**.** There exists some positive constant $\beta_1 \in \mathbb{R}_{>0}$ such that for all $\widehat{\boldsymbol{u}} \in \mathbb{U}$, the equilibrium steady-state map $\ell(\boldsymbol{u})$ is such that

$$\frac{\partial \ell}{\partial \boldsymbol{u}}\bigg|_{\widehat{\boldsymbol{u}}} (\widehat{\boldsymbol{u}} - \boldsymbol{u}^*) \geq \beta_1 \left\| \widehat{\boldsymbol{u}} - \boldsymbol{u}^* \right\|^2 . \tag{3.9}$$

Assumption 3.2 (Stabilizability)**.** There exists a positive definite function $\Omega(\boldsymbol{x}, \boldsymbol{z}) = V_z(\boldsymbol{z}) + J(\boldsymbol{x})$ and positive constants $\beta_{2,x}, \beta_{2,z}, \beta_{3,x}, \beta_{3,z} \in \mathbb{R}_{>0}$ such that

$$\Omega(\boldsymbol{x}, \boldsymbol{z}) - \Omega(\boldsymbol{\pi}_x(\widehat{\boldsymbol{u}}), \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}})) \geq \beta_{2,x} \left\| \boldsymbol{x} - \boldsymbol{\pi}_x(\widehat{\boldsymbol{u}}) \right\|^2 + \beta_{2,z} \left\| \boldsymbol{z} - \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}}) \right\|^2 \tag{3.10}$$

$$\Omega(\boldsymbol{x}, \boldsymbol{z}) - \Omega(\boldsymbol{\pi}_x(\widehat{\boldsymbol{u}}), \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}})) \leq \beta_{3,x} \left\| \boldsymbol{x} - \boldsymbol{\pi}_x(\widehat{\boldsymbol{u}}) \right\|^2 + \beta_{3,z} \left\| \boldsymbol{z} - \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}}) \right\|^2 . \tag{3.11}$$

Furthermore, there exists $K_g^*, \beta_{4,x}, \beta_{4,z} \in \mathbb{R}_{>0}$ such that $\Omega(\boldsymbol{x}, \boldsymbol{z})$ satisfies

$$\begin{aligned} \frac{\partial V_z}{\partial \boldsymbol{z}} \boldsymbol{f}_z + \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{f} + \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{G} \widehat{\boldsymbol{u}} - K_g^* \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{G} \boldsymbol{G}^\top \left(\frac{\partial J}{\partial \boldsymbol{x}} \right)^\top & \\ \leq -\beta_{4,x} \left\| \boldsymbol{x} - \boldsymbol{\pi}_x(\widehat{\boldsymbol{u}}) \right\|^2 - \beta_{4,z} \left\| \boldsymbol{z} - \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}}) \right\|^2 . & \end{aligned} \tag{3.12}$$

for all $(\boldsymbol{x}, \boldsymbol{z}) \in \mathbb{X}$, and for all $\widehat{\boldsymbol{u}} \in \mathbb{U}$. This assumption determines a class of minimum-phase unstable nonlinear system that can be optimized and stabilized by ESC. These systems must be stabilizable by the state-feedback $\boldsymbol{u} = -K_g^* \boldsymbol{G}^\top(\boldsymbol{x}, \boldsymbol{z}) \frac{\partial J^\top}{\partial \boldsymbol{x}} + \widehat{\boldsymbol{u}}$, for some $K_g^* > 0$. If the system is open-loop stable then $K_g^* = 0$ meets the assumption.

3.3 Distributed extremum-seeking controller

In this section, we design the distributed ESC that is used to minimize (3.3) while stabilizing (3.4). Each agent uses only local cost measurements and communication with neighbouring agents to compute its control action. The overall system is coordinated by a dynamic average consensus algorithm. This consensus takes the local costs y_i as inputs and returns estimates $\widehat{J}_i$ of the average cost as outputs. The average cost $\frac{1}{p}\sum_{i=1}^{p} h_i = \frac{1}{p}J$ is proportional to the total cost which is the objective function to be minimized. Each agent uses a local ESC to minimize their average cost estimates. Since the average cost estimates converge, the minimization of the average cost estimates results in the minimization of the total cost. In this way, the local ESCs are coordinated and the distributed ESC can achieve the overall system objective despite no agent having full knowledge about the entire system.

3.3.1 Consensus algorithm

The consensus algorithm provides each agent with an estimate of the average cost. Most existing average consensus algorithms are only valid for static consensus problems (see Section 2.2). When the inputs to the consensus algorithm—in this case, the local costs—are time-varying, a dynamic average consensus protocol is needed.

We use the PI consensus algorithm developed by Freeman *et al.* [38]. Let $\widehat{J}_i$ denote agent i's estimate of $\frac{1}{p}J$ and $\widehat{\boldsymbol{J}} = [\widehat{J}_1 \quad \ldots \quad \widehat{J}_p]^\top$ and let $\boldsymbol{\rho} = [\rho_1 \quad \ldots \quad \rho_p]^\top$ be the integrator state. The dynamic average consensus algorithm is given by

$$\begin{bmatrix} \dot{\widehat{\boldsymbol{J}}} \\ \dot{\boldsymbol{\rho}} \end{bmatrix} = \begin{bmatrix} -\kappa_0 \boldsymbol{I} - \kappa_P \boldsymbol{L} & \kappa_I \boldsymbol{L} \\ -\kappa_I \boldsymbol{L} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{J}} \\ \boldsymbol{\rho} \end{bmatrix} + \begin{bmatrix} \kappa_0 \boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{y} \tag{3.13}$$

where $\kappa_0, \kappa_P, \kappa_I \in \mathbb{R}_{>0}$ are user-specified tuning parameters. The parameter κ_0 controls how much weight each agent gives its own local cost or how selfish each agent is. The parameters κ_P and κ_I are the PI control gains which determine how fast the agents coordinate their actions. If $\kappa_P \gg \kappa_I$, the consensus tracks the average faster but is more susceptible to noise. If $\kappa_I \gg \kappa_P$ the consensus does a better job at rejecting noise but tracks the average slower.

Convergence of this algorithm depends on the properties of the graph. In Theorems 5 and 6 of the paper by Freeman *et. al* [38], several convergence requirements for (3.13) are provided. For time-invariant graphs, a sufficient condition is that the communication occurs over a connected weight-balanced digraph. A connected graph ensures that all information can propagate to all areas of the network and thus all agents' estimates converge to the same value. The weight-balanced property ensures that all information gets used equally and so that the estimates converge to the average instead of some other value. This consensus algorithm is limited to cases where $\boldsymbol{y}$ and $\dot{\boldsymbol{y}}$ are bounded. It achieves dynamic tracking and has zero high-frequency gain, allowing it to filter out noise in $\boldsymbol{y}$. This algorithm was selected over other average consensus algorithms as it can achieve dynamic average tracking and its requirements on the graph structure are easy to achieve.

Assumption 3.3. The communication network for local costs is represented by a time-invariant connected weight-balanced digraph.

The consensus algorithm in (3.13) is written in a vector format that relates all of the agents' total cost estimates in one equation. In reality, the total costs are not computed in a vector format, but the computation is instead distributed over the network. Each agent computes its own local cost, so agent i only computes the rows of (3.13) corresponding to $\widehat{J}_i$ and ρ_i. These rows have dynamics

$$\dot{\widehat{J}}_i = \kappa_0(y_i - \widehat{J}_i) - \kappa_P \boldsymbol{\ell}_{i,:}\widehat{\boldsymbol{J}} + \kappa_I \boldsymbol{\ell}_{i,:}\boldsymbol{\rho} \tag{3.14}$$

$$\dot{\rho}_i = -\kappa_I \boldsymbol{\ell}_{i,:}\widehat{\boldsymbol{J}} \tag{3.15}$$

where $\boldsymbol{\ell}_{i,:}$ is the i^{th} row of $\boldsymbol{L}$. Since agent i does not have access to all information, it is important to check that this algorithm can actually be implemented in a distributed way. Therefore we must check that each agent can actually compute its total cost update law.

At first, it seems that agent i cannot compute (3.14) or (3.15) since they both depend on all agents' $\widehat{J}_i$ and ρ_i but agent i only knows the value of these variables for its neighbours. However, since the Laplacian encodes the neighbour structure of the graph, whenever agents i and j are not neighbours, $\ell_{i,j} = 0$. Therefore (3.14) and (3.15) can be expanded as

$$\dot{\widehat{J}}_i = \kappa_0(y_i - \widehat{J}_i) - \sum_{j \in \mathcal{N}_i} \kappa_P \ell_{i,j} \widehat{J}_j + \sum_{j \in \mathcal{N}_i} \kappa_I \ell_{i,j} \rho_j \tag{3.16}$$

$$\dot{\rho}_i = -\sum_{j \in \mathcal{N}_i} \kappa_I \ell_{i,j} \widehat{J}_j \tag{3.17}$$

where $\mathcal{N}_i = \{j \in \{1, \ldots, p\} \mid e_{i,j} \in \mathcal{E}\}$ is the neighbour set of agent i on the communication graph. Therefore agent only requires information that it does have access to and (3.13) is a valid distributed algorithm.

3.3.2 Input-output dynamics of agent i

Estimation-based ESC is based on the idea of parameterizing the total cost dynamics by a locally linear model, estimating the gradient of the cost function based on this local model, and then optimizing the cost by a gradient descent. Therefore, ESC relies on a local input-output model. We consider the dynamics of the total cost J by differentiating (3.3) along the vector fields given by (3.4), resulting in

$$\dot{J}(t) = \mathcal{L}_{\boldsymbol{f}} H + \mathcal{L}_{\boldsymbol{G}} H \boldsymbol{u} \tag{3.18}$$

where $\mathcal{L}_{\boldsymbol{f}} H$ and $\mathcal{L}_{\boldsymbol{G}} H$ are Lie derivates of H along $\boldsymbol{f}$ and $\boldsymbol{G}$.

The input-output dynamics of (3.18) relate all agents' inputs to the total cost. Since each agent only controls its own inputs, it is useful to rewrite the input-output dynamics in a form that explicitly state the dependence on a specific agent's input. By expanding the definition of the Lie derivative $\mathcal{L}_{\boldsymbol{G}} H$ in (3.18), the dynamics of J as seen by each agent i are given by:

$$\dot{J} = \mathcal{L}_{\boldsymbol{f}} H + \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{G}_i \boldsymbol{u}_i + \sum_{j \neq i} \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{G}_j \boldsymbol{u}_j \tag{3.19}$$

where $\boldsymbol{G}_i \in \mathbb{R}^n$ is i^{th} column of the matrix-valued function $\boldsymbol{G}(\boldsymbol{x})$.

Each agent must estimate some parameters based on (3.19). Agent i has access to $\boldsymbol{u}_i$ and, through the consensus algorithm, a convergent estimate of $\frac{J}{p}$. Therefore, we parameterize the average cost dynamics as

$$\frac{\dot{J}}{p} = \theta_{0,i}(t) + \boldsymbol{\theta}_{1,i}^{\top}(t) \boldsymbol{u}_i. \tag{3.20}$$

In this form, an agent can use data of $\widehat{J}_i$ and $\boldsymbol{u}_i$ to estimate some linear parameters $\theta_{0,i}(t)$ and $\boldsymbol{\theta}_{1,i}(t)$. By inspecting (3.19) and noting that it differs from (3.20) by a factor of $\frac{1}{p}$, the parameters can be exactly expressed as:

$$\theta_{0,i}(t) = \frac{1}{p}\left(\mathcal{L}_{\boldsymbol{f}} H + \sum_{j \neq i} \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{G}_j \boldsymbol{u}_j\right) \tag{3.21}$$

$$\boldsymbol{\theta}_{1,i}(t) = \frac{1}{p}\left(\frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{G}_i\right)^{\top}. \tag{3.22}$$

The parameter $\theta_{0,i}(t)$ describes the effect of the system drift and the other agents' control on the total cost dynamics. The parameter $\boldsymbol{\theta}_{1,i}(t)$ is the gradient of the total cost with respect to agent i's control inputs. Note that both parameters are time-varying even though the vector fields are not, as the model is locally linear while the system is nonlinear.

3.3.3 Time-varying parameter estimation

The locally linear input-output model in (3.20) is useful for minimizing J by a gradient descent. Each ESC works by implementing a gradient descent using the parameter $\boldsymbol{\theta}_{1,i}(t)$. Since the agents only know $J(t)$ and $\boldsymbol{u}_i(t)$, they cannot directly measure $\boldsymbol{\theta}_{1,i}(t)$ and instead estimate it. While ESC only requires the estimate $\widehat{\boldsymbol{\theta}}_{1,i}$, for this estimate to converge, $\widehat{\theta}_{0,i}$ must also be estimated.

Each agent implements a parameter estimation routine similar to the one presented by Adetola and Guay [2]. The parameter estimation is based on (3.20) with the average cost replaced by an estimate of it. Let $\boldsymbol{\phi}_i = [1 \;\; \boldsymbol{u}_i^{\top}]^{\top}$ be the regressor vector and $\boldsymbol{\theta}_i = [\theta_{0,i} \;\; \boldsymbol{\theta}_{1,i}^{\top}]^{\top}$ be the true parameter vector. Then the estimation is based on the equation

$$\dot{\widehat{J}}_i = \boldsymbol{\phi}_i^T \boldsymbol{\theta}_i + \dot{\nu}_i \tag{3.23}$$

where $\nu_i = \widehat{J}_i - \frac{1}{p} J$ is the consensus error which is assumed to be small and exponentially decaying. In standard least squares estimation, we would estimate $\boldsymbol{\theta}_i$ by computing a covariance matrix $\boldsymbol{\Sigma}_i$ from measurements of $\boldsymbol{\phi}_i$ and then compute the parameter estimate $\widehat{\boldsymbol{\theta}}_i$ from this covariance matrix and measurements of $\dot{\widehat{J}}_i$. The resulting parameter estimate would minimize a function relating an estimator of $\dot{\widehat{J}}_i$ to the actual measured $\dot{\widehat{J}}_i$.

The parameter estimation algorithm we use is a continuous-time algorithm which regresses on (3.23) and uses a covariance matrix and an estimator. The estimator $\widehat{y}_i$ is predicts the total cost using current parameter estimates. Its dynamics are

$$\dot{\widehat{y}}_i = \boldsymbol{\phi}_i^\top \widehat{\boldsymbol{\theta}}_i + K e_i + \boldsymbol{w}_i^\top \dot{\widehat{\boldsymbol{\theta}}}_i \tag{3.24}$$

where $K \in \mathbb{R}_{\geq 0}$ is the estimation gain and $e_i = \widehat{J}_i - \widehat{y}_i$ is the error between the total cost estimate and the model's prediction of it. The $\boldsymbol{\phi}_i^\top \widehat{\boldsymbol{\theta}}_i$ term in (3.24) resembles (3.23) with $\widehat{\boldsymbol{\theta}}_i$ replacing $\boldsymbol{\theta}_i$. This term results in $\widehat{y}_i$ and $\widehat{J}_i$ having similar dynamics when the parameter estimate is close to the true parameter. The term Ke_i in (3.24) is reduces the error by moving $\widehat{y}_i$ closer to $\widehat{J}_i$ when e_i is large. The last term in (3.24) uses derivative information to compensate for time-varying parameters.

The last term in (3.24) also involves $\boldsymbol{w}_i$ which is a filtered version of $\boldsymbol{\phi}_i$. Unwanted noise in the regressor vector is removed by a simple low pass filter

$$\dot{\boldsymbol{w}}_i = -K\boldsymbol{w}_i + \boldsymbol{\phi}_i. \tag{3.25}$$

In addition to its use in (3.24), $\boldsymbol{w}_i$ is used in defining the dynamics of the covariance matrix and parameter update law.

The parameter update law also relies on an auxiliary variable $\eta_i = e_i - \boldsymbol{w}_i^\top \widetilde{\boldsymbol{\theta}}_i$ where $\widetilde{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i - \widehat{\boldsymbol{\theta}}_i$ is the parameter estimation error. This auxiliary variable has relatively simple dynamics and depends on the parameter estimation error, which is a quantity of interest. Using (3.23), (3.24), and (3.25), its dynamics are

$$\dot{\eta}_i = -K\eta_i - \boldsymbol{w}_i^\top \dot{\boldsymbol{\theta}}_i + \dot{\nu}_i. \tag{3.26}$$

Assuming that the consensus converges and the true parameter values change slowly, η_i has stable first-order linear dynamics with a known gain K. Therefore, we can easily estimate η_i using an estimator of the form

$$\dot{\widehat{\eta}}_i = -K\widehat{\eta}_i. \tag{3.27}$$

Since $\widehat{\eta}_i$ has very similar dynamics to η_i, we can use it as an estimator of η_i and since η_i contains information about $\widetilde{\boldsymbol{\theta}}_i$, $\widehat{\eta}_i$ provides some information about $\widetilde{\boldsymbol{\theta}}_i$. Note that since (3.26) depends on $\dot{\boldsymbol{\theta}}_i$ and $\dot{\nu}_i$ but (3.27) does not, $\widehat{\eta}_i$ is only a good estimate of η_i when $\boldsymbol{\theta}_i$ changes slowly and the consensus error is small.

The covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{(m_i+1)\times(m_i+1)}$ is a symmetric positive definite matrix with dynamics given by

$$\dot{\boldsymbol{\Sigma}}_i(t) = \boldsymbol{w}_i \boldsymbol{w}_i^\top - K_T \boldsymbol{\Sigma}_i + \sigma_1 \boldsymbol{I} \tag{3.28}$$

where $K_T, \sigma_1 \in \mathbb{R}_{\geq 0}$ are user-defined constants. The $\boldsymbol{w}_i \boldsymbol{w}_i^\top$ term in (3.28) is the standard update term for a least squares covariance matrix. The $K_T \boldsymbol{\Sigma}$ term in (3.28) is used to reduce the weight given to past measurements and allow the covariance matrix to change rapidly as the system evolves. If $K_T = 0$, all values of $\boldsymbol{w}_i$ affect $\boldsymbol{\Sigma}_i$ equally, while if $K_T \gg 0$ little weight is given to past measurements. The $\sigma_1 \boldsymbol{I}$ term in (3.28) is used to ensure that the eigenvalues of $\boldsymbol{\Sigma}_i$ have a positive lower bound and the covariance matrix is therefore always invertible. Since $\boldsymbol{\Sigma}_i$ must be symmetric and positive definite, we initialize this variable as $\boldsymbol{\Sigma}_i(0) = \sigma_0 \boldsymbol{I}$ for some $\sigma_0 \in \mathbb{R}_{>0}$. Since σ_1 is solely used to ensure the invertibility of $\boldsymbol{\Sigma}_i$, it should be small.

The parameter update law depends on the variables with dynamics described by (3.24)–(3.28). Then using a similar update law to the one proposed by Adetola and Guay [2], the parameter estimate has dynamics

$$\dot{\widehat{\boldsymbol{\theta}}}_i = \operatorname{proj}\left\{ \boldsymbol{\Sigma}_i^{-1} \left(\boldsymbol{w}_i (e_i - \widehat{\eta}_i) - \sigma_1 \widehat{\boldsymbol{\theta}}_i \right), \Theta \right\}, \quad \widehat{\boldsymbol{\theta}}_i(0) = \widehat{\boldsymbol{\theta}}_{i_0} \tag{3.29}$$

where $\operatorname{proj}\{\cdot, \Theta\}$ is a Lipschitz projection onto a compact set Θ. The main term in (3.29) is $\boldsymbol{\Sigma}_i^{-1} \boldsymbol{w}_i (e_i - \widehat{\eta}_i)$ which is the standard least squares update law. The other term $-\sigma_1 \boldsymbol{\Sigma}_i^{-1} \widehat{\boldsymbol{\theta}}_i$ compensates for the $\sigma_1 \boldsymbol{I}$ term in (3.28).

The Lipschitz projection operator in (3.29) ensures that the parameter estimates are bounded [70]. This projection guarantees that $\widehat{\boldsymbol{\theta}}_i(t) \in \Theta$ for all $t > 0$, whenever

$\widehat{\boldsymbol{\theta}}_i(0) \in \Theta$. Since the projection operator bounds $\widehat{\boldsymbol{\theta}}_i$, it also guarantees that

$$\widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \operatorname{proj}\left\{\boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{w}_i(e_i-\widehat{\eta}_i)-\sigma_1\widehat{\boldsymbol{\theta}}_i\right),\Theta\right\} \le \widetilde{\boldsymbol{\theta}}_i^\top\left(\boldsymbol{w}_i(e_i-\widehat{\eta}_i)-\sigma_1\widehat{\boldsymbol{\theta}}_i\right). \tag{3.30}$$

This inequality is useful in the proof of convergence of distributed ESC.

Following the requirements from Adetola and Guay [2], this parameter estimation technique converges provided a persistence of excitation condition on $\boldsymbol{w}_i$ is met.

Assumption 3.4 (Persistence of Excitation)**.** There exist constants $\gamma_w, T^- \in \mathbb{R}_{>0}$ such that for each agent i and for any $t > 0$ we have

$$\int_t^{t+T} \boldsymbol{w}_i(\tau)\boldsymbol{w}_i^\top(\tau)\mathrm{d}\tau \ge \gamma_w^- \boldsymbol{I}. \tag{3.31}$$

3.3.4 Proportional-integral extremum-seeking control

The local ESCs perform the optimization of the total cost using a gradient descent based on the gradient estimate $\widehat{\boldsymbol{\theta}}_{1,i}$ from the parameter estimation routine. Typical ESC techniques only have integral action and require a time-scale separation. Guay and Dochain added a proportional term to the ESC input, removing the need for time-scale separation [48]. We use a similar PI ESC technique for each agent:

$$\boldsymbol{u}_i = -K_g\widehat{\boldsymbol{\theta}}_{1,i} + \widehat{\boldsymbol{u}}_i + \boldsymbol{d}_i(t) \tag{3.32}$$

$$\dot{\widehat{\boldsymbol{u}}}_i = -\frac{1}{\tau_I}\widehat{\boldsymbol{\theta}}_{1,i} \tag{3.33}$$

where $K_g \in \mathbb{R}_{>0}$ is a the proportional gain for the ESC and $\tau_I \in \mathbb{R}_{>0}$ is the integral time constant for the ESC. These tuning parameters play a similar role to the tuning parameters of a standard PI controller.

The ESC control law in (3.32) also contains a dither signal $\boldsymbol{d}_i(t)$ which provides persistence of excitation to satisfy Assumption 3.4. We take the dither signals to be sinusoidal with amplitude $D \in \mathbb{R}_{>0}$. To meet Assumption 3.4, all dithers must have different frequencies and no frequency can be the sum of two other frequencies. While there are many ways to select dither frequencies that meet these criteria, the easiest way is to use prime dither frequencies. Since the dither frequencies should also be chosen in a particular bandwidth, depending on the speed of the system's dynamics, we typically choose the dither frequencies to be distinct primes multiplied by the same rational number.

The technique used here is quite similar to the technique used by Guay and Dochain [48] with the main contribution being the extension to multi-agent systems. While the PI control law, (3.32)–(3.33) takes the same form as in [48], the parameter $\widehat{\boldsymbol{\theta}}_{1,i}$ has a different interpretation. In [48], the total cost J is known exactly and $\widehat{\boldsymbol{\theta}}_1$ is the estimate of the gradient of the total cost. In this application, since there are multiple agents, J is not known and $\widehat{\boldsymbol{\theta}}_{1,i}$ is instead the estimate of the gradient of the estimated average cost. Since $\widehat{\boldsymbol{\theta}}_{1,i}$ is based on $\widehat{J}_i$ which is itself uncertain, $\widehat{\boldsymbol{\theta}}_{1,i}$ has additional uncertainty in the multi-agent case. Despite this additional uncertainty, the PI control law based on $\widehat{\boldsymbol{\theta}}_{1,i}$ is still effective.

3.3.5 Convergence analysis

At this point we have developed the entire distributed ESC (Figure 3.1). Agent i controls inputs $\boldsymbol{u}_i$ to the system, and measures a local cost y_i from the system. It also receives some average cost estimates $\widehat{J}_j$ from other agents $j \neq i$. It then computes its own average cost estimate as a convex combination of its existing average

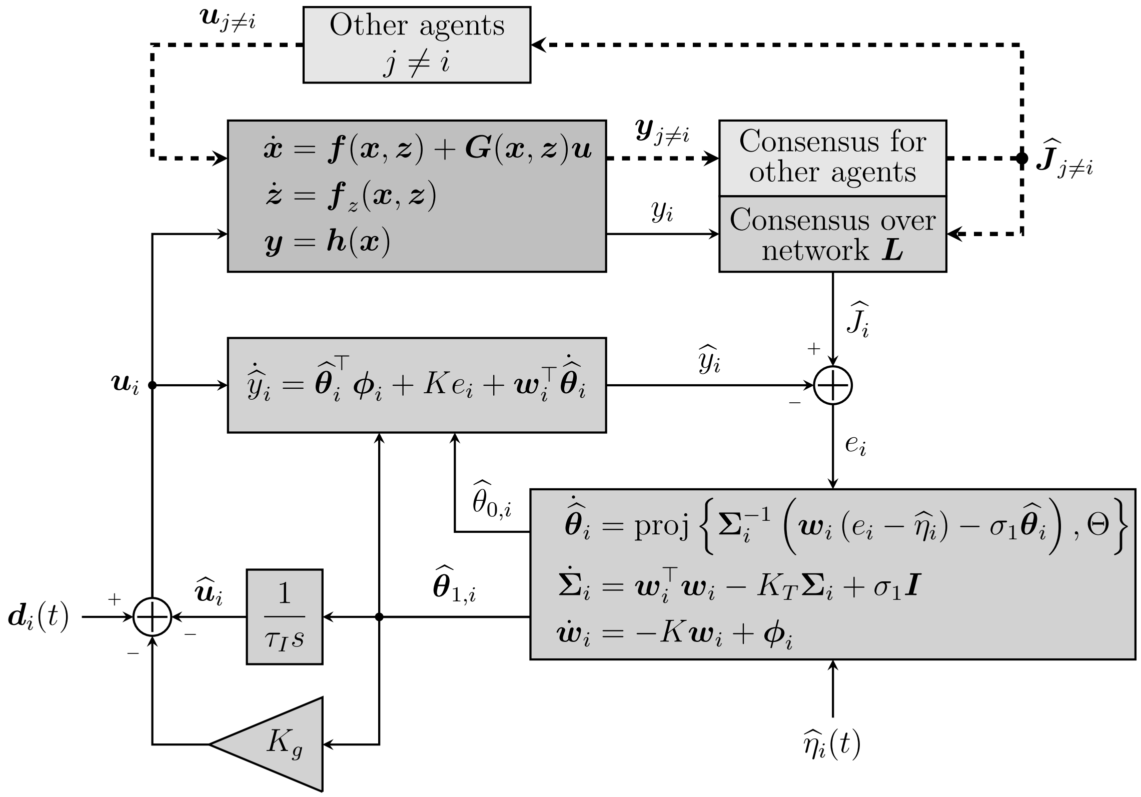

$\boldsymbol{u}_{j\neq i}$
Other agents $j \neq i$
$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{z}) + \boldsymbol{G}(\boldsymbol{x},\boldsymbol{z})\boldsymbol{u}$
$\dot{\boldsymbol{z}} = \boldsymbol{f}_z(\boldsymbol{x},\boldsymbol{z})$
$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x})$
$\boldsymbol{y}_{j\neq i}$
y_i
Consensus for other agents
Consensus over network $\boldsymbol{L}$
$\widehat{\boldsymbol{J}}_{j\neq i}$
$\widehat{J}_i$
$\boldsymbol{u}_i$
$\dot{\widehat{y}}_i = \widehat{\boldsymbol{\theta}}_i^\top \boldsymbol{\phi}_i + Ke_i + \boldsymbol{w}_i^\top \dot{\widehat{\boldsymbol{\theta}}}_i$
$\widehat{y}_i$
+
−
e_i
$\widehat{\theta}_{0,i}$
$\dot{\widehat{\boldsymbol{\theta}}}_i = \text{proj}\left\{\boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{w}_i\left(e_i - \widehat{\eta}_i\right) - \sigma_1\widehat{\boldsymbol{\theta}}_i\right), \Theta\right\}$
$\dot{\boldsymbol{\Sigma}}_i = \boldsymbol{w}_i^\top \boldsymbol{w}_i - K_T\boldsymbol{\Sigma}_i + \sigma_1 \boldsymbol{I}$
$\dot{\boldsymbol{w}}_i = -K\boldsymbol{w}_i + \boldsymbol{\phi}_i$
$\widehat{\boldsymbol{\theta}}_{1,i}$
$\widehat{\boldsymbol{u}}_i$
$\frac{1}{\tau_I s}$
$\boldsymbol{d}_i(t)$
+
−
−
K_g
$\widehat{\eta}_i(t)$

Figure 3.1: Block diagram of continuous-time distributed PI ESC as it appears to one agent. Each agent is aware of its local cost measurement y_i, its own extremum-seeking controller, and some other agents' total cost estimates. It is not aware of other agents' local costs or extremum-seeking controllers.

cost estimate, its local cost measurements, and other agents' average cost estimates. Then using a copy of its input signal and its average cost estimate, it estimates a drift parameter $\widehat{\theta}_{0,i}$ and a gradient parameter $\widehat{\boldsymbol{\theta}}_{1,i}$. The parameter estimation is driven by the estimation error e_i which is the difference between $\widehat{J}_i$ and a predictor state $\widehat{y}_i$. The gradient estimate is used for a PI-type gradient descent with tuning parameters K_g and τ_I. The ESC input is the PI gradient descent plus a dither signal $\boldsymbol{d}_i$.

Next we examine the convergence of the distributed ESC. Its convergence depends on the convergence of the consensus and the parameter estimation algorithms. The consensus must converge so that agents have an accurate value of the total cost to use for parameter estimation; the parameter estimation must converge so that the agents have an accurate value of the gradient to use for extremum-seeking. The main theorem in this chapter says that the consensus, parameter estimation, and overall system converge to the optimum of the total cost function.

Theorem 3.1. *Consider a continuous-time nonlinear control-affine system* (3.4)–(3.5) *with total cost* (3.3) *and let Assumptions 3.1–3.4 hold. Consider the distributed extremum-seeking controller* (3.32)*, the parameter estimation algorithm* (3.24)–(3.29) *and the dynamic average consensus algorithm* (3.13)*. Then there exists ESC tuning parameters K_g, K_T, K, σ_1, and τ_I and dynamic consensus gains κ_0, κ_P and κ_I such that the system converges exponentially to an $\mathcal{O}({}^{K_g}/_{K_T}+\sigma_1\left\|\boldsymbol{\theta}_0\right\|+{}^{D}/_{K_g})$-neighbourhood of the minimizer $\boldsymbol{x}^*$ of the total cost function J.*

Proof. See Appendix A □

The main limitation of this controller is its ability to stabilize systems. Stabilization is provided by the proportional control action with gain K_g. For stabilization, we must have $K_g > pK_g^*$; however, if K_g is too large, the input—and therefore system—changes too quickly and prevent the parameter estimates from converging. Therefore, distributed ESC can only be applied to systems that can be stabilized by gains $K_g^* < \frac{2\gamma_\Sigma^-}{pT}$ which limits the controller to systems with slow unstable poles. Furthermore, since the upper bound for K_g^* decreases with p, the distributed ESC is less effective at stabilization as the number of agents increases.

For large-scale unstable systems controlled by many agents, tuning becomes more difficult as there is a narrower range of proportional ESC gains which are large enough to stabilize the system, but not so large that parameter estimation does not converge.

In the case of fast unstable large-scale systems, it may well be impossible to stabilize the system if too many agents are used. One way to stabilize such systems is to to use fewer agents with each agent controlling more inputs. This approach puts a higher computational burden on each agent as the dimension of $\boldsymbol{u}_i$ and thus Σ_i increases and the controller must invert a larger matrix. In the extreme case, a centralized ESC may be required or the system may be impossible to stabilize using a model-free approach.

3.4 Simulation examples

This section contains three simulation examples which highlight some types of systems which can be solved by distributed ESC. The first system is a 25-dimensional stable linear quadratic (LQ) system which shows distributed ESC's ability to solve large-scale problems. The second example is a 5-dimensional unstable LQ problem which shows distributed ESC's ability to stabilize an unknown system. The third example is a 5-agent nonlinear system with nonlinear costs which shows that distributed ESC is not limited to linear systems.

3.4.1 Stable 25 agent linear system

Consider a system with 25 states, 25 inputs and linear dynamics given by

$$\dot{\boldsymbol{x}} = \begin{bmatrix} -2 & 1 & 0 & 0 & \dots & 0 \\ 0 & -2 & 1 & 0 & \ddots & \vdots \\ 0 & 0 & -2 & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & 1 & 0 \\ \vdots & \ddots & \ddots & 0 & -2 & 1 \\ 0 & \dots & 0 & 0 & 1 & -2 \end{bmatrix} \boldsymbol{x} + \boldsymbol{u}.$$

This system is controlled by 25 agents. The i^{th} agent measures a local cost y_i and manipulates the input u_i. The local costs are quadratic functions of the states

$$y_i = \begin{cases} (x_i - i)^2 + 2(x_{i+1} - 2i)^2 & \text{if } i \in \{1, \dots, 24\} \\ (x_{25} - 25)^2 + x_1^2 & \text{if } i = 25. \end{cases}$$

The total cost is the sum of the local costs $J = \sum_{i=1}^{25} y_i$. Each agent communicates with two neighbours (Figure 3.2), resulting in the Laplacian matrix

$$\boldsymbol{L} = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 & -1 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & \ddots & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & -1 & 0 \\ 0 & \vdots & \ddots & -1 & 2 & -1 \\ -1 & 0 & \dots & 0 & -1 & 2 \end{bmatrix}.$$

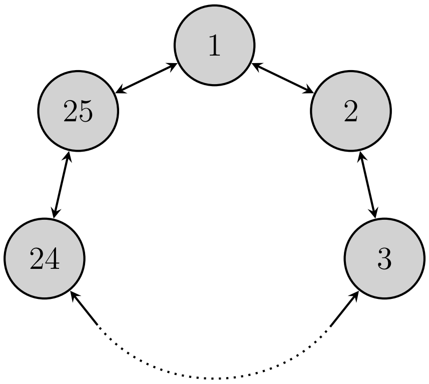


Figure 3.2: Communication graph for the 25 agents controlling the stable linear system. This graph has the structure of a cycle.

Table 3.1: Tuning parameters for the 25-agent linear system controlled by continuous-time distributed ESC.

κ_0	κ_P	κ_I	K	σ_1	K_T	K_g	τ_I
1×10^4	1×10^4	1×10^4	50	1×10^{-6}	50	0.25	1

Each agent implements an identical PIESC algorithm with the same tuning parameters (Table 3.1) and initial conditions (Table 3.2). The dither signals each have different angular frequencies ranging between $\omega_1 = 123$ and $\omega_{25} = 251$.

The optimal cost can be computed to be $J^* = 2883.16$. The control system was simulated from $\boldsymbol{x}(0) = \mathbf{0}$ using the PIESC controller. The distributed ESC algorithm is able to quickly find the optimal input, state, and minimize the total cost (Figure 3.3).

Table 3.2: Initial conditions for the 25-agent linear system controlled by continuous-time distributed ESC.

$\widehat{y}_i(0)$	$\widehat{J}_i(0)$	$\rho_i(0)$	$\boldsymbol{w}_i(0)$	$\widehat{\eta}_i(0)$	$\widehat{\boldsymbol{\theta}}_i(0)$	$\boldsymbol{\Sigma}_i(0)$
$y_i(0)$	$y_i(0)$	0	$[0,0]^\top$	0	$[0,0]^\top$	$\boldsymbol{I}_{2\times2}$

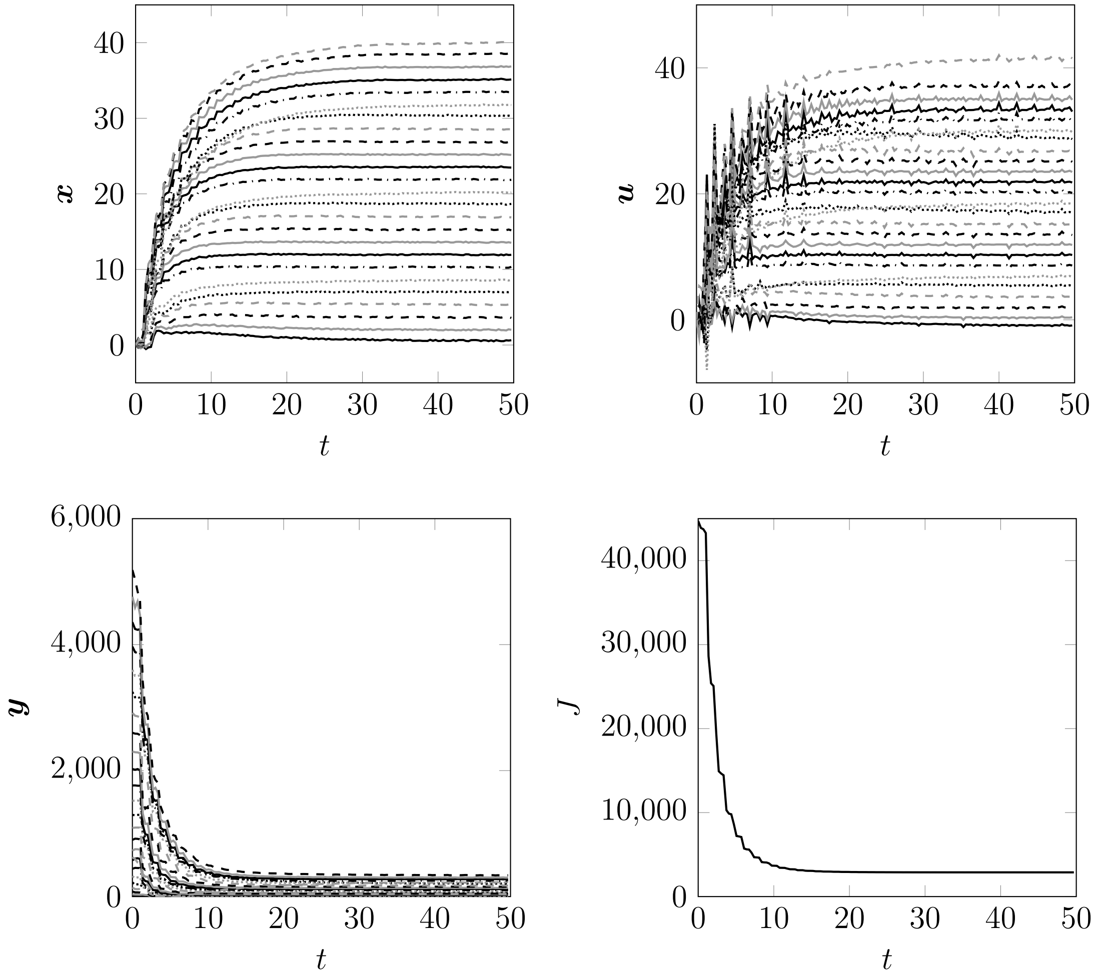


Figure 3.3: State (top left), input (top right), local cost (bottom left), and total cost (bottom right) trajectories for the stable 25-agent linear system controlled by PIESC with consensus.

3.4.2 Unstable 5 agent linear system

Consider a system with 5 states, 5 inputs and linear dynamics given by

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0.1108 & 1 & 0 & 0 & 0 \\ 0 & 0.1367 & 1 & 0 & 0 \\ 0 & 0 & 0.0548 & 1 & 0 \\ 0 & 0 & 0 & 0.1172 & 1 \\ 0 & 0 & 0 & 0 & 0.1000 \end{bmatrix} \boldsymbol{x} + \boldsymbol{u}.$$

Table 3.3: Tuning parameters for the 5-agent unstable linear system controlled by continuous-time distributed ESC.

κ_0	κ_P	κ_I	K	σ_1	K_T	K_g	τ_I
1×10^4	1×10^4	1×10^4	50	1×10^{-6}	50	0.5	1

This system has poles at 0.1108, 0.1367, 0.0548, 0.1172 and 0.1000 and which are all slow, unstable poles. This system is controlled by 5 agents. The i^{th} agent measures a local cost y_i and manipulates the input u_i. The local costs are quadratic functions of the states:

$$\begin{aligned} y_1 &= (x_1 - 1)^2 + 2(x_2 - 2)^2 & y_4 &= (x_4 - 4)^2 + 2(x_5 - 8)^2 \\ y_2 &= (x_2 - 2)^2 + 2(x_3 - 4)^2 & y_5 &= (x_5 - 5)^2 + 2x_1^2. \\ y_3 &= (x_3 - 3)^2 + 2(x_4 - 6)^2 & & \end{aligned}$$

The total cost is the sum of the local costs $J = \sum_{i=1}^{5} y_i$. Each agent communicates with two neighbours (Figure 3.4), resulting in the Laplacian matrix

$$\boldsymbol{L} = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}.$$

Each agent implements an identical PIESC algorithm with the same tuning parameters (Table 3.3) and initial conditions (Table 3.4). The dither signals each have different angular frequencies ranging between $\omega_1 = 123$ and $\omega_5 = 251$.

The optimal cost can be calculated to be $J^* = 9.833$, which occurs when the input

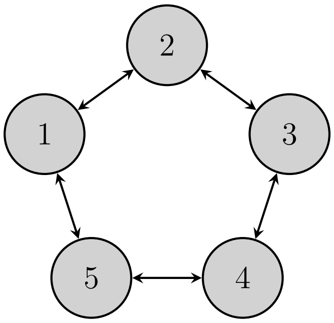


Figure 3.4: Communication graph for the 5 agents controlling the unstable linear system. This graph has the structure of a cycle.

Table 3.4: Initial conditions for the 5-agent unstable linear system controlled by continuous-time distributed ESC.

$\widehat{y}_i(0)$	$\widehat{J}_i(0)$	$\rho_i(0)$	$\boldsymbol{w}_i(0)$	$\widehat{\eta}_i(0)$	$\widehat{\boldsymbol{\theta}}_i(0)$	$\boldsymbol{\Sigma}_i(0)$
$y_i(0)$	$y_i(0)$	0	$[0,0]^\top$	0	$[0,0]^\top$	$\boldsymbol{I}_{2\times 2}$

bias and state are:

$$\boldsymbol{u}^* = \begin{bmatrix} -2.0554 & -3.9400 & -5.5344 & -7.6253 & -0.7000 \end{bmatrix}^\top$$
$$\boldsymbol{x}^* = \begin{bmatrix} 0.5000 & 2.0000 & 3.6667 & 5.3333 & 7.0000 \end{bmatrix}^\top.$$

The control system was simulated from $\boldsymbol{x}(0) = \mathbf{0}$. The simulation results (Figure 3.5) show that the distributed extremum-seeking controller is able to stabilize the unstable system and find the minimum of the total cost function. Initially, some oscillatory transient behaviour is observed as the parameter estimates are still far from their true values. However, once the algorithm has been running for a short period of time, the controllers have accurate estimates of the gradient and can then move in the appropriate direction. All 5 input biases converge to the optimal input $\boldsymbol{u}^*$. Despite the instability of the open-loop system, the closed-loop system is stable

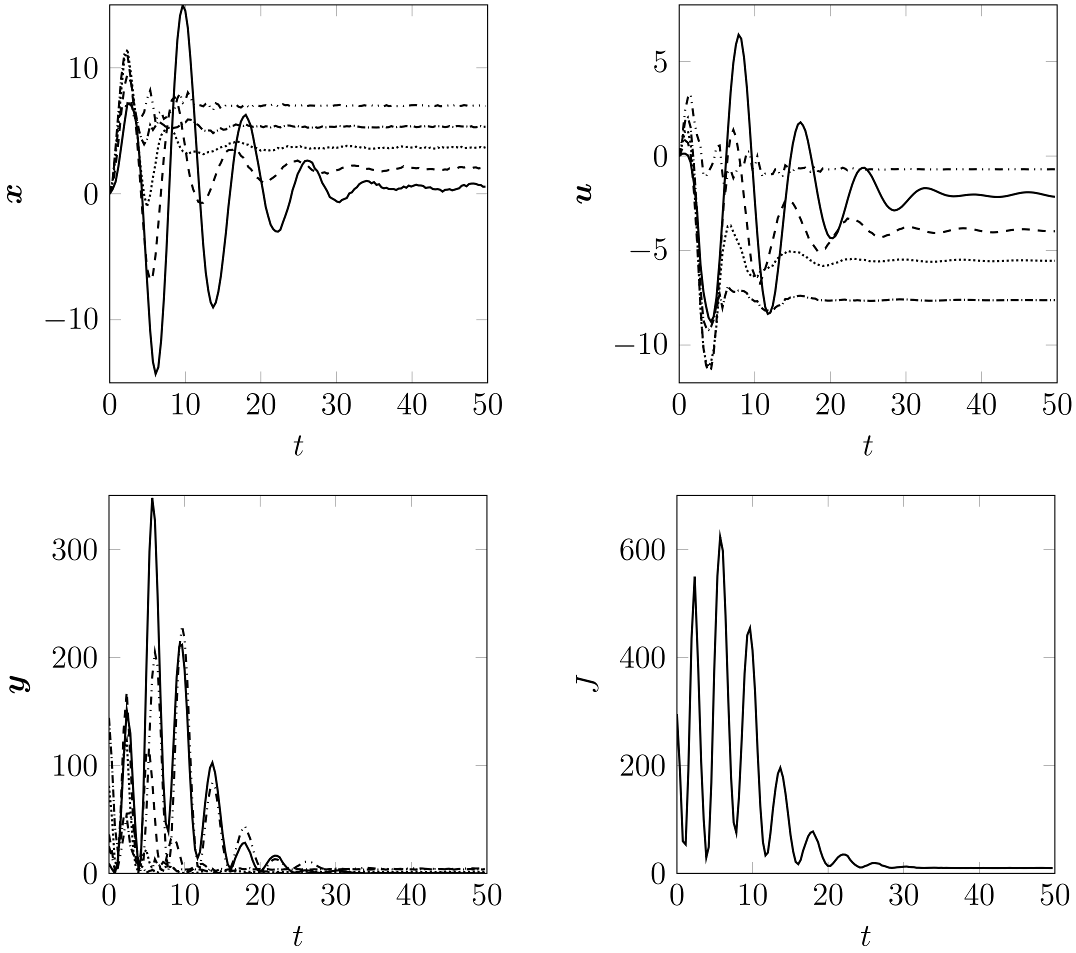


Figure 3.5: State (top left), input (top right), local cost (bottom left), and total cost (bottom right) trajectories for the unstable 5-agent linear system controlled by PIESC with consensus.

and so the states, converge to their optimal values. At these conditions, the convex total cost function is also minimized.

3.4.3 Unstable 5 agent nonlinear system

Consider a system with 6 states, 5 inputs and nonlinear dynamics given by

$$\begin{aligned}
\dot{x}_1 &= \frac{3}{10}x_1 - 4x_1^3 + x_2x_5 + \frac{u_1}{x_1^2+1} + 2u_4 \\
\dot{x}_2 &= \sin(x_2) + zx_4 + \frac{1}{4}\sin(z)u_3 - u_5 \\
\dot{x}_3 &= -x_3 + \log(x_3^2+1) + x_1x_4 + 3u_2 + \frac{1}{3}\operatorname{atan}(x_3)u_4 \\
\dot{x}_4 &= -\frac{7}{10}x_4 + \frac{x_5^2}{x_5^2+x_2^2+1} + u_1 + \frac{1}{5}\mathrm{e}^{-x_2^2}u_5 \\
\dot{x}_5 &= -\cos(x_1+z) + \frac{2}{5}x_5 + \frac{1}{2}\cos(x_4)u_2 - 2u_3 \\
\dot{z} &= -z + x_3.
\end{aligned}$$

This system is controlled by 5 agents. The i^{th} agent measures a local cost y_i and manipulates the input u_i. The local costs are quadratic functions of the states:

$$\begin{aligned}
y_1 &= (x_5-8)^2 + 2(x_2-2)^2 \\
y_2 &= (x_2+5)^4 + (x_3-4)^2 + |x_5+3| - x_1 \\
y_3 &= (x_3-3)^2 + \cosh(x_4-6) + (x_2-x_1)^2 \\
y_4 &= \frac{1}{2}(x_4-4)^2 + (x_3-x_1x_2)^2 \\
y_5 &= (x_5+x_2)^2 + x_1.
\end{aligned}$$

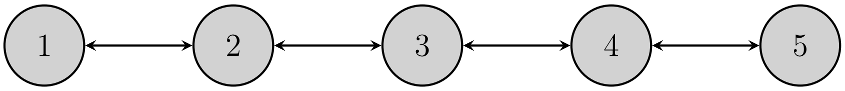


Figure 3.6: Communication graph for the 5 agents controlling the nonlinear system. This graph has the structure of a path which is a type of tree.

Table 3.5: Tuning parameters for the 5-agent nonlinear system controlled by continuous-time distributed ESC.

κ_0	κ_P	κ_I	K	σ_1	K_T	K_g	τ_I
1×10^4	1×10^4	1×10^4	50	1×10^{-7}	50	0.25	2

The total cost is the sum of the local costs $J = \sum_{i=1}^{5} y_i$. Each agent communicates with one or two neighbours (Figure 3.6), resulting in the Laplacian matrix

$$\boldsymbol{L} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

Each agent implements an identical PIESC algorithm with the same tuning parameters (Table 3.5) and initial conditions (Table 3.6). The dither signals each have different angular frequencies ranging between $\omega_1 = 123$ and $\omega_5 = 251$.

The optimal cost can be calculated to be $J^* = 90.95$, which occurs when the input

Table 3.6: Initial conditions for the 5-agent nonlinear system controlled by continuous-time distributed ESC.

$\widehat{y}_i(0)$	$\widehat{J}_i(0)$	$\rho_i(0)$	$\boldsymbol{w}_i(0)$	$\widehat{\eta}_i(0)$	$\widehat{\boldsymbol{\theta}}_i(0)$	$\boldsymbol{\Sigma}_i(0)$
$y_i(0)$	$y_i(0)$	0	$[0,0]^\top$	0	$[0,0]^\top$	$\boldsymbol{I}_{2\times 2}$

bias and state are

$$\boldsymbol{u}^* = \begin{bmatrix} 2.834 & 2.136 & 1.642 & 3.440 & 19.065 \end{bmatrix}^\top$$
$$\boldsymbol{x}^* = \begin{bmatrix} -1.352 & -3.235 & 3.791 & 5.070 & 5.367 \end{bmatrix}^\top .$$

The control system was simulated from $\boldsymbol{x}(0) = \mathbf{0}$ (Figure 3.7) and the distributed ESC steered the system to the optimum of the steady-state cost function. Since this system is nonlinear the parameter estimates $\boldsymbol{\theta}_i$ change as the system evolves. Despite having no knowledge of a model for the system, the distributed ESC is able to track the time-varying parameters and therefore always has an accurate local model. Using this ever changing local model, it successfully implements a gradient descent and reaches the optimum of the steady-state cost.

3.5 Conclusion

In this chapter, a distributed ESC technique was proposed to solve a class of RTO problems over a network of dynamic agents with unknown dynamics. A dynamic average consensus algorithm provides each agent with an estimate of the total network cost. Each agent implements an extremum-seeking controller which contributes to the optimization of the total cost, in a cooperative fashion. The extremum-seeking control technique is based on a proportional-integral approach that avoids the explicit need for time-scale separation. This technique is novel as it is the first example of a distributed model-free output-feedback controller which can simultaneously stabilize and optimize a nonlinear system. Three simulation examples were used to show the techniques applicability to large-scale, unstable, and nonlinear systems.

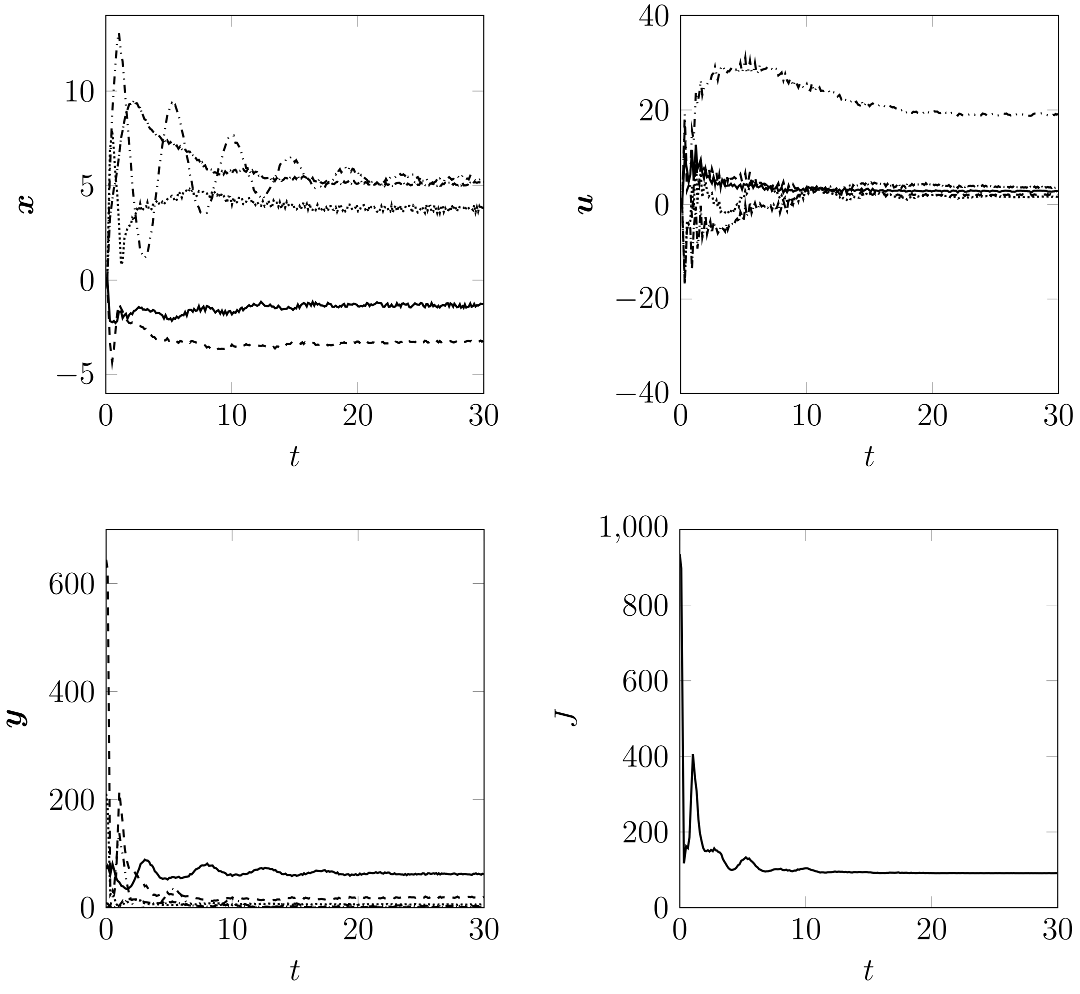


Figure 3.7: State (top left), input (top right), local cost (bottom left), and total cost (bottom right) trajectories for the 5-agent nonlinear system controlled by PIESC with consensus.

Chapter 4

Discrete-time distributed extremum-seeking control

This chapter continues the development of distributed ESC by extending it to discrete-time systems. Section 4.2 defines the problem to be solved. A discrete-time distributed extremum-seeking controller is designed in Section 4.3 and its convergence is analyzed in Section 4.4. Several discrete-time simulation examples are provided in Section 4.5.

4.1 Introduction

Distributed ESC has been shown in Chapter 3 to be effective when implemented in continuous-time. It is also important to consider its discrete-time implementation so that it can be applied to commonly used digital control systems. In this chapter we develop a discrete-time distributed extremum-seeking controller. It has the same main components—consensus, parameter estimation, and gradient-based optimization—as the continuous-time controller, but with discrete-time algorithms for each component.

Both extremum-seeking control and average consensus have already been considered in discrete-time by several researchers. Both extremum-seeking and consensus are more difficult in discrete-time than in continuous-time as time step limitations can lead to poor transient performance and instability. A proportional-integral version of discrete-time ESC for a single agent has been considered by Guay [45] using a recursive least squares (RLS)-based time-varying parameter estimation technique to estimate the gradient. Several n^{th}-order discrete-time dynamic average consensus algorithms have been considered by Kia *et al.* [66] and can provide dynamic average consensus for signals with bounded n^{th}-differences.

This chapter provides an effective method for distributed extremum-seeking control of large-scale systems in discrete-time. The system has unknown nonlinear dynamics, which may potentially be unstable. Each agent is equipped with a sensor and measures a local cost. The global objective is to minimize the sum of the local costs. Agents communicate over a network and use the dynamic average consensus algorithm from Kia *et al.* [66] to obtain an estimate of the average cost. Each agent implements a extremum-seeking controller using its average cost estimate. The PI ESCs are based on the controller proposed by Guay [45] which removes the explicit need for time-scale separation and allows the ESC to stabilize unknown nonlinear discrete-time systems. The estimation algorithm provided in this chapter has different update laws for the covariance matrix and parameter estimate than in [45], which are simpler and result in convergence to a smaller neighbourhood of the true values. By adding a proportional term to the standard ESC technique, the PI ESC is able to improve transient performance and thus eliminates the need for time scale separation. The resulting distributed control system is able to achieve global RTO objectives and can be used to stabilize a plant with unknown dynamics using only measurements of local cost functions.

4.2 Problem description

Consider a general discrete-time non-linear control-affine system

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\boldsymbol{u}[k] \tag{4.1}$$

where $k \in \mathbb{Z}$ is the time, $\boldsymbol{x} \in \mathbb{X} \subseteq \mathbb{R}^n$ is the state, and $\boldsymbol{u} \in \mathbb{U} \subseteq \mathbb{R}^m$ is the input, and $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{G}(\boldsymbol{x})$ are smooth vector fields. While these vector fields do not depend explicitly on a time step, in reality, the system's dynamics evolve in continuous-time so $\boldsymbol{f}$ and $\boldsymbol{G}$ are obtained by the exponential map and are therefore $\mathcal{O}\left(\Delta t\right)$.

This system is controlled by p agents which each measure a local output y_i and control m_i inputs $\boldsymbol{u}_i$. The vector of outputs is $\boldsymbol{y} = [y_1, \dots, y_p]^\top \in \mathbb{R}^p$ and the vector of inputs is $\boldsymbol{u} = [\boldsymbol{u}_1^\top, \dots, \boldsymbol{u}_p^\top]^\top \in \mathbb{R}^m$. The cost function can therefore be modeled by

$$\boldsymbol{y}[k] = \boldsymbol{h}\left(\boldsymbol{x}[k]\right) \tag{4.2}$$

$$J[k] = \sum_{i=1}^{p} y_i[k] = H\left(\boldsymbol{x}[k]\right) \tag{4.3}$$

where J is the total cost which is simply the sum of the local costs. We also consider the average cost $\frac{1}{p}J$.

Since the controllers do not have a model for the system, nor measurements of $\boldsymbol{x}$, it is useful to also describe the system by an input-output model for each agent. For some nominal input $\widehat{\boldsymbol{u}}_i$, the average cost dynamics can be expressed as

$$\begin{aligned}\frac{1}{p}\Delta J[k+1] &= \frac{1}{p}J[k+1] - \frac{1}{p}J[k] \\ &= \frac{1}{p}H\left(\boldsymbol{x}[k+1]\right) - \frac{1}{p}H\left(\boldsymbol{x}[k]\right)\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{p}H\big(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\boldsymbol{u}[k]\big) - \frac{1}{p}H\left(\boldsymbol{x}[k]\right)\\
&= \frac{1}{p}H\big(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\boldsymbol{u}[k]\big)\\
&\quad - \frac{1}{p}H\big(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\widehat{\boldsymbol{u}}[k]\big)\\
&\quad + \frac{1}{p}H\big(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\widehat{\boldsymbol{u}}[k]\big) - \frac{1}{p}H\left(\boldsymbol{x}[k]\right).
\end{aligned}$$

By the mean-value theorem [11], there exists $\overline{\boldsymbol{x}} = \left[\overline{x}_1, \ldots, \overline{x}_n\right]^\top$ with $\overline{x}_i = f_i\left(\boldsymbol{x}[k]\right) + \delta_i \boldsymbol{g}_i[k]\left(\boldsymbol{u}[k] - \widehat{\boldsymbol{u}}[k]\right)$ for some $\delta_i \in [0,1]$ such that

$$\begin{aligned}
\frac{1}{p}\Delta J[k+1] &= \frac{1}{p}\nabla H\left(\overline{\boldsymbol{x}}\right)\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\left(\boldsymbol{u}[k] - \widehat{\boldsymbol{u}}[k]\right)\\
&\quad + \frac{1}{p}H\big(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\widehat{\boldsymbol{u}}[k]\big) - \frac{1}{p}H\left(\boldsymbol{x}[k]\right)\\
&= \frac{1}{p}H\big(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\widehat{\boldsymbol{u}}[k]\big) - \frac{1}{p}H\left(\boldsymbol{x}[k]\right)\\
&\quad + \frac{1}{p}\sum_{j\neq i}\nabla H\left(\overline{\boldsymbol{x}}\right)\boldsymbol{G}_j\left(\boldsymbol{x}[k]\right)\left(\boldsymbol{u}_j[k] - \widehat{\boldsymbol{u}}_j[k]\right)\\
&\quad + \frac{1}{p}\nabla H\left(\overline{\boldsymbol{x}}\right)\boldsymbol{G}_i\left(\boldsymbol{x}[k]\right)\left(\boldsymbol{u}_i[k] - \widehat{\boldsymbol{u}}_i[k]\right)\\
&= \theta_{0,i}[k] + \boldsymbol{\theta}_{1,i}^\top[k]\left(\boldsymbol{u}_i[k] - \widehat{\boldsymbol{u}}_i[k]\right)\\
&= \boldsymbol{\theta}_i^\top[k]\boldsymbol{\phi}_i[k] \qquad (4.4)
\end{aligned}$$

where

$$\begin{aligned}
\theta_{0,i}[k] &= \frac{1}{p}\Big(H\Big(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\widehat{\boldsymbol{u}}[k]\Big)\\
&\quad - \frac{1}{p}H\left(\boldsymbol{x}[k]\right) \qquad (4.5)\\
&\quad + \frac{1}{p}\sum_{j\neq i}\nabla H\left(\overline{\boldsymbol{x}}\right)\boldsymbol{G}_j\left(\boldsymbol{x}[k]\right)\left(\boldsymbol{u}_j[k] - \widehat{\boldsymbol{u}}_j[k]\right)
\end{aligned}$$

$$\boldsymbol{\theta}_{1,i}[k] = \frac{1}{p}\boldsymbol{G}_i^\top\left(\boldsymbol{x}[k]\right)\nabla^\top H\left(\overline{\boldsymbol{x}}\right). \tag{4.6}$$

Note that the parameterization of $\Delta J[k+1] = \boldsymbol{\theta}_i^\top[k]\boldsymbol{\phi}_i[k]$ in (4.4) is exact due to the use of the mean-value theorem and is not simply a linear approximation. The time-varying parameter $\theta_{0,i}$ describes the effect of drift and other agents' inputs on the system; it describes all the changes in the total cost which cannot be affected by agent i. The time-varying parameter $\boldsymbol{\theta}_{1,i}$ is a vector describing the how each of agent i's inputs affects the total cost. If $\boldsymbol{\theta}_i$ is known, the agent can minimize the total cost by setting $\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i = -K_g\boldsymbol{\theta}_{1,i}$. If all agents implement this control law with a suitably large gain, they can dominate the drift term and ensure that J is decreasing.

The objective of the controller is to stabilize the dynamics and minimize the total cost at steady-state. Let $\mathsf{S} = \{(\boldsymbol{x},\boldsymbol{u}) \in \mathbb{X}\times\mathbb{U} \mid \boldsymbol{0} = \boldsymbol{f}\left(\boldsymbol{x}\right) + \boldsymbol{G}\left(\boldsymbol{x}\right)\boldsymbol{u}\}$ be the steady-state manifold. We assume that for every $\boldsymbol{u} \in \mathbb{U}$, there exists a unique $\boldsymbol{x} \in \mathbb{X}$ such that $(\boldsymbol{x},\boldsymbol{u}) \in \mathsf{S}$. Therefore, we can instead describe the steady-state by a map $\boldsymbol{\pi} : \mathbb{U} \to \mathbb{X}$. The steady-state cost $\ell : \mathbb{U} \to \mathbb{R}$, is defined by $\ell\left(\boldsymbol{u}\right) = H\left(\boldsymbol{\pi}\left(\boldsymbol{x}\right)\right)$. Let $\boldsymbol{u}^* = \operatorname{argmin}\left(\ell\right)$ and $\boldsymbol{x}^* = \boldsymbol{\pi}\left(\boldsymbol{u}^*\right)$. The control objective is to find $\boldsymbol{u}^*$ and stabilize $\boldsymbol{x}$ to $\boldsymbol{x}^*$.

For the optimization problem to have a well-defined solution, we need to assume that the steady-state cost is locally convex around $\boldsymbol{u}^*$.

Assumption 4.1 (Convexity of total cost)**.** There exists some positive constant $\beta_1 \in \mathbb{R}_{>0}$ such that for all $\boldsymbol{u} \in \mathbb{U}$, the equilibrium input-output map is such that

$$\left.\frac{\partial \ell}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}}\left(\boldsymbol{u} - \boldsymbol{u}^*\right) \geq \beta_1\left\|\boldsymbol{u} - \boldsymbol{u}^*\right\|^2. \tag{4.7}$$

For the stabilization problem to be well-defined, the system must be stabilizable using output-feedback. We assume that the dynamics (4.1) have relative degree one so there exists a diffeomorphism $\boldsymbol{\psi} : \mathbb{X} \to \mathbb{R}^m \times \mathbb{R}^{n_z}$ with $\boldsymbol{\psi}(\boldsymbol{x}) = (\boldsymbol{\xi}, \boldsymbol{z})$ such that

$$\boldsymbol{\xi}[k+1] = \boldsymbol{\xi}[k] + \boldsymbol{f}_\xi\left(\boldsymbol{\xi}[k], \boldsymbol{z}[k]\right) + \boldsymbol{G}_\xi\left(\boldsymbol{\xi}[k], \boldsymbol{z}[k]\right)\boldsymbol{u}[k] \tag{4.8}$$

$$\boldsymbol{z}[k+1] = \boldsymbol{z}[k] + \boldsymbol{f}_z\left(\boldsymbol{\xi}[k], \boldsymbol{z}[k]\right) \tag{4.9}$$

$$\boldsymbol{y}[k] = \boldsymbol{h}_\xi\left(\boldsymbol{\xi}[k]\right) \tag{4.10}$$

where $\boldsymbol{z} \in \mathbb{R}^{n_z}$ are the zero dynamics of the system and $\boldsymbol{G}_\xi \in \mathbb{R}^{m\times m}$ is invertible. Let $\boldsymbol{\psi}\left(\boldsymbol{x}\right) = (\boldsymbol{\xi}, \boldsymbol{z}) = \left(\boldsymbol{\psi}_\xi\left(\boldsymbol{x}\right), \boldsymbol{\psi}_z\left(\boldsymbol{x}\right)\right)$. Then $\boldsymbol{f}_\xi = \boldsymbol{\psi}_\xi \circ \boldsymbol{f} \circ \boldsymbol{\psi}^{-1}$, $\boldsymbol{G}_\xi = \boldsymbol{\psi}_\xi \circ \boldsymbol{G} \circ \boldsymbol{\psi}^{-1}$, $\boldsymbol{f}_z = \boldsymbol{\psi}_z \circ \boldsymbol{G} \circ \boldsymbol{\psi}^{-1}$, and $\boldsymbol{h}_\xi\left(\boldsymbol{\xi}\right) = \boldsymbol{h} \circ \boldsymbol{\psi}^{-1}\left(\boldsymbol{\xi}, \boldsymbol{z}\right)$ for any $\boldsymbol{z} \in \mathbb{R}^{n_z}$. We assume that the zero dynamics are stable and therefore the system can be stabilized by a controller of the form $\boldsymbol{u}_i[k] = -K_g^*\boldsymbol{\theta}_{1,i}[k] + \widehat{\boldsymbol{u}}_i$. For a stable system $K_g^* = 0$. This stabilizability assumption can be expressed in terms of a Lyapunov function.

Assumption 4.2 (Stabilizability)**.** For a fixed $\widehat{\boldsymbol{u}} \in \mathbb{U}$, let $\widetilde{\boldsymbol{x}}[k] = \boldsymbol{x}[k] - \boldsymbol{\pi}\left(\widehat{\boldsymbol{u}}\right)$. Then there exists a positive definite function V_z and positive constants $\beta_2, \beta_3 \in \mathbb{R}_{>0}$ such that

$$\beta_2 \left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \leq V_z\left(\boldsymbol{z}[k]\right) + H\left(\boldsymbol{x}[k]\right) \leq \beta_3 \left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 . \tag{4.11}$$

Furthermore, there exists $K_g^*, \beta_4 \in \mathbb{R}_{\geq 0}$ such that

$$\begin{aligned} V_z\left(\boldsymbol{z}[k+1]\right) + H\left(\boldsymbol{x}[k] + \boldsymbol{f}(\boldsymbol{x}[k]) + \boldsymbol{G}(\boldsymbol{x}[k])\left(\widehat{\boldsymbol{u}}[k] - K_g^*\boldsymbol{\theta}_1[k]\right)\right)& \\ -V_z\left(\boldsymbol{z}[k]\right) - H\left(\boldsymbol{x}[k]\right) \leq -\beta_4 \left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 .& \end{aligned} \tag{4.12}$$

To keep the problem general, we would like to make as few assumptions as possible on $\boldsymbol{f}$, $\boldsymbol{G}$, H, and $\boldsymbol{\pi}$. The main assumption is that they are Lipschitz.

Assumption 4.3 (Lipschitz properties)**.** The vector fields $\boldsymbol{f}$ and $\boldsymbol{G}$, the cost function H, and the steady-state map $\boldsymbol{\pi}$, are Lipschitz continuous. Therefore there exist constants $L_f, L_G, L_H, L_\pi \in \mathbb{R}_{\geq 0}$ such that for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{X}$ and $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{U}$, we have

$$\left\|\boldsymbol{f}\left(\boldsymbol{x}_1\right) - \boldsymbol{f}\left(\boldsymbol{x}_2\right)\right\| \leq L_f \left\|\boldsymbol{x}_1 - \boldsymbol{x}_2\right\| \tag{4.13}$$

$$\left\|\boldsymbol{G}\left(\boldsymbol{x}_1\right) - \boldsymbol{G}\left(\boldsymbol{x}_2\right)\right\| \leq L_G \left\|\boldsymbol{x}_1 - \boldsymbol{x}_2\right\| \tag{4.14}$$

$$\left\|H\left(\boldsymbol{x}_1\right) - H\left(\boldsymbol{x}_2\right)\right\| \leq L_H \left\|\boldsymbol{x}_1 - \boldsymbol{x}_2\right\| \tag{4.15}$$

$$\left\|\boldsymbol{\pi}\left(\boldsymbol{u}_1\right) - \boldsymbol{\pi}\left(\boldsymbol{u}_2\right)\right\| \leq L_\pi \left\|\boldsymbol{u}_1 - \boldsymbol{u}_2\right\|. \tag{4.16}$$

Since $\boldsymbol{f}, \boldsymbol{G} \in \mathcal{O}\left(\Delta t\right)$, we also have that $L_f, L_G \in \mathcal{O}\left(\Delta t\right)$.

4.3 Discrete-time controller design

We use distributed extremum-seeking control to minimize the steady-state cost. This technique only requires measurements of the local costs and communication between agents, but does not require a model. There are three main components to this controller: dynamic average consensus, parameter estimation, and the PI extremum-seeking controller (Figure 4.1). The PI extremum-seeking controller consists of proportional and integral components and a dither signal.

Extremum-seeking control is based on a gradient descent algorithm. The controller can minimize the total cost by moving in the direction of the gradient $\boldsymbol{\theta}_{1,i}$. Unfortunately, $\boldsymbol{\theta}_{1,i}$ is not measured and changes with time as $\boldsymbol{x}$ changes. Since $\boldsymbol{\theta}_{1,i}$ relates the total cost to the input by (4.4), it can be estimated from data of $\frac{1}{p}J$ and $\boldsymbol{u}_i - \widehat{\boldsymbol{u}}_i$ using recursive least squares. RLS involves a forgetting factor $\alpha \in (0, 1)$ which

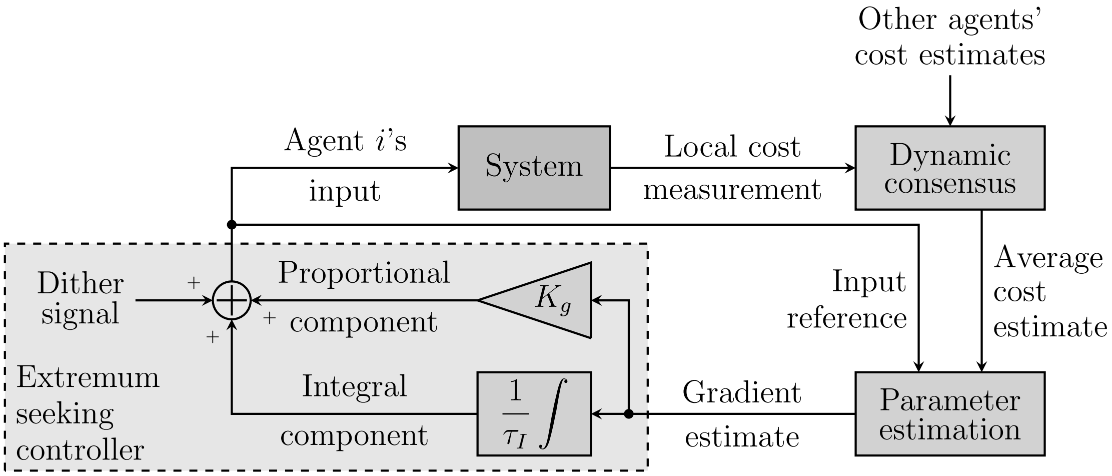


Figure 4.1: Block diagram of the main components of the discrete-time distributed ESC as seen by one agent.

can be used to control how much old measurements affect the regression to control how fast $\widehat{\boldsymbol{\theta}}_{1,i}$ changes. Unfortunately, the agents only measure y_i and not $\frac{1}{p}J$ which is needed for the parameter estimation. Since the agents can communicate with each other, a dynamic average consensus is used to provide each agent with an estimate $\widehat{J}_i$ of the total cost which is then used in the regression.

4.3.1 Dynamic average consensus

We use the consensus algorithm from [66] which has the form

$$\begin{aligned}\begin{bmatrix}\widehat{\boldsymbol{J}}[k+1]-\widehat{\boldsymbol{J}}[k]\\ \boldsymbol{\rho}[k+1]-\boldsymbol{\rho}[k]\end{bmatrix} &= \begin{bmatrix}-\kappa_P\boldsymbol{I}-\kappa_I\boldsymbol{L} & -\boldsymbol{I}\\ \kappa_P\kappa_I\boldsymbol{L} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\widehat{\boldsymbol{J}}[k]\\ \boldsymbol{\rho}[k]\end{bmatrix}\Delta t\\ &\quad+\begin{bmatrix}\kappa_P\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{y}[k]\Delta t+\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}\Delta\boldsymbol{y}[k]\end{aligned} \tag{4.17}$$

where $\kappa_P, \kappa_I \in \mathbb{R}_{>0}$ are tuning parameters. As with the continuous-time consensus algorithm (see Subsection 3.3.1), this algorithm can be implemented in a distributed fashion. For every agent's estimate to converge to the average, we place the following requirements on the communication network.

Assumption 4.4 (Graph properties)**.** The network structure can be described by a strongly connected time-invariant weight-balanced directed graph.

For a network which satisfies Assumption 4.4, when the changes in the average cost are bounded, the convergence error $\widetilde{\boldsymbol{J}} = \widehat{\boldsymbol{J}} - \frac{1}{p}J$ is ultimately bounded. The size of the bound depends on the time step, tuning parameters, and network structure. In particular, it is inversely proportional to λ_2, the second smallest eigenvalue of $\overline{\boldsymbol{L}} = \frac{1}{2}\left(\boldsymbol{L} + \boldsymbol{L}^\top\right)$. Note that since $\overline{\boldsymbol{L}}$ is symmetric, it can be viewed as the Laplacian of a undirected graph. For an arbitrary undirected Laplacian, all of the eigenvalues are non-negative and $\lambda_1 = 0$. Furthermore, when the graph is connected $\lambda_2 > 0$. We have assumed a strongly connected graph, which results in the graph associated with $\overline{\boldsymbol{L}}$ being a connected undirected graph. Therefore $\overline{\boldsymbol{L}}$ has only one zero eigenvalue and so $\lambda_2 > 0$ and the consensus algorithm is convergent.

4.3.2 Time-varying parameter estimation

The parameters $\boldsymbol{\theta}_i$ are estimated using a modification of recursive least squares that includes some additional features that improve the performance. RLS is a method of estimating time-varying linear parameters which is based on minimizing a weighted squared error [84]. Since the parameters change with time, recent measurements are given more weight than past measurements. A tuning gain is used to determine how much weight is given to past measurements and how much is given to recent measurements. If more weight is given to recent measurements, the parameter estimates can change faster but are more susceptible to noise. The effect of noise can be reduced by increasing the weight of past measurements, but this change results in the parameter estimates changing slower.

The estimation algorithm used is based on the algorithm presented by Guay [45]. It is a variation of RLS which includes some additional filters and predictors to improve tracking of time varying parameters. Since the average cost is unknown, the parameter estimation instead uses the estimate $\widehat{J}_i$. The parameter estimation update law is be driven by the estimation error

$$e_i[k] = \widehat{J}_i[k] - \widehat{y}_i[k] \tag{4.18}$$

which compares the total cost estimates obtained by communication and parameter estimation. The total cost predictor $\widehat{y}_i$ uses the parameter estimates by

$$\begin{aligned} \widehat{y}_i[k+1] = \widehat{y}_i[k] + \widehat{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\phi}_i[k] + Ke_i[k] \\ + \boldsymbol{w}_i^\top[k+1]\left(\widehat{\boldsymbol{\theta}}_i[k+1] - \widehat{\boldsymbol{\theta}}_i[k]\right) \end{aligned} \tag{4.19}$$

where $\boldsymbol{w}_i \in \mathbb{R}^{n_i+1}$ is a filtered version of $\boldsymbol{\phi}_i$. Its dynamics are

$$\boldsymbol{w}_i[k+1] = \boldsymbol{w}_i[k] + \boldsymbol{\phi}_i[k] - K\boldsymbol{w}_i[k]. \tag{4.20}$$

Using $\boldsymbol{w}_i$ and e_i, the parameter update law is

$$\boldsymbol{\Sigma}_i[k+1] = \alpha\boldsymbol{\Sigma}_i[k] + \boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k] \tag{4.21}$$

$$\widehat{\boldsymbol{\theta}}_i[k+1] = \mathrm{Proj}_{\gamma_\theta}\left(\widehat{\boldsymbol{\theta}}_i[k] + \frac{\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]\left(e_i[k] - \widehat{\eta}[k]\right)}{\alpha + \boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]}\right) \tag{4.22}$$

where $\mathrm{Proj}_{\gamma_\theta}(\cdot)$ denotes a Lipschitz projection onto a ball of radius γ_θ. This projection ensures that $||\widehat{\boldsymbol{\theta}}_i[k]|| \leq \gamma_\theta$ for all $k \in \mathbb{Z}_{\geq 0}$. Note that the update law (4.21)–(4.22) is different than the law used by Guay [45] as it does not require a perturbation of $\sigma_1\boldsymbol{I}$ to be added to the covariance matrix. This term was required in [45] to ensure invertibility of $\Sigma[k]$ for all $k > 0$; however, as is shown in Lemma 4.4, the covariance matrix update law in (4.21) ensures invertibility of $\Sigma[k]$ without requiring a perturbation. The parameter estimate update law (4.22) has also been modified in a corresponding way. By reducing the need for the perturbation $\sigma_1\boldsymbol{I}$, this parameter update law achieves convergence to a smaller neighbourhood than in [45].

The parameter $\widehat{\eta}_i[k]$ is an estimate of the auxiliary variable $\eta_i[k] = e_i[k] - \boldsymbol{w}_i^\top[k]\widetilde{\boldsymbol{\theta}}_i[k]$. Its dynamics are

$$\widehat{\eta}_i[k+1] = \widehat{\eta}_i[k] - K\widehat{\eta}_i[k]. \tag{4.23}$$

This parameter estimation algorithm relies on two parameters K and α which should both be chosen between 0 and 1 to ensure that $\boldsymbol{w}_i$, $\widehat{\eta}_i$, and $\boldsymbol{\Sigma}_i$ are stable.

4.3.3 Proportional-integral extremum-seeking control

Each agent implements an identical proportional-integral extremum-seeking controller which is based on a gradient descent using the parameter estimate $\widehat{\boldsymbol{\theta}}_i$. The PI ESCs have the form

$$\boldsymbol{u}_i[k] = -K_g\widehat{\boldsymbol{\theta}}_{1,i}[k] + \widehat{\boldsymbol{u}}_i[k] + \boldsymbol{d}_i[k] \tag{4.24}$$

$$\widehat{\boldsymbol{u}}_i[k+1] = \widehat{\boldsymbol{u}}_i[k] - \frac{1}{\tau_I}\widehat{\boldsymbol{\theta}}_{1,i}[k] \tag{4.25}$$

where $\boldsymbol{d}_i$ is a dither signal, K_g is the proportional gain and τ_I is the integral time constant. These parameters can be tuned to affect the stability of the PI ESC and to improve its transient performance. The dither signals are chosen to be sinusoids with amplitude D and frequencies $\omega_1, \ldots, \omega_n$ which are unique primes multiplied by some common rational factor. This choice of dither is sufficient to provide the persistence of excitation necessary for parameter estimation.

Assumption 4.5 (Persistence of excitation)**.** The dither signals $\boldsymbol{d}_i\,(t)$ have been chosen to be sinusoids with frequencies $\omega_1, \ldots, \omega_n$ such that no frequencies are integer multiples of each other or the sum of two other frequencies. This assumption implies that the dithers are persistently exciting so that for some $T \in \mathbb{Z}_{>0}$ and $\gamma_u^-, \gamma_u^+ \in \mathbb{R}_{>0}$, we have that

$$\gamma_u^- \boldsymbol{I} < \frac{1}{T}\sum_{j=k}^{k+T-1} \boldsymbol{\phi}_i[j]\boldsymbol{\phi}_i^\top[j] < \gamma_u^+ \boldsymbol{I}. \tag{4.26}$$

Since $\boldsymbol{u}_i$ is one component of $\boldsymbol{\phi}_i$, this assumption also implies that

$$\gamma_u^- \boldsymbol{I} < \frac{1}{T}\sum_{j=k}^{k+T-1} \boldsymbol{u}_i[j]\boldsymbol{u}_i^\top[j] < \gamma_u^+ \boldsymbol{I}. \tag{4.27}$$

Since $\boldsymbol{w}_i$ is a filtered version of $\boldsymbol{\phi}_i$ and the two signals are initialized at the same value, this assumption implies that there exist $\gamma_w^-, \gamma_w^+ \in \mathbb{R}_{>0}$ such that

$$\gamma_w^- \boldsymbol{I} < \frac{1}{T}\sum_{j=k}^{k+T-1} \boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j] < \gamma_w^+ \boldsymbol{I}. \tag{4.28}$$

Furthermore, this assumption implies that $\|\boldsymbol{u}[k]\| < \sqrt{p\gamma_u^+ T}$, $\|\boldsymbol{u}_i[k]\| < \sqrt{\gamma_u^+ T}$, $\|\boldsymbol{w}[k]\| < \sqrt{p\gamma_w^+ T}$, and $\|\boldsymbol{w}_i[k]\| < \sqrt{\gamma_w^+ T}$.

In practice, Assumption 4.5 requires that the input signals be bounded and that the dither signals be persistently exciting, which can be achieved using sinusoids of different frequencies. Since bounded inputs are already guaranteed by the projection algorithm in (4.22), the only real requirement of Assumption 4.5 is persistently exciting dithers. While the inequalities in Assumption 4.5 are more detailed than in the equivalent assumption for continuous time systems, Assumption 3.4, the practical requirements are nearly indentical in continuous- and discrete-time. The only additional consideration in discrete-time is that the time step must be considered when choosing dither frequencies as no dither can have a frequency which is a multiple of the sampling frequency.

4.4 Convergence analysis

In this section, we consider the convergence of the distributed extremum-seeking controller described in Section 4.3. When discussing convergence, we are interested in practical stability, rather than asymptotic stability. There are always small perturbations to the system due to the dither signal, and therefore the system is not asymptotically stable. Instead, we are interested in showing that various signals converge to a small set containing the origin. This form of stability is known as practical stability [89].

The control objectives are stabilization and optimization. For stability, we are interested in showing that $\boldsymbol{x}$ converges to a neighbourhood of $\boldsymbol{\pi}(\widehat{\boldsymbol{u}})$. For extremum-seeking, we are interested in showing that $\widehat{\boldsymbol{u}}$ converges to a neighbourhood $\boldsymbol{u}^*$. These objectives are equivalent to showing that $\widetilde{\boldsymbol{x}} = \boldsymbol{\pi}(\widehat{\boldsymbol{u}}) - \boldsymbol{x}$ and $\widetilde{\boldsymbol{u}} = \boldsymbol{u}^* - \widehat{\boldsymbol{u}}$ converge to a neighbourhood of zero. While the control objectives do not require parameter convergence or convergence of consensus, we will also consider their convergence.

Before proving the main theorem, we prove several lemmas pertaining to the boundedness of various signals. The results of these lemmas are useful for proving the main theorem.

Since the system is nonlinear, the controller cannot exploit any mathematical structure, and instead relies on a simple, local model generated from recent data. For this approach to be effective, the system cannot change too fast as otherwise previous data would not be relevant. To bound changes in the state, we use the Lipschitz bounds from Assumption 4.3. Changes in the state also depend on the control action. The controller stabilizes and optimizes the system by reducing $\widetilde{\boldsymbol{x}} = \boldsymbol{\pi}(\widehat{\boldsymbol{u}}) - \boldsymbol{x}$ and $\widetilde{\boldsymbol{u}} = \boldsymbol{u}^* - \widehat{\boldsymbol{u}}$. The magnitude of the control depends on the tuning parameters K_g, τ_I, and D. Therefore we would like to bound $\|\Delta \boldsymbol{x}\|$ in terms of $\|\widetilde{\boldsymbol{x}}\|$, $\|\widetilde{\boldsymbol{u}}\|$, K_g, $\frac{1}{\tau_I}$, and D.

Lemma 4.1. *For the system* (4.1)–(4.3) *with controller* (4.17)–(4.25)*, there exists constants* $c_1, \ldots, c_4 \in \mathbb{R}_{>0}$ *such that*

$$\|\Delta \boldsymbol{x}[k+1]\| \leq c_1 \|\widetilde{\boldsymbol{x}}[k]\| + c_2 \|\widetilde{\boldsymbol{u}}[k]\| + c_3 K_g + c_4 D.$$

Proof. See Appendix B. □

The controller does not measure $\Delta \boldsymbol{x}$, but instead relies on its parameter estimates. Therefore, it is also useful to bound $\|\Delta \boldsymbol{\theta}_i\|$. Using the definition of $\boldsymbol{\theta}_i$ (4.5)–(4.6) and the result of Lemma 4.1, we can bound $\|\Delta \boldsymbol{\theta}_i\|$ in terms of $\|\widetilde{\boldsymbol{x}}\|$, $\|\widetilde{\boldsymbol{u}}\|$, K_g, $\frac{1}{\tau_I}$ and D.

Lemma 4.2. *For the system* (4.1)–(4.3) *with controller* (4.17)–(4.25)*, there exists constants* $c_6, \ldots, c_{10} \in \mathbb{R}_{>0}$ *such that*

$$\|\Delta\boldsymbol{\theta}_i[k+1]\| \leq c_6 \|\widetilde{\boldsymbol{x}}[k]\| + c_7 \|\widetilde{\boldsymbol{u}}[k]\| + c_8 K_g + c_9 \frac{1}{\tau_I} + c_{10} D$$

for every $i \in \{1, \ldots, p\}$*.*

Proof. See Appendix C. □

It is also useful to bound the consensus error $\widetilde{J}_i = \widehat{J}_i - \frac{1}{p} J$. Since the controllers are trying to minimize J but can only use $\widehat{J}_i$, we would like to show that the errors between the average cost and the estimates of it are small. Therefore we would also like to bound $\|\widetilde{J}_i\|$ in terms of $\|\widetilde{\boldsymbol{x}}\|$, $\|\widetilde{\boldsymbol{u}}\|$, K_g, $\frac{1}{\tau_I}$ and D. This result is based on the results from [66].

Lemma 4.3. *For the system* (4.1)–(4.3) *with controller* (4.17)–(4.25)*, there exists constants* $c_{11}, \ldots, c_{14} \in \mathbb{R}_{>0}$ *and* $N \in \mathbb{Z}_{>0}$

$$\left\|\Delta\widetilde{J}_i[k+1]\right\| \leq c_{11} \|\widetilde{\boldsymbol{x}}[k]\| + c_{12} \|\widetilde{\boldsymbol{u}}[k]\| + c_{13} K_g + c_{14} D$$

for every $i \in \{1, \ldots, p\}$ *whenever* $k > N$*.*

Proof. See Appendix D. □

The parameter update law (4.22) involves the inversion of the covariance matrix $\boldsymbol{\Sigma}_i$. We must ensure that $\boldsymbol{\Sigma}_i$ remains invertible for all time. It is always initialized as a positive definite—and thus invertible—matrix $\sigma_0 \boldsymbol{I}$. It is updated according to (4.21) by scaling it by $\alpha \in (0, 1)$ and then adding a positive semidefinite term $\boldsymbol{w}_i \boldsymbol{w}_i^\top$.

Since the sum of a positive definite and positive semidefinite matrix is always positive definite, the covariance matrix is always positive definite; however the smallest eigenvalue could get arbitrarily small leading to poor conditioning of numerical results. Therefore it is useful to provide a lower bound for the smallest eigenvalue of $\boldsymbol{\Sigma}_i$. While we are bounding $\boldsymbol{\Sigma}_i$, it is also useful to find an upper bound for its largest eigenvalue.

Lemma 4.4. *The covariance matrix from* (4.21) *is positive definite for all time. Furthermore, there exists* $\gamma_\Sigma^-, \gamma_\Sigma^+ \in \mathbb{R}_{>0}$ *such that*

$$\gamma_\Sigma^- \boldsymbol{I} < \boldsymbol{\Sigma}_i[k+1] < \gamma_\Sigma^+ \boldsymbol{I}.$$

Proof. See Appendix E. □

At this point, we have provided enough bounds and we can state the main theorem. This theorem concerns the practical stability of the system and simultaneously considers parameter estimation, stabilization to the steady-state manifold, and extremum-seeking to find the optimum steady-state cost. The theorem states that these objectives are achieved in a practical sense, with all variables converging to a neighbourhood of their true values. Furthermore, this neighbourhood can be made arbitrarily small by choice of the tuning parameters K_g, τ_I, D and Δt.

Theorem 4.1. *Consider a nonlinear discrete-time system,* (4.1)–(4.3)*, under the control of the distributed extremum-seeking controller,* (4.17)–(4.25)*. Suppose Assumptions 4.1–4.5 hold. Then for* Δt *sufficiently small, there exists consensus gains* $\kappa_P, \kappa_I \in \mathbb{R}_{>0}$*, estimation parameters* $K, \alpha \in (0,1)$*, PI tuning parameters* $K_g, \tau_I \in \mathbb{R}_{>0}$*, and dither amplitude* D*, such that* $\widetilde{\boldsymbol{J}}, \widetilde{\boldsymbol{\theta}}, \widetilde{\boldsymbol{\eta}}, \widetilde{\boldsymbol{x}}$*, and* $\widetilde{\boldsymbol{u}}$ *converge to an* $\mathcal{O}(K_G + \frac{1}{\tau_I} +$

D)-neighbourhood of the origin.

Proof. See Appendix F. □

4.5 Simulation examples

We consider two simulation examples that demonstrate the effectiveness of this technique: a large-scale linear system and a smaller scale nonlinear system.

The first system is a large system with 100 states, 100 inputs, and 100 local costs, which is controlled by a network of 100 agents. This example is designed to show the power of the consensus algorithm. The agents communicate over a very sparse network; the information that each agent requires is only measured by the agent which is farthest away from it. Therefore this information must be shared through the entire network before it can be used for control. To reduce the simulation time, this example uses linear dynamics and quadratic costs although the controllers are generic and do not rely on the linearity of the dynamics. Distributed ESC is very effective and finds the optimum of this system in under 100 seconds (10,000 time steps).

The second system shows that this technique is also applicable to nonlinear systems. The second example involves only five agents controlling a system whose dynamics involve five states. This system has nonlinear dynamics and costs which are not quadratic. Again, the technique is effective, finding the optimum within 40 seconds. This simulation example demonstrates the validity of this technique for nonlinear systems and shows that the controller does not require any model of the system.

4.5.1 100 agent LQ system

Consider the discretization of a continuous-time stable linear system with dynamics

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + (\boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k])\,\Delta t$$

where $\boldsymbol{x}, \boldsymbol{u} \in \mathbb{R}^{100}$. The matrices are defined by

$$\boldsymbol{A} = \begin{bmatrix} -1 & 1/2 & 0 & \dots & 0 \\ 0 & -1 & 1/2 & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & -1 & 1/2 \\ 0 & 0 & \dots & 0 & -1 \end{bmatrix}$$

and $\boldsymbol{B} = \boldsymbol{I}_{100\times100}$. This system is control-affine and can be represented in the form of (4.1). Since this system is linear, it is globally Lipschitz and therefore meets Assumption 4.3. $\boldsymbol{A}$ has 100 eigenvalues at -1, so the discretized system has 100 poles at $-\Delta t$. Therefore when $\Delta t < 1$, the system is stable and meets Assumption 4.2.

The local cost functions for this system are

$$\begin{aligned} y_1 &= (x_{51} - 1)^2 & y_{51} &= (x_1 - 51)^2 \\ y_2 &= (x_{52} - 2)^2 & y_{52} &= (x_2 - 52)^2 \\ &\vdots & &\vdots \\ y_{50} &= (x_{100} - 50)^2 & y_{100} &= (x_{50} - 100)^2 . \end{aligned}$$

The local costs are all quadratic and convex. Since the global cost is the sum of the local costs, it is convex and meets Assumption 4.1. Furthermore, since each state only appears in one quadratic term in the global cost, the optimum state can easily be determined by inspection to occur at $\boldsymbol{x}^* = (51, 52, \dots, 100, 1, 2, \dots, 50)$ which results in a total cost of $J^* = 0$. Since $\boldsymbol{A}$ is invertible, there is exactly one optimal input at $\boldsymbol{u}^* = \boldsymbol{A}^{-1}\boldsymbol{x}^*$.

This system is controlled by 100 agents each implementing the discrete-time distributed extremum-seeking control technique we have described. Each agent communicates with exactly two other agents. The network structure can be described by a cycle graph with 100 nodes. Since cycle graphs are connected, the controller satisfies Assumption 4.4. The Laplacian associated with this graph is

$$\boldsymbol{L} = \begin{bmatrix} 2 & -1 & 0 & 0 & \dots & 0 & -1 \\ -1 & 2 & -1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 2 & -1 & \ddots & \vdots & \vdots \\ 0 & 0 & -1 & 2 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & -1 & 0 \\ 0 & 0 & \dots & 0 & -1 & 2 & -1 \\ -1 & 0 & \dots & 0 & 0 & -1 & 2 \end{bmatrix}.$$

This system can be visualized as 100 agents arranged in a circle (Figure 4.2). Each agent communicates only with its two neighbours; however it measures a cost based on the state associated with the agent directly across the circle from it. Therefore each agent relies heavily on the communication network to find its optimal input. For example, consider agent 1. It manipulates u_1 which only directly affects x_1. The only agent whose cost depends on x_1 is agent 51. Agents 1 and 51 are as far away

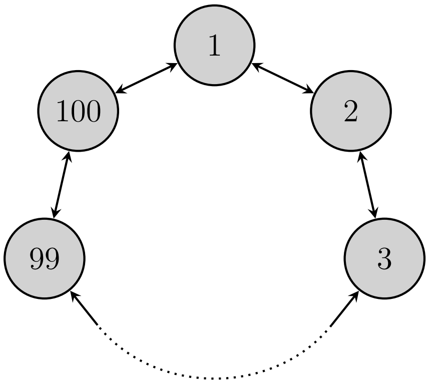


Figure 4.2: Communication graph for the 100 agents controlling the stable linear discrete-time system. This graph has the structure of a cycle.

Table 4.1: Tuning parameters used by the agents implementing discrete-time distributed ESC for the 100-agent linear system.

Δt	κ_P	κ_I	K	α	τ_I	K_g	D
0.01	10	5	0.8	0.8	1	10	0.1

from each other as possible with the shortest path between them having length 50. Therefore there is a delay of 50 time steps between when agent 1 makes an action and when it receives any information about how that action affected the system. This delay makes this system difficult to optimize.

Each agent uses identical tuning parameters (Table 4.1) with dither frequencies which are 100 consecutive primes all multiplied by the same rational scaling factor. This choice of dither frequencies is used to satisfy Assumption 4.5. The controllers are easy to initialize with $\widehat{y}_i[0] = \widehat{J}_i[0] = y_i[0]$, $\mathbf{\Sigma}_i[0]$ equal to the identity, and all other controller variables initialized at zero (Table 4.2).

The distributed ESC is able to quickly find the optimum of the total cost (Figure 4.3). Within 20 seconds, most of the local costs are close to zero. After 100 seconds, all local costs are very close to zero, and the total cost is also very close

Table 4.2: Initial conditions for the 100-agent stable linear system controlled by discrete-time distributed ESC.

$\widehat{y}_i[0]$	$\widehat{J}_i[0]$	$\rho_i[0]$	$\boldsymbol{w}_i[0]$	$\widehat{\eta}_i[0]$	$\widehat{\boldsymbol{\theta}}_i[0]$	$\boldsymbol{\Sigma}_i[0]$
$y_i[0]$	$y_i[0]$	0	$[0,0]^\top$	0	$[0,0]^\top$	$\boldsymbol{I}_{2\times 2}$

to zero. The optimal value of the state is $\boldsymbol{x}^* = (51, 52, \ldots, 100, 1, 2, \ldots, 50)$. At $\boldsymbol{x}^*$, there should be one state at each integer between 1 and 100. The state does not quite reach this point after 100 seconds; however, when the simulation runs slightly longer, we can see that it reaches this optimum within 300 seconds. Similarly, the inputs converge to the optimal value with 300 seconds.

4.5.2 5 agent nonlinear system

Consider a discretization of a continuous-time nonlinear system with dynamics

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + \boldsymbol{F}(\boldsymbol{x}[k], \boldsymbol{u}[k])\Delta t$$

where the nonlinear vector field is given by

$$\begin{aligned}
F_1(\boldsymbol{x}, \boldsymbol{u}) &= -x_1 + x_3 x_5 + u_1 \\
F_2(\boldsymbol{x}, \boldsymbol{u}) &= -2x_2 + \sin(x_1) + \left(1 - \frac{x_4}{10}\right) u_2 \\
F_3(\boldsymbol{x}, \boldsymbol{u}) &= -x_3^3 + \frac{1}{x_2^2 + 1} + u_3 \\
F_4(\boldsymbol{x}, \boldsymbol{u}) &= -2x_4 + \log(x_3^2 + 1) - u_4 \\
F_5(\boldsymbol{x}, \boldsymbol{u}) &= -5 \arctan\left(\frac{x_5}{5}\right) + x_4 + \frac{1 + \cos\left(\frac{x_3}{10}\right)}{2} u_5.
\end{aligned}$$

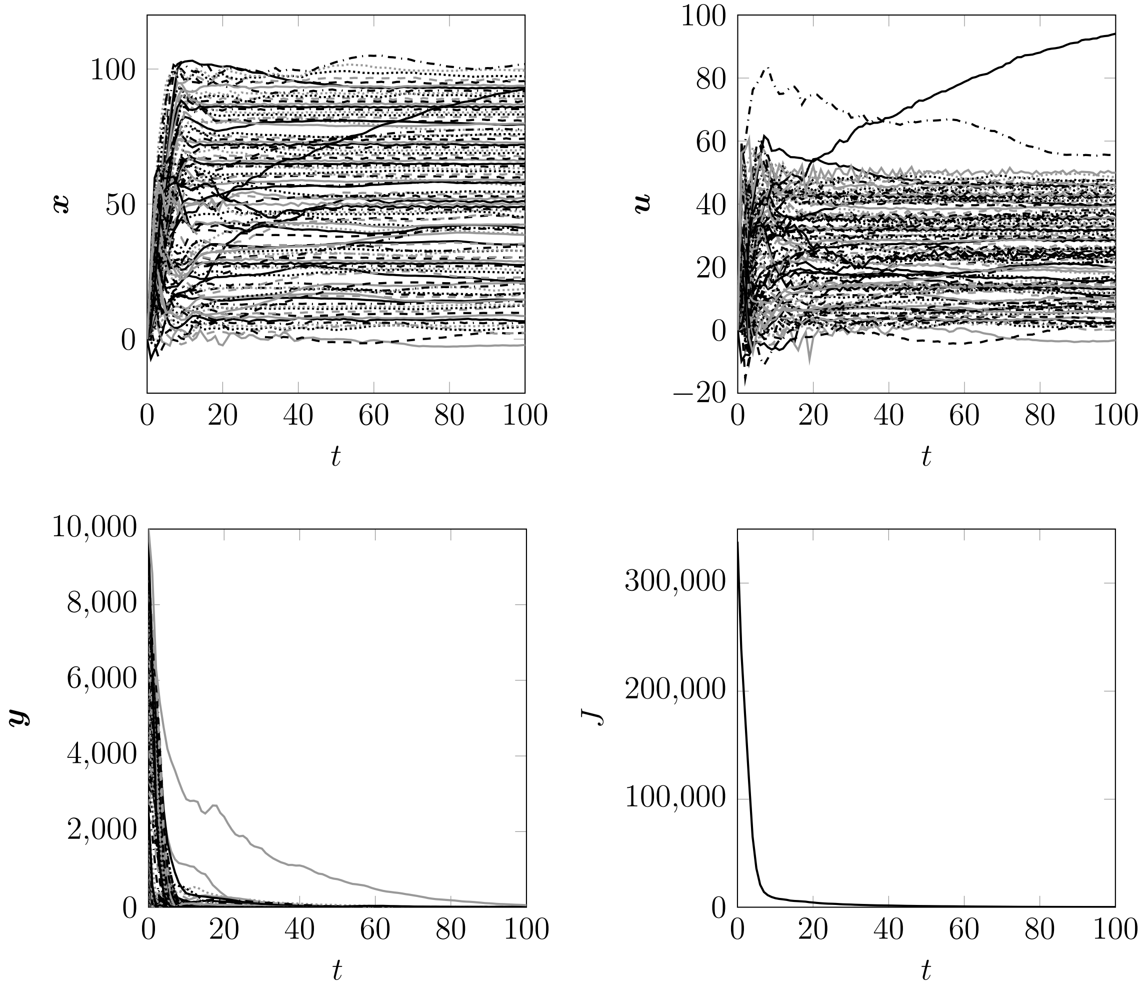


Figure 4.3: State, input, local cost, and total cost trajectories for the LQ discrete-time system under control by 100 agents implementing distributed ESC.

Note that this vector field is control-affine as required. It can also be shown that the vector fields are Lipschitz and the system is stabilizable, so this example meets Assumption 4.3 and Assumption 4.2.

Table 4.3: Tuning parameters used by the agents implementing discrete-time distributed ESC for the 5-agent nonlinear system.

Δt	κ_P	κ_I	K	α	τ_I	k_g	D
0.05	2	1	0.8	0.8	8	1	0.1

Table 4.4: Initial conditions for the 5-agent nonlinear system controlled by discrete-time distributed ESC.

$\widehat{y}_i[0]$	$\widehat{J}_i[0]$	$\rho_i[0]$	$\boldsymbol{w}_i[0]$	$\widehat{\eta}_i[0]$	$\widehat{\boldsymbol{\theta}}_i[0]$	$\boldsymbol{\Sigma}_i[0]$
$y_i[0]$	$y_i[0]$	0	$[0,0]^\top$	0	$[0,0]^\top$	$\boldsymbol{I}_{2\times 2}$

This system's local cost functions are

$$y_1[k] = (x_1[k]-4)^2 - x_3[k] \qquad y_4[k] = \cosh(x_4[k]+2)$$

$$y_2[k] = (x_2[k]-3)^4 \qquad y_5[k] = (x_5[k]-8)^2 + x_3[k].$$

$$y_3[k] = (x_3[k]+3)^2$$

These local costs are not convex because y_1 and y_5 contain linear terms, but the total cost function is indeed convex as the linear terms cancel eachother and therefore Assumption 4.1 is satisfied. The optimum of the cost function occurs at $\boldsymbol{x}^* = (4, 3, -3, -2, 8)$ which results in a total cost of $J^* = 0$.

This system is controlled by 5 agents each implementing the discrete-time distributed ESC with identical tuning parameters (Table 4.3). The dither frequencies are the consecutive primes 11, 13, 17, 19, and 23 which results in the input signals meeting Assumption 4.5. Again, the controller is initialized with most variables equal to zero (Table 4.4).

The agents communicate over a network which can be described by a path graph

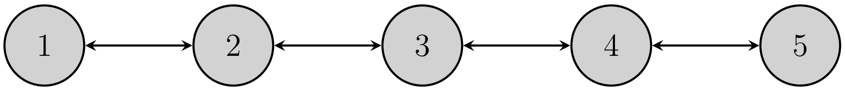


Figure 4.4: Communication graph for the 5 agents controlling the nonlinear discrete-time system. This graph has the structure of a path which is a type of tree.

(Figure 4.4), which results in the Laplacian matrix

$$\boldsymbol{L} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

Since path graphs are connected, this network satisfies Assumption 4.4.

The simulation results show that the distributed ESC is able to find the optimum of this nonlinear system within 40 seconds (Figure 4.5). Note that some of the states experience some overshoot and undershoot. These kinds of trajectories are common with nonlinear systems. The controller is based on a locally linear parameterization of the dynamics. Since the system is nonlinear, this parameterization changes when the state changes and is therefore time-varying. When the parameterization changes rapidly, the parameter estimation algorithm can take a while to adjust, resulting in some poor transient performance. By tuning α, it is possible to change this performance. When α is close to zero, historic data have less weight, which results in parameter estimates updating faster, but they are more susceptible to noise. When α is close to 1, the parameter estimates are more robust to signal noise, but they do not update as fast.

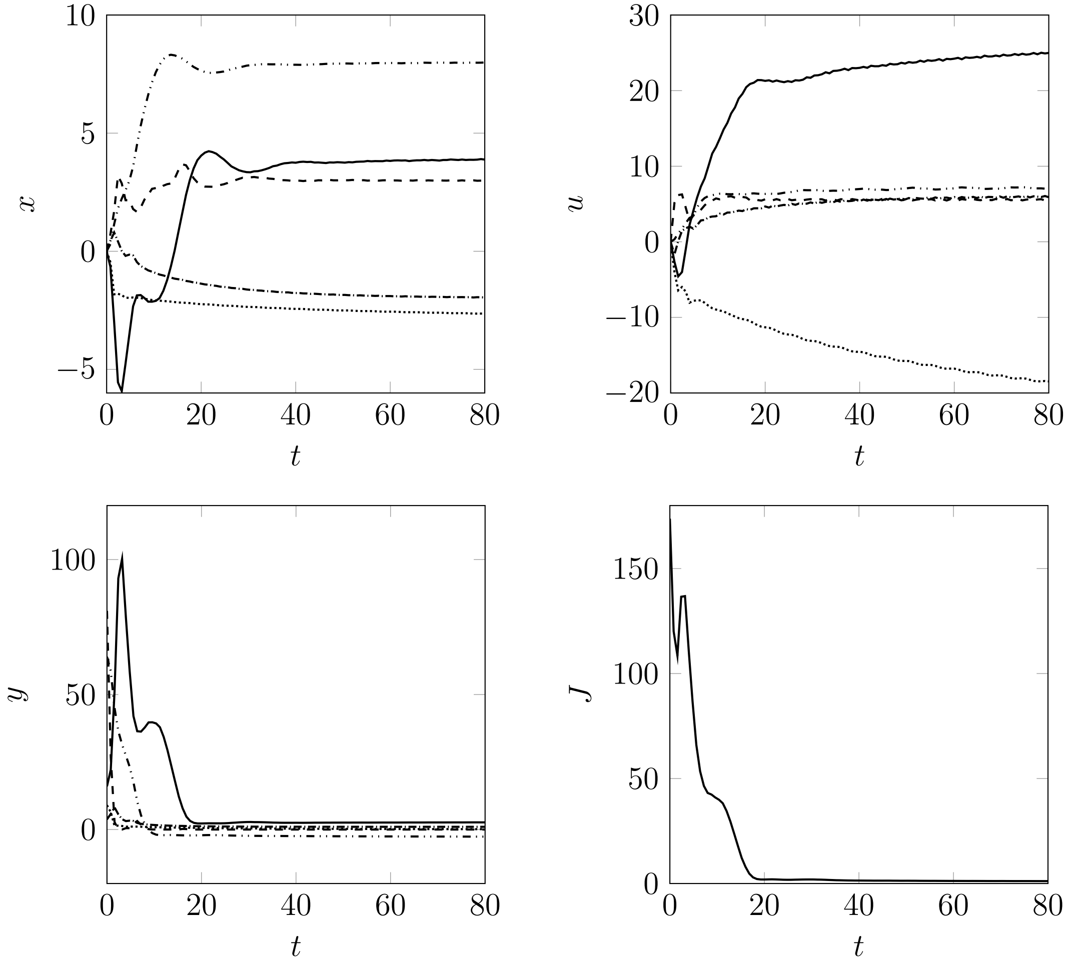


Figure 4.5: State, input, local cost, and total cost trajectories for the nonlinear discrete time system under control by 5 agents implementing distributed ESC.

4.6 Conclusion

In this chapter, a discrete-time implementation of distributed ESC was developed. The approach—similar to the continuous-time approach—uses a consensus algorithm to coordinate several agents implementing a local ESC. A discrete-time dynamic average consensus algorithm can provide a convergent total cost estimate whenever the first differences are bounded which, by design of the ESC, they always are. Each agent uses an RLS-type parameter estimation routine to estimate the time-varying total cost gradient and then minimizes the total cost by a PI gradient descent. The stability of the overall system is dependent on the choice of time step. For a small enough time step, discrete-time distributed ESC is able to stabilize slowly unstable nonlinear discrete-time systems with unknown dynamics, and is able to optimize an unknown, but measured cost function.

Chapter 5

Formation control of high-altitude balloons

In this chapter, we examine one potential application of distributed ESC: the formation control of high-altitude balloons for wireless internet. Section 5.1 introduces the technology and explains why formation control is important. Section 5.2 outlines the requirements for the controller of these balloons. Several dynamic models are developed in Section 5.3 and cost functions are designed in Section 5.4. A discussion of the application of distributed ESC to this problem can be found in Section 5.5. Continuous- and discrete-time simulations are provided in Section 5.6.

5.1 Introduction

Widespread internet access can help many countries improve in diverse areas such as education, health care, agriculture, and manufacturing. As of May 2016, only 46.1% of people have access to the internet [58]. In Europe and North America more than three quarters of people have internet access; however, the two most populous continents of Asia and Africa have the lowest internet penetration rates with less than half of the people being connected (Figure 5.1). These two continents consist of many

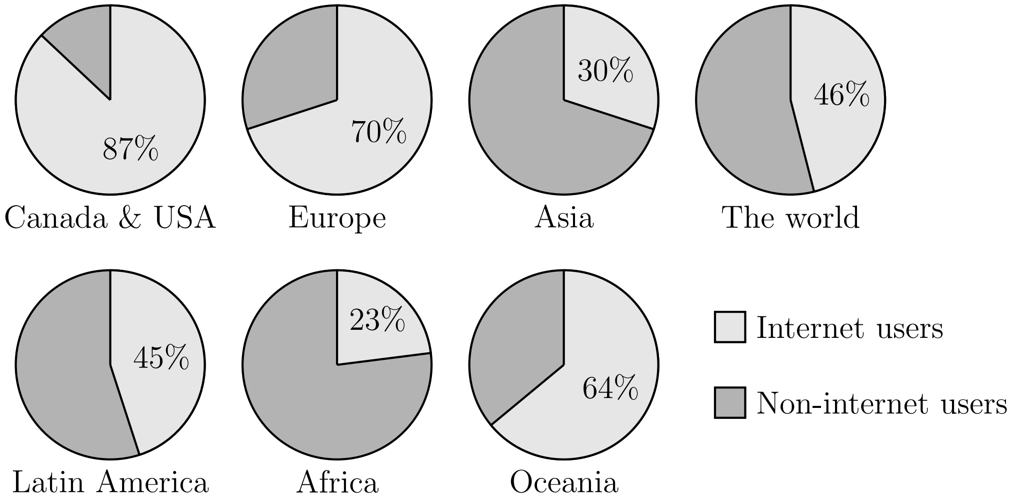


Figure 5.1: Fraction of people with internet access in each geographical region on Earth. Data from [58].

emerging countries and represent the largest two regions for increased internet use. The relative lack of internet use in emerging countries is typically due to prohibitive costs associated with existing internet technologies. To connect the entire world, new technology is required which reduces the financial barriers to internet access in geographically isolated and poor regions.

The past fifty years have seen a doubling of the Earth's population and a recent explosion in the number of internet users (Figure 5.2). Prior to the 1990s, very few people had internet access; however in the past twenty years, many widely used internet technologies have been developed, enabling almost half of all people to connect to the internet. To put this growth in perspective, in 2016 there are 3.4 billion internet users, equal to the entire population of Earth in 1967. Each year the number of people not connected to the internet decreases despite consistent increases in Earth's population. While it may seem that this upward trend is inevitable and will eventually result in an equal number of people as internet users, this trend cannot

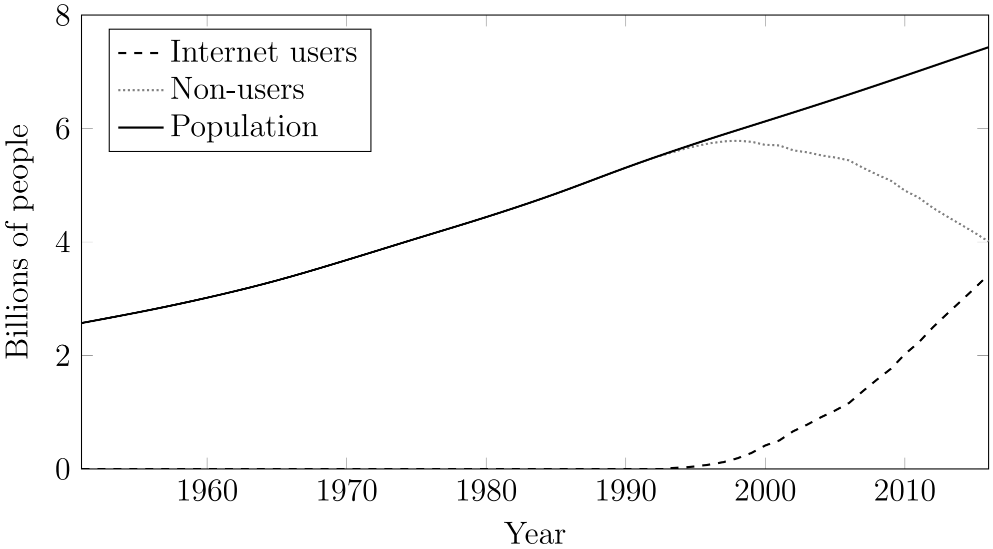


Figure 5.2: Global population and number of internet users from 1951–2016. Population data from [136] and internet user data from [58].

continue without new technology more suited to providing internet access to people in emerging countries.

Current broadband internet technologies include digital subscriber line (DSL), cable modem, fiber, broadband over powerline (BPL), wireless, and satellite [37]. DSL, cable modem, fiber, and BPL are all ground-based technologies which require the installation of physical wires. Wires can be expensive to install, especially for remote areas with low population densities. Current wireless technologies, such as WiFi and LTE, rely on wireless access points around Earth's surface which are costly to install and can only operate over short distances as the signals are attenuated by obstacles on Earth's surface. Again, costs are prohibitive for widespread wireless internet use in emerging countries with current technology. Satellite internet is mostly used in remote locations where ground-based internet technologies are not practical. Unfortunately, satellite internet is the most expensive internet technology that is widely used today, and is therefore only practical for remote areas of developed countries. Furthermore, as low Earth orbit satellites orbit the Earth at 1200 km altitude, satellite internet has noticeable latency of 40–400 ms, which makes it inconvenient to use when other, faster technologies are available [54]. Since wired, wireless, and satellite internet are not suitable in emerging countries, none of these technologies are well suited for expanding the internet to be truly global.

Several large tech companies have plans to create a global internet using new air-based technologies. SpaceX and Virgin-backed OneWeb both hope to use satellites orbiting at much lower altitudes than current satellite internet uses [135]. Facebook's connectivity labs plans to use drones that actively fly around the troposphere [59]. Google's Project Loon wants to use high-altitude balloons which float passively in the stratosphere [44]. The common trait shared by all of these prospective technologies is that they consist of a network mobile agents moving through Earth's atmosphere and connecting many internet users wirelessly. Since they are in the air, there are fewer obstacles between the wireless source and receiver, resulting in wireless communication over much larger distances. Furthermore since each of these networks would cover the entire Earth, they could drastically reduce the cost of providing internet to emerging countries and help create a world where everyone has internet access everywhere.

High-altitude balloons are a very promising technology for global internet. They float in the stratosphere at an altitude of 15–50 km, which is high enough that they do not interfere with birds or airplanes, but is still low enough to provide a high-speed connection without noticeable latency. In the stratosphere, wind currents are much smoother and more predictable than in the troposphere (Figure 5.3). Drones flying in the troposphere need to actively fly against the chaotic wind currents found near Earth's surface, requiring significant amounts of energy. Balloons float passively on smooth stratospheric wind currents, requiring very little energy, only needing motors to power the pumps that maintain the balloon's pressure and control its altitude. Additionally, since balloons float above the clouds, they receive consistent solar radiation, so their pumps and onboard electronics can be powered completely by solar power, making this technology very environmentally friendly.

One of the main challenges for a large fleet of high-altitude balloons is formation control. The balloons must be launched from designated launch sites and then navigate into a favourable, spread out formation so that all people on Earth are close to at least one balloon. Balloons float passively on wind currents and are therefore constantly moving so they cannot simply be placed in an optimal formation and left there. The formation of a fleet of balloons is therefore dynamic and must be controlled to maintain an acceptable formation. The balloons are able to control their position by pumping helium in and out of the balloon to change its buoyancy and thus altitude. Since wind currents vary with altitude, balloons can steer into a more favourable formation by changing their altitude.

Successful implementation of a network of balloons requires control algorithms for launch and for maintaining a formation. Google intends on "solving this with some complex algorithms and lots of computing power" [98]. Sniderman *et al.* showed that this problem can be solved without large amounts of computing power [114]. They derived a block-circulant control law which relied on a linear model of the wind. They do not provide a method for dealing with the nonlinearities of real wind currents and only consider the simpler case of balloons on a circle. While effective on a circle, it is

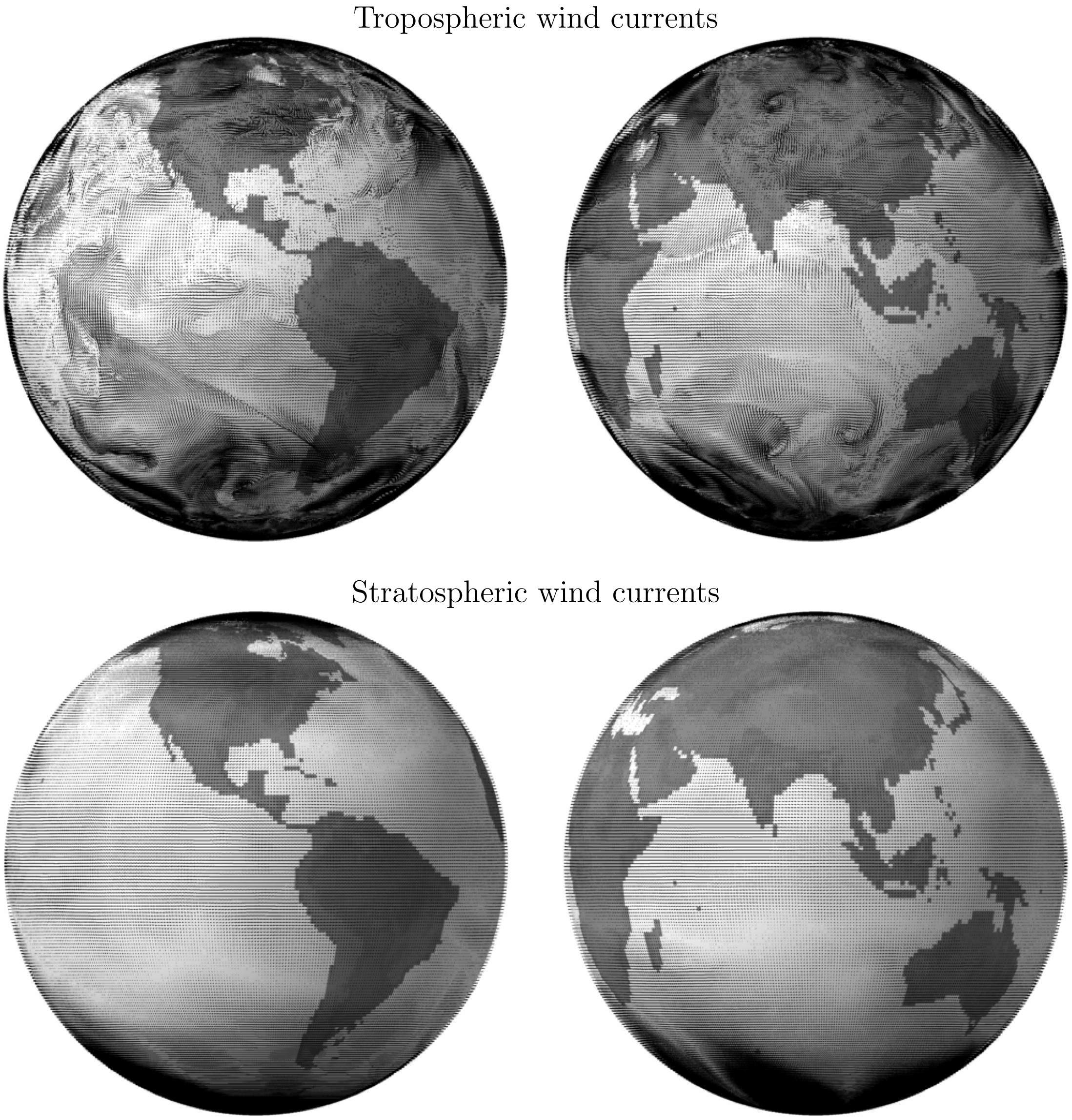


Figure 5.3: Earth's wind currents at its surface (top) and in the stratosphere at 5 kPa (bottom). Data obtained on May 11, 2016 at 10:00 UTC from [95].

unclear that this algorithm could be generalized to a sphere.

In this chapter, we consider the application of distributed ESC to the formation control of high-altitude balloons. In this approach, the distributed agents are the individual balloons. Each balloon must be able to measure some cost which is a function of its own position position, the position of nearby balloons, and the density of nearby internet users. These local cost functions must be designed so that their sum, the total cost function, is minimized when the balloons are in an optimal formation. The balloons can use ESC to minimize the total cost and thereby improve their formation. The design of the local cost functions is one of the most important aspects of successfully implementing distributed ESC. Since ESC is model-free and only requires a measurement of a cost function, it does not require knowledge of the wind currents, thereby drastically reducing the computational burden associated with the algorithms Google intends to use. Despite its computational simplicity, this approach can effectively control the formation of balloons which are subject to realistic wind currents. Because of its computational simplicity, it is a scalable approach can be used for systems of over 1000 balloons.

5.2 Problem definition

Consider a system of p balloons floating above the Earth. The objective is to steer these balloons into a formation which can provide adequate internet coverage for all internet users on Earth. Furthermore, once the balloons reach this formation, they must always remain in a similar formation. Such a formation must meet the following requirements:

1. Each user must be able to connect to at least one balloon at all times,

2. Users must be less than a minimum distance away from a balloon to connect to it.

These requirements are met when there are sufficiently many balloons and the balloons are adequately spread out. Let $\ell_{\max}$ be the maximum distance that a balloon and user can communicate over. Then a balloon located at $q_i \in \mathbb{S}^2$ is capable of communicating with any balloon located in the closed geodesic ball,

$$\mathcal{B}_{\ell_{\max}}(q_i) = \left\{ q \in \mathbb{S}^2 \mid \ell_{\mathbb{G}}\left(q, q_i\right) \leq \ell_{\max} \right\} \tag{5.1}$$

where $\ell_{\mathbb{G}}(\cdot, \cdot)$ is the distance function on $\mathbb{S}^2$ associated with the standard round metric $\mathbb{G} : \mathsf{T}\mathbb{S}^2 \times \mathsf{T}\mathbb{S}^2 \to \mathbb{R}_{\geq 0}$. Intuitively, $\mathcal{B}_{\ell_{\max}}(q_i)$ is balloon i's region of coverage. Since we want every point on Earth to be in at least one balloon's coverage region, the first two requirements are met when $\cup_{i=1}^{p} \mathcal{B}_{\ell_{\max}}(q_i) = \mathbb{S}^2$. Alternatively since we don't need to provide coverage to uninhabited places on Earth, we can use the slightly weaker requirement that the union of the coverage regions contains all regions of Earth where people live.

Google expects to use "thousands of balloons" [115]. With such a large number of balloons, the control algorithm must scale well with p. Ideally, the algorithm should be fully parallelizable, so that each balloon computes its own actions, resulting in an algorithm whose computation time does not depend on p. To coordinate their motion, the balloons must communicate; however the amount of communication should also be kept as small as possible so that the overall algorithm is scalable. The computation and communication requirements can be summarized as:

3. Each balloon must coordinate its own movements; no centralized coordinator

may be used,

4. The control algorithm must rely only a balloon's own measurement of its position, the number of users it is connected to, their bandwidth, and information received from nearby balloons; no complex wind models may be used,

5. Balloons may only communicate with balloons which are less than a maximum distance away.

A distributed control system meets these requirements. In a distributed setting, the system is controlled by p agents which communicate over a network but each plan their own local movements. The last requirement essentially determines the structure of the network which the balloons can communicate over with balloons i and j communicating if $\ell_{\mathbb{G}}(q_i, q_j) \leq \ell_{\max}$. Typically, distributed algorithms require a connected network which occurs if every pair of balloons is connected by a sequence of balloons which are all less $\ell_{\max}$ away from their neighbours. Since balloons tend to be closer as there are more balloons, this requirement is more likely to be met as p increases.

To minimize the amount of energy used by each balloon and maximize their lifespan, balloons are only equipped with pumps to change their buoyancy and are not equipped with any other motors and cannot fly actively. Therefore, we have the following requirement:

6. Each balloon must float passively with the wind current; balloons may not move in any direction in which there is no wind current.

This requirement limits the performance of the network. Since balloons may not remain statically above one location, the formation is dynamic. Users in one location can only connect to any one balloon for a short period of time before that balloon floats away and is replaced by a new balloon. It is not possible to find an optimal formation and remain there due to this drift. Instead, the objective should be for each balloon to constantly try to improve the formation of the network. Therefore a final, soft requirement is:

7. Each balloon should try to position itself in such a way that it is connected to $1/p^{\text{th}}$ of the users so that internet traffic is shared equally between all balloons.

Given these seven requirements, the objective is to design a discrete-time controller which can be implemented on a fleet of balloons and used to control the formation of balloons while satisfying these seven requirements.

5.3 Balloon dynamics model

Each balloon moves in a region of the Earth's atmosphere which has the geometry of a spherical shell. High-altitude balloons are designed to withstand the low temperatures and high radiation levels of Earth's stratosphere but are not suited to other conditions [137]. Therefore each balloon's altitude must be constrained to float in the stratosphere between 10–50 km above sea level. The Earth's radius of 6371 km is much larger than the thickness of stratosphere, so we can neglect the altitude when describing the position of the balloons and when modeling their dynamics.

Under this simplification, the configuration space of each balloon is the two-dimensional sphere $\mathbb{S}^2$. Since there are p balloons, the configuration space of the entire system is

$$\mathsf{Q}_{\text{sys}} = \underbrace{\mathbb{S}^2 \times \cdots \times \mathbb{S}^2}_{p \text{ times}}. \tag{5.2}$$

A point $q \in \mathsf{Q}_{\text{sys}}$ represents the position of all of the balloons in the network. To obtain the position of a single balloon, we define a map $\Pi_i : \mathsf{Q}_{\text{sys}} \to \mathbb{S}^2$ such that $q_i = \Pi_i(q)$ is the position of the i^{th} balloon.

5.3.1 Coordinate charts

Each copy of $\mathbb{S}^2$ requires two coordinate charts to create an atlas. We primarily use the standard longitude-latitude chart which is defined by

$$\lambda = \operatorname{arctan2}(y, x) \tag{5.3}$$

$$\varphi = \arctan\left(\frac{z}{\sqrt{x^2 + y^2}}\right) \tag{5.4}$$

where $\lambda \in (-180, 180)$ is the longitude, $\varphi \in (-90, 90)$ is the latitude, and $\operatorname{arctan2} : \mathbb{R}^2 \setminus \{(0,0)\} \to (-180, 180]$ is the four-quadrant inverse tangent function. As is the convention in geography, we write the longitude and latitude in degrees instead of radians. This chart is a mapping from $\mathsf{U} = \mathbb{S}^2 \setminus \{(x, y, z) \in \mathbb{S}^2 \mid x \leq 0, y < 0\}$ to $(-180, 180) \times (-90, 90)$.

We need a second chart to create an atlas. Other charts can be obtained by rotating the sphere and applying the regular longitude-latitude chart. Since rotation is a linear operator, these charts are all compatible with the standard chart. To get an atlas, we only need one of these charts provided that its domain contains

$\mathbb{S}^2 \setminus \mathsf{U} = \{(x, y, z) \in \mathbb{S}^2 \mid x \leq 0, y < 0\}$. One such chart is

$$\widetilde{\lambda} = \operatorname{arctan2}(-y, z) \tag{5.5}$$

$$\widetilde{\varphi} = \arctan\left(\frac{-x}{\sqrt{y^2 + z^2}}\right). \tag{5.6}$$

This chart is a mapping from $\widetilde{\mathsf{U}} = \mathbb{S}^2 \setminus \{(x, y, z) \in \mathbb{S}^2 \mid x \leq 0, y < 0\}$ to $(-180, 180) \times (-90, 90)$. Since the charts are compatible and $\mathsf{U} \cup \widetilde{\mathsf{U}} = \mathbb{S}^2$, these two charts create an atlas for $\mathbb{S}^2$. While both charts are needed to cover $\mathbb{S}^2$, we perform simulations entirely in the standard longitude-latitude chart.

5.3.2 General nonlinear time-varying model

The balloons' longitude-latitude dynamics depend on nearby wind currents. Since the balloons do not fly actively, we can assume that every balloon's motion only depends on the wind currents at that location. Therefore the balloon velocity is always equal to the wind velocity. Since wind currents vary significantly with altitude, altitude must be considered when describing the wind currents' vector fields. A balloon's altitude depends on the balance between its buoyancy, gravity, and drag. By changing the rate that helium is pumped into or out of the balloon, the balloon can change its buoyant force and thus its altitude. The resulting altitude dynamics are second-order with the pump rate as an input [114]. These altitude dynamics are fast relative to the wind speeds, and it is assumed that each balloon has an adequate altitude controller such that the altitude dynamics can be neglected when considering the longitude-latitude dynamics. Therefore we can treat altitude as an adjustable parameter which is the control input for the longitude-latitude dynamics.

Let $\mathbb{U} = [u_{\min}, u_{\max}]^p$ be the set of allowable altitudes and $\boldsymbol{u} = (u_1, \ldots, u_p)$ where $u_i \in [u_{\min}, u_{\max}]$ is the altitude of the i^{th} balloon. Then we can define a smooth map $F : \mathbb{U} \to \mathfrak{X}(\mathbb{R}, \mathsf{Q}_{\text{sys}})$ which maps each input $\boldsymbol{u}$ to a smooth time-varying vector field $F\{\boldsymbol{u}\} : \mathbb{R} \times \mathsf{Q}_{\text{sys}} \to \mathsf{TQ}_{\text{sys}}$. In this general case, the dynamics of the entire system are

$$\dot{q} = F\{\boldsymbol{u}\}(t, q). \tag{5.7}$$

The vector field $F\{\boldsymbol{u}\}$ is a vector field on the entire system. This vector field consists of p component vector fields on $\mathbb{S}^2$ which describe the dynamics of each balloon. These component vector fields are smooth maps $f : [u_{\min}, u_{\max}] \to \mathfrak{X}(\mathbb{R}, \mathbb{S}^2)$ which map u_i to a smooth time-varying vector field $f\{u_i\} : \mathbb{R} \times \mathbb{S}^2 \to \mathsf{T}\mathbb{S}^2$. For a fixed altitude u_i, $f\{u_i\}$ is the wind currents at that altitude. Since component vector fields represent the wind currents experienced by each balloon, and since each balloon is subjected to the same wind currents, the component vector fields are all equal. Therefore $F\{\boldsymbol{u}\} = (f\{u_1\}, \ldots, f\{u_p\})$ where $f : [u_{\min}, u_{\max}] \to \mathfrak{X}(\mathbb{R}, \mathbb{S}^2)$ represents the wind currents at an altitude u_i. Using this notation, the dynamics of a single balloon are

$$\dot{q}_i = f\{u_i\}(t, q_i). \tag{5.8}$$

This model is the most general model and it can describe any smooth wind current.

We assume that Earth's wind currents are smooth so the maps $f : [u_{\min}, u_{\max}] \to \mathfrak{X}(\mathbb{R}, \mathbb{S}^2)$ and $f\{u_i\} : \mathbb{R} \times \mathbb{S}^2 \to \mathsf{T}\mathbb{S}^2$ are smooth functions of their inputs. Since the vector fields are assumed to be smooth functions of u_i, t and q_i, they must also be locally Lipschitz in all of its arguments. Furthermore, since $[u_{\min}, u_{\max}]$ and $\mathbb{S}^2$ are compact, we can conclude that f is globally Lipschitz in u_i and q_i.

5.3.3 Data-based time-invariant model

To simulate the system, we need to be able to evaluate $f\{u_i\}(t, q_i)$. Closed form expressions for f are not available as wind patterns are complex and unpredictable. Models are generated by numerical integration of fluid flow partial differential equations or using stochastic predictions based on historical weather data [134]. The resulting models are available as gridded data which can be interpolated to evaluate $f\{u_i\}(t, q_i)$ at non-grid points.

The United States' National Oceanic and Atmospheric Adminstration (NOAA) provides free high-resolution weather data online through the NOAA Operational Model Archive and Distribution System (NOMADS) [95]. These data are available on a longitude-latitude-isobaric pressure grid. Our original model was derived using altitude as the input; however, as there exists a diffeomorphism between pressure and altitude, we can simply treat pressure as the input to the model without needing to make any modifications. For simulations, we use data on a 0.25° grid for longitude and latitude at isobaric pressures of 1, 2, 3, 5, 7, and 10 kPa. At non-grid points, we approximate the wind currents by three-dimensional linear interpolation.

5.3.4 Control-affine model

Simulations with realistic wind-data are slow as they require large data sets and interpolation. For basic simulations, we use a simpler time-invariant control-affine model. Let $F_0 : \mathsf{Q}_{\text{sys}} \to \mathsf{TQ}_{\text{sys}}$ and $F_1 : \mathsf{Q}_{\text{sys}} \to \mathsf{TQ}_{\text{sys}}$ be two smooth vector fields. The control-affine system is

$$\dot{q} = F_0(q) + F_1(q)\boldsymbol{u}. \tag{5.9}$$

As before, there exist component vector fields $f_0 : \mathbb{S}^2 \to \mathsf{T}\mathbb{S}^2$ and $f_1 : \mathbb{S}^2 \to \mathsf{T}\mathbb{S}^2$ describing the currents for each balloon. Then for a single balloon,

$$\dot{q}_i = f_0(q_i) + f_1(q_i)u_i. \tag{5.10}$$

This model describes any smooth wind current which varies linearly with altitude.

For simulations, we use a control-affine model which has a simple form in the standard longitude-latitude coordinates. In this model, the wind blows directly east and only depends on latitude. In coordinates, this model is

$$\dot{\lambda}_i = f_0(\varphi_i) + f_1(\varphi_i)u_i \tag{5.11}$$

$$\dot{\varphi}_i = 0. \tag{5.12}$$

This model is too simple to describe the wind currents in the troposphere but can be used for the stratosphere where wind is less affected by the geography of Earth's surface. For example, at 60° south (Figure 5.4) the wind is blowing predominantly east with only minor north-south components. Therefore we model the wind as having no north-south components. Additionally, while there is some longitudinal variation it is smaller than variation due to pressure or latitude and is therefore neglected in the model.

5.4 Network coverage cost model

The main objective of the controller is to spread out all of the balloons to provide good coverage over the entire Earth. Intuitively, a good formation is one where all areas on Earth are close to at least one balloon and a bad formation is one where there

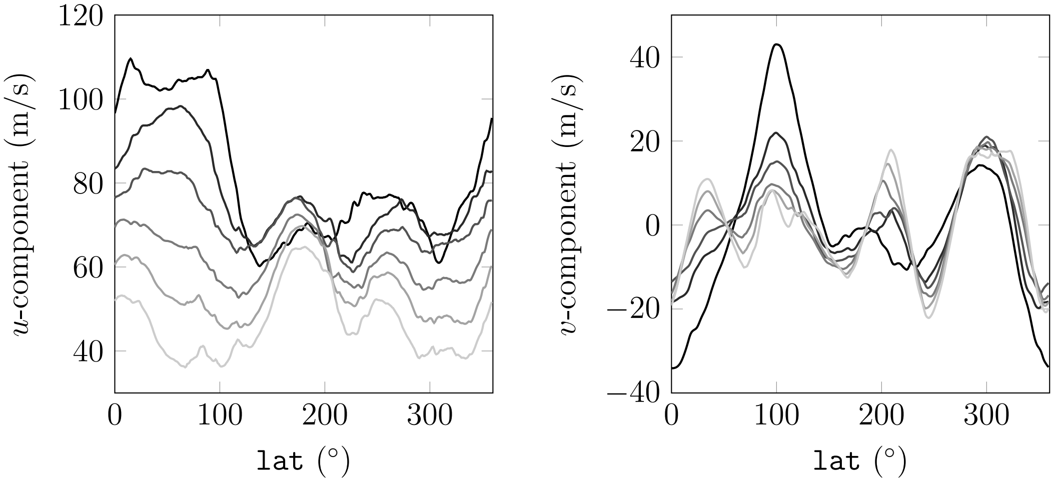
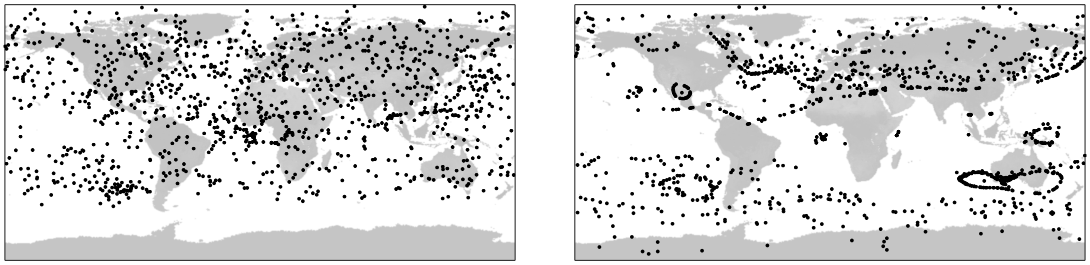


Figure 5.4: The u- (east-west) and v- (north-south) components of wind velocities at 60° south on September 27, 2015 at 2:00 AM UTC using data obtained from NOMADS [95]. Each line corresponds to a different isobar. The data are at 1 (lightest), 2, 3, 5, 7, and 10 Pa (darkest).



Good configuration

Bad configuration

Figure 5.5: Examples of good and bad balloon formations. In the good formation (left), the balloons are spread out in such a way that all inhabited places on Earth are close to at least one balloon. In the bad formation (right), many places are not close to a balloon.

are places on Earth that are far from all of the balloons (Figure 5.5). While it is easy to judge whether not a formation is adequate by eye, we must formalize this notion using a cost function. To formalize this objective, we consider the minimization of a cost function that measures how spread out the balloons are.

In the framework of distributed ESC, we require one local cost per balloon and then try to minimize the total cost which is the sum of the local costs. Each local cost must be a function only of values that a balloon can actually measure. Some variables that can be measured and could be used in a balloon's cost function are:

- It's own position,
- The position of other balloons which are less than a distance of $\ell_{\max}$ away,
- The number of users that it is connected to, and
- The bandwidth of data flowing through it.

Using some or all of these data, the balloon can compute its local cost. These costs must be designed to measure how well-positioned each individual balloon is. For ESC to be effective, we require that the total cost be convex about its optimum. To ensure that the total cost is convex, we simply define all local costs to be convex. We provide several convex functions which can be used in cost functions. Many of these functions require the computation of a Voronoi partition—a useful concept from geometry.

5.4.1 Voronoi partitions

A Voronoi partition is a partition of a set into p disjoint regions $\Gamma_1, \ldots, \Gamma_p$ based on a set of p points $q_1, \ldots, q_p$ [129]. The Voronoi cell associated with q_i is defined by

$$\Gamma_i = \{q \in \mathsf{Q} \mid \ell_{\mathbb{G}}(q, q_i) < \ell_{\mathbb{G}}(q, q_j) \ \forall j \neq i\} \tag{5.13}$$

where $\ell_{\mathbb{G}}(\cdot, \cdot)$ is the distance function obtained from the metric $\mathbb{G}$ on the manifold Q. For our system, $\mathsf{Q} = \mathbb{S}^2$, q_i is the position of the i^{th} balloon, and $\Gamma_i \subset \mathbb{S}^2$ is the regions on Earth where users would connect to balloon i. Based on this definition,

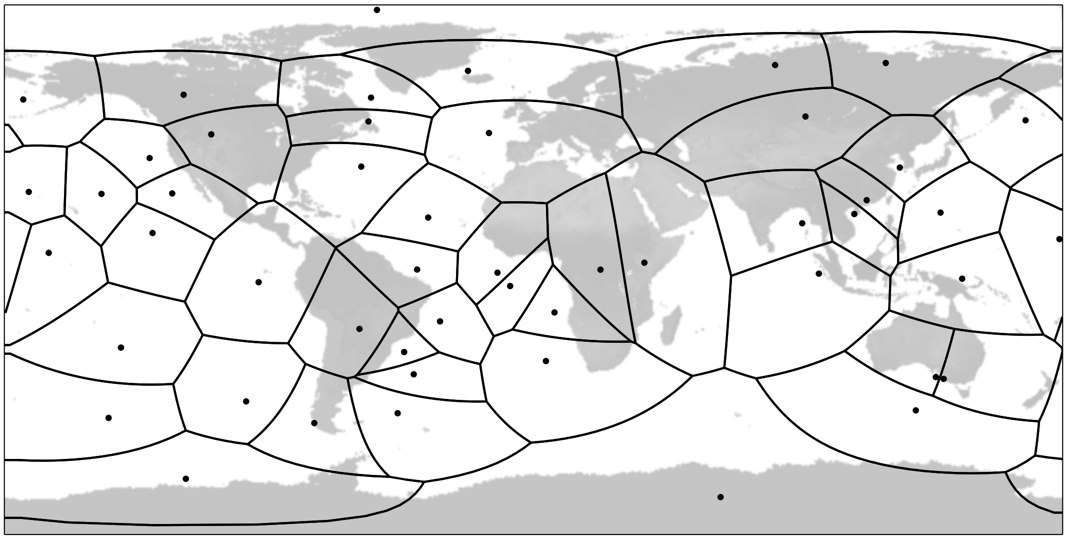


Figure 5.6: Example of a Voronoi partition on the sphere, shown here using the plate carrée projection. Note that the projection distorts the spherical polygons.

$\cup_{i=1}^{n}\Gamma_i = \mathbb{S}^2$, so each place is assigned to at least one balloon, and $\Gamma_i \cap \Gamma_j = \{\}$ whenever $i \neq j$, so each place is assigned to exactly one balloon.

In Euclidean space, the definition of a Voronoi partition results in convex polygonal cells with each each point q_i in the interior of its Voronoi cell Γ_i. On a more general manifold with nonzero curvature, the boundary of each Voronoi cell consists of finitely many geodesic paths. For a sphere such as Earth, the geodesics are great circles and the Voronoi cells are spherical polygons (Figure 5.6).

An analytical represenation of a Voronoi partition on a sphere can be efficiently computed given a set of points [107]. First the Delaunay triangulation—a triangulation which is based on a set of points and tends to avoid narrow triangles—is computed using the convex hull of the points embedded in $\mathbb{R}^3$. The edges of the Delaunay triangulation are the great circle arcs which are obtained by projecting the edges of the convex hull polyhedron onto the surface of the sphere. The Voronoi

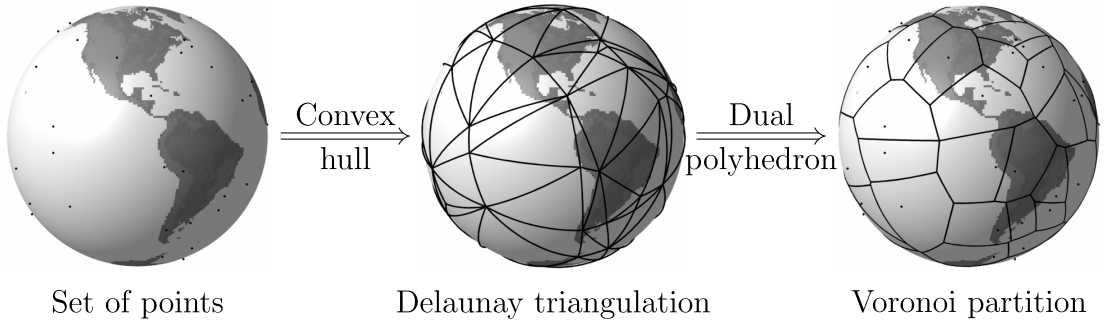


Figure 5.7: Construction of a Voronoi partition from a set of points on the sphere. First the convex hull of the points is taken to obtain the Delaunay triangulation. Then the Voronoi parition is obtained as the dual polyhedron of the Delaunay triangulation.

partition is the dual of the Delaunay triangulation. Therefore the Voronoi partition can be obtained by taking the dual of the convex hull polyhedron and then projecting the edges of the dual polyhedron to the surface of the sphere (Figure 5.7).

In real life, balloons can compute their own Voronoi cell by communicating with nearby balloons. Let $\mathbb{J}_i$ be the index set of balloons which would share an edge with balloon i in the Delaunay triangulation. Suppose that $q_j \in \mathcal{B}_{\ell_{\max}}(q_i)$ for each $j \in \mathbb{J}_i$ and assume that balloon i communicates with every balloon in $\mathcal{B}_{\ell_{\max}}(q_i)$. Then for each $q_j \in \mathcal{B}_{\ell_{\max}}(q_i)$ compute the geodesic path $\gamma_{i,j}$ connecting q_i and q_j. Let $q_{i,j}$ be the midpoint of this path and compute the unique great circle $\overline{\gamma}_{i,j}$ which passes through $q_{i,j}$ such that $\mathbb{G}(\gamma_{i,j}(q_{i,j}), \overline{\gamma}_{i,j}(q_{i,j})) = 0$. This great circle is the perpendicular bisector of the geodesic path connecting balloons i and j. The resulting great circles $\overline{\gamma}_{i,j}$ for $j \in \mathbb{J}_i$ result in a partition of the sphere into several polygonal regions. The Voronoi cell Γ_i is the region of this partition which contains q_i.

If $\{q_j \mid j \in \mathbb{J}_i\} \notin \mathcal{B}_{\ell_{\max}}(q_i)$ then balloon i cannot communicate with all of the balloons that it needs information from to compute Γ_i. In this case, the same procedure can be used to compute $\widehat{\Gamma}_i$, an estimate of Γ_i. While $\widehat{\Gamma}_i$ is not the Voronoi cell, it has the property that $\Gamma_i \subseteq \widehat{\Gamma}_i$. Since the computation of $\widehat{\Gamma}_i$ only requires knowledge of q_i and q_j for $j \in \mathcal{B}_{\ell_{\max}}(q_i)$, each balloon can always calculate its own Voronoi cell, or at least an oversized estimate of it. When $\ell_{\max}$ and p are both large enough, we can assume that $\widehat{\Gamma}_i = \Gamma_i$.

5.4.2 Distance-based cost function

The simplest possible cost function is one based on the distance between nearby balloons. The cost function must be convex, so we can use quadratic functions of $\ell_{\mathbb{G}}(q_i, q_j)$. Since the objective is to spread the balloons, we want to maximize the distance between balloons which could be achieved by a cost function such as

$$y_{D,i} = -\frac{1}{p}\sum_{j=1}^{p} \ell_{\mathbb{G}}^2(q_i, q_j). \tag{5.14}$$

While this cost function may be useful for spreading balloons out, balloon i cannot actually compute y_i as it only has access to q_j if $q_j \in \mathcal{B}_{\ell_{\max}}(q_i)$ so it is unlikely that balloon i knows the position of all other balloons. Another option is to only consider balloons that balloon i can actually communicate with, however such a function would be discontinuous as q_j approaches the boundary of $\mathcal{B}_{\ell_{\max}}(q_i)$, and would therefore not be convex.

Instead of simply trying to maximize the distance, we could use a cost function designed to move the balloons a set distance apart. Suppose we would like to set nearby balloons a distance of ℓ^* apart. Then we could use the cost function

$$y_{D,i} = \frac{1}{\|\mathbb{J}_i\|} \sum_{j \in \mathbb{J}_i} \left(\ell_{\mathbb{G}}(q_i, q_j) - \ell^*\right)^2 . \tag{5.15}$$

This cost function is minimized when all nearby balloons—specifically the balloons which share an edge in the Delaunay triangulation—are an equal distance apart. The main difficulty with this cost function is the choice of ℓ^*. As the number of balloons increases, the balloons should on average get closer together so ℓ^* should decrease as p increases, but there is no obvious way to choose ℓ^*. Furthermore, this cost function is discontinuous when $\|\mathbb{J}_i\|$ changes and is therefore not necessarily convex.

5.4.3 Voronoi centroid-based cost function

Rather than use the distance between nearby balloons, we can use a cost function based on the distance between a balloon and $q_{c,i}$, the centroid of its Voronoi cell:

$$y_{C,i} = \ell^2_{\mathbb{G}}(q_i, q_{c,i}). \tag{5.16}$$

Since $q_{c,i}$ is a continuous function of the positions of nearby balloons, $y_{C,i}$ is continuous and, since it is quadratic, it is convex. Using this cost function results in each balloon moving towards the centroid of its Voronoi cell, resulting in an algorithm resembling k-means clustering or Lloyd's algorithm which is a well known algorithm for evenly spacing a set of points [61]. Furthermore, this cost function has the advantage that it moves each balloon to the centroid of its Voronoi cell, so it tends to result in more consistent signal strengths for all of the users that are connected to that balloon.

The centroids of Voronoi cells can be computed by decomposing the Voronoi cell into several triangles, computing their centroids, averaging the triangular centroids weighted by the triangular areas, and then projecting the resulting point onto the sphere. Since each balloon knows an estimate of its Voronoi cell, it can compute its centroid and therefore has access to $\widehat{q}_{c,i}$, the centroid of $\widehat{\Gamma}_i$. Assuming that $\ell_{\max}$ and p are large enough, $\widehat{q}_{c,i} = q_{c,i}$ so balloon i can measure $y_{C,i}$.

5.4.4 Voronoi area-based cost function

Another way to use the Voronoi partition is to define a cost function based on the areas of the Voronoi cells. Γ_i represents the region on Earth where users would connect to balloon i from. To share the internet traffic amongst all balloons equally, these regions should all be the same area. Let A_i be the area of Γ_i and A_t be the total area of Earth. Then we can define a cost function by

$$y_{A,i} = \left(A_i - \frac{A_t}{p}\right)^2. \tag{5.17}$$

This cost function is convex and is minimized when all of the balloons have the same coverage area.

The Voronoi areas can be easily computed by each balloon given Γ_i. Since Voronoi cells are spherical polygons, their areas can be computed from the interior angles $\alpha_1, \ldots, \alpha_m$ of Γ_i by the formula

$$A_i = R_\oplus^2 \left((2-m)\pi + \sum_{i=1}^{m} \alpha_i\right) \tag{5.18}$$

where $R_\oplus$ is the radius of Earth. Alternatively, the cost functions can be based on the solid angle to avoid multiplying by $R_\oplus^2$.

5.4.5 Internet bandwidth-based cost function

Balloons can also measure r_i, the number of users they are connected to, and b_i, the bandwidth of data flowing through the balloon. To equally share the internet traffic, we can use a cost function based on the bandwidth:

$$y_{B,i} = \left(b_i - \frac{1}{p} \sum_{j=1}^{p} b_j \right)^2 . \tag{5.19}$$

This cost function is convex and is minimized when all balloons the total bandwidth is shared equally between all balloons. Unfortunately, this cost function relies on the total bandwidth of the entire network, which is unknown since balloon i can only obtain b_j if $q_j \in \mathcal{B}_{\ell_{\max}}(q_i)$. However, since the total bandwidth does not change that much from day to day $\widehat{b}_t$, a nominal estimate of $\sum_{j=1}^{p} b_j$, could be used instead resulting in a cost function

$$y_{B,i} = \left(b_i - \frac{\widehat{b}_t}{p} \right)^2 . \tag{5.20}$$

Another major problem with using a bandwidth-based cost function is that the bandwidth in a particular location varies with the time of day. Since fewer people use the internet at night than during the day, $y_{B,i}$ would oscillate at a frequency of one day if balloon i were to stay still. Since the balloons do move, they would try to move west fast enough that they always remain in a timezone where it is currently a time of day that a lot of people are using the internet. Since the balloons can not move nearly fast enough to cycle the Earth once per day, using a cost function based on the current bandwidth would be ineffective. One way to fix this problem would be to integrate the bandwidth over a model-free hour period; however doing so would result in the costs changing much slower and forcing the formation controller to operate on a time-scale slower than one day, which is not practical.

Connected users r_i could be used instead of current bandwidth or bandwidth in the past day. Then the cost function would take the form

$$y_{U,i} = \left(r_i - \frac{\widehat{r}_t}{p} \right)^2 \tag{5.21}$$

where $\widehat{r}_t$ is a nominal estimate of the total number of internet users. Since users often remain connected to an internet source all day even when not actively sending or receiving data, $y_{U,i}$ would not have the same problems associated with time of day that $y_{B,i}$ would suffer from. Furthermore, since the number of internet users is more constant than bandwidth, it is easier to estimate $\widehat{r}_t$ than $\widehat{b}_t$.

The main drawback of using $y_{U,i}$ is that it tends to concentrate balloons where there are already internet users. The optimal formation would involve many balloons over North America and Europe, where the majority of people have internet access, but not over Asia and Africa, where the majority of people are not connected. Since the performance of the network in an area depends on how many balloons are there, the internet connection would be strongest in places where there is already good internet infrastructure. Over emerging countries, where there are few internet users, the internet connection from the balloons would be quite weak, and therefore few people would want to connect to the balloons. Therefore this cost function would not result in many new people connecting to the internet, and would not be successful in the goal of making the internet truly global.

5.4.6 Population-based cost function

To provide strong internet connections everywhere that people live, and thereby encourage more people to become internet users, we could use a cost function based on population instead of internet users:

$$y_{P,i} = \left(p_i - \frac{p_t}{p}\right)^2 \tag{5.22}$$

where p_i is the population in Γ_i and p_t is the total population of Earth. Given a population density function $\rho : \mathbb{S}^2 \to \mathbb{R}_{\geq 0}$, the population in a Voronoi cell can be computed as

$$p_i = \int_{\Gamma_i} \rho(q)\mathrm{d}q. \tag{5.23}$$

This cost relies on knowledge of ρ. While an analytical form of ρ is not known, gridded population density (Figure 5.8) is avilable from NASA's Socioeconomic Data and Applications Center [94]. These data can be numerically integrated over the known region Γ_i allowing each balloon to compute its own $y_{P,i}$.

When using a population-based cost function, very few balloons would be positioned over the oceans. Since $p_i = 0$ for a balloon over the ocean, the balloon would travel across the ocean as quickly as possible. This behaviour may or may not be desirable. Balloons over the ocean are necessary for routing so that users can connect across the ocean; however the number of users over the ocean should be minimized since they do not connect directly to users. Using a cost function only based on population could result in poor performance due to too few balloons for routing across the ocean, or it could result in enough balloons for routing and have better performance

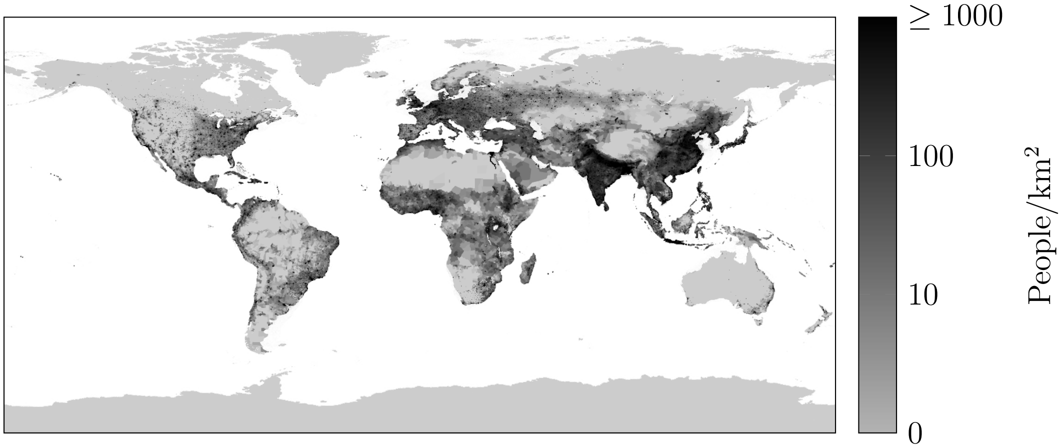


Figure 5.8: Population density of Earth. Data taken from [94].

as more balloons are over land.

5.4.7 Summary of cost functions

We have provided several cost functions which, when minimized, would result in balloons spread out over the Earth, providing everyone with internet access.

- $y_{D,i}$ is based solely on distance between balloons. It is not recommended as balloons can only communicate with nearby balloons, and as it is hard to choose the optimal distance between balloons.
- $y_{C,i}$ is based on the distance between a balloon and the centroid of its Voronoi cell. It results in an algorithm resembling Lloyd's algorithm [85].
- $y_{A,i}$, is based on the area of Voronoi cells. It results in each balloon being responsible for an equal area of coverage.
- $y_{B,i}$, is based on the bandwidth of each balloon. It is not recommended as bandwidth in one location changes too much over the course of a day.

- $y_{U,i}$, is based on the number of users connected to each balloon. It is not recommended as it only provides good internet coverage to regions that already have high internet penetration.

- $y_{P,i}$, is based on the total population each balloon could connect to. It results in each balloon being responsible for an equal population of potential internet users.

In practice, multiple cost functions can be used simultaneously, to combine their characteristics, and obtain a more robust system while allowing for the performance of the system to be changed by tuning the cost function. For example, consider a cost function of the form

$$y_i = K_C y_{C,i} + K_A y_{A,i} + K_P y_{P,i} \tag{5.24}$$

where $K_C, K_A, K_P \in \mathbb{R}_{\geq 0}$ are tuning parameters. This cost function spreads the balloons out so that all the balloon's Voronoi cell's have similar populations and areas and so that the balloons are located at the centroids of their Voronoi cells. By increasing K_P, balloons will be more concentrated over highly populated areas and by increasing K_A, balloons will be more evenly spread out.

5.5 Distributed ESC for balloons

We consider both the continuous- and discrete-time distributed ESC controllers from Chapter 3 and Chapter 4 for solving the balloon formation control problem. Each balloon measures a local cost based on its position and connectivity and controls one input which is the pressure inside the balloon. Distributed ESC is useful for this problem as it meets the design requirements of locally coordinate motion based only on local measurements and communication with neighbours without using a complex model. The effectiveness of these controllers in simulation depends on the choices of wind model, cost function, and tuning parameters. The main requirement is to ensure that all the assumptions made during the controller design are met by the balloon system.

Before considering the convergence properties of the controllers, we briefly summarize the equations defining the control systems. In continuous-time, the distributed ESC has the form

$$\begin{bmatrix} \dot{\widehat{\boldsymbol{J}}} \\ \dot{\boldsymbol{\rho}} \end{bmatrix} = \begin{bmatrix} -\kappa_0 \boldsymbol{I} - \kappa_P \boldsymbol{L} & \kappa_I \boldsymbol{L} \\ -\kappa_I \boldsymbol{L} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{J}} \\ \boldsymbol{\rho} \end{bmatrix} + \begin{bmatrix} \kappa_0 \boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{y} \tag{5.25}$$

$$e_i = \widehat{J}_i - \widehat{y}_i \tag{5.26}$$

$$\dot{\widehat{y}}_i = \boldsymbol{\phi}_i^T \widehat{\boldsymbol{\theta}}_i + K e_i + \boldsymbol{w}_i^T \dot{\widehat{\boldsymbol{\theta}}}_i \tag{5.27}$$

$$\dot{\boldsymbol{w}}_i = -K \boldsymbol{w}_i + \boldsymbol{\phi}_i \tag{5.28}$$

$$\dot{\widehat{\eta}}_i = -K \widehat{\eta}_i \tag{5.29}$$

$$\dot{\boldsymbol{\Sigma}}_i = \boldsymbol{w}_i \boldsymbol{w}_i^T - K_T \boldsymbol{\Sigma}_i + \sigma_1 \boldsymbol{I} \tag{5.30}$$

$$\dot{\widehat{\boldsymbol{\theta}}}_i = \operatorname{proj} \left\{ \boldsymbol{\Sigma}_i^{-1} \left(\boldsymbol{w}_i \left(e_i - \widehat{\eta}_i \right) - \sigma_1 \widehat{\boldsymbol{\theta}}_i \right), \Theta \right\} \tag{5.31}$$

$$u_i = -K_g \widehat{\theta}_{1,i} + \widehat{u}_i + d_i(t) \tag{5.32}$$

$$\dot{\widehat{u}}_i = -\frac{1}{\tau_I} \widehat{\theta}_{1,i} \tag{5.33}$$

where the tuning parameters are κ_0, κ_P, κ_I, σ_1, K, K_T, K_g, and τ_I. All of these parameters must be positive. The consensus gains κ_0, κ_P, and κ_I should be chosen to be quite large, as they affect the convergence rate of the consensus and without a good estimate of the total cost, the parameter estimation and optimization do not perform well. The estimation gains K and K_T should generally be smaller than the consensus gains. The ESC gain K_g and time constant τ_I can be tuned like typical PI tuning parameters. The parameter σ_1 is used to ensure the covariance matrix remains invertible and should chosen to be very small.

The discrete-time distributed ESC has the form

$$\begin{aligned}\begin{bmatrix}\widehat{\boldsymbol{J}}[k+1]-\widehat{\boldsymbol{J}}[k]\\ \boldsymbol{\rho}[k+1]-\boldsymbol{\rho}[k]\end{bmatrix} &= \begin{bmatrix}-\kappa_P\boldsymbol{I}-\kappa_I\boldsymbol{L} & -\boldsymbol{I}\\ \kappa_P\kappa_I\boldsymbol{L} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\widehat{\boldsymbol{J}}[k]\\ \boldsymbol{\rho}[k]\end{bmatrix}\Delta t\\ &\quad+\begin{bmatrix}\kappa_P\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{y}[k]\Delta t+\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}\Delta\boldsymbol{y}[k]\end{aligned} \tag{5.34}$$

$$e_i[k] = \widehat{J}_i[k]-\widehat{y}_i[k] \tag{5.35}$$

$$\begin{aligned}\widehat{y}_i[k+1] &= \widehat{y}_i[k]+\widehat{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\phi}_i[k]+Ke_i[k]\\ &\quad+\boldsymbol{w}_i^\top[k+1]\left(\widehat{\boldsymbol{\theta}}_i[k+1]-\widehat{\boldsymbol{\theta}}_i[k]\right)\end{aligned} \tag{5.36}$$

$$\boldsymbol{w}_i[k+1] = \boldsymbol{w}_i[k]+\boldsymbol{\phi}_i[k]-K\boldsymbol{w}_i[k] \tag{5.37}$$

$$\widehat{\eta}_i[k+1] = \widehat{\eta}_i[k]-K\widehat{\eta}_i[k] \tag{5.38}$$

$$\boldsymbol{\Sigma}_i[k+1] = \alpha\boldsymbol{\Sigma}_i[k]+\boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k] \tag{5.39}$$

$$\widehat{\boldsymbol{\theta}}_i[k+1] = \operatorname{Proj}_{\gamma_\theta}\left(\widehat{\boldsymbol{\theta}}_i[k]+\frac{\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]\left(e_i[k]-\widehat{\eta}[k]\right)}{\alpha+\boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]}\right) \tag{5.40}$$

$$u_i[k] = -K_g\widehat{\theta}_{1,i}[k]+\widehat{u}_i[k]+d_i[k] \tag{5.41}$$

$$\widehat{u}_i[k+1] = \widehat{u}_i[k]-\frac{1}{\tau_I}\widehat{\theta}_{1,i}[k] \tag{5.42}$$

where Δt is the time step and κ_P, κ_I, K, α, K_g, and τ_I are positive tuning parameters. The consensus gains κ_P and κ_I should be inversely proportional to the time step. The estimation gain K and RLS forgetting factor α should both be between 0 and 1 to ensure stability of the parameter estimation. The ESC gain K_g and time constant τ_I can be tuned like typical PI tuning parameters.

Both continuous- and discrete-time controllers use a projection algorithm on the update law for $\widehat{\boldsymbol{\theta}}$ which bounds this parameter. This projection has the same effect as a saturation on the input. The radius γ_θ of the parameter set Θ should be chosen based on the maximum rate that a balloon can increase or decrease its altitude by changing its internal pressure. Additionally, a projection algorithm can be used on the input bias $\widehat{u}_i$ to prevent integrator windup [109].

The choice of dither signals is also important to a successful distributed ESC implementation. The dither frequencies must be unique so that each balloon can identify its own effect on the total cost. The dither amplitude must be large enough that the dither signals can be distinguished from noise. The frequencies should be slow enough that the balloon can actually move up and down that fast, but should not be too slow, or else the estimation would be slow. Furthermore, in discrete-time, the time step should be considered when choosing the range of dither frequencies so that several samples are taken period of each dither and so that the sampling frequency and dither frequencies are also unique.

For the local ESC cost functions, we propose a function similar to (5.24). This cost function relies on three parameters K_C, K_A, and K_P which weigh the relative contributions of the centroid-based cost (5.16), area-based cost (5.17), and population-based cost (5.22). The choice of these parameters depends on the units each cost are measured in as well as the desired closed-loop performance. In simulation, we use the cost function

$$y_i = \left(A_i - \frac{A_t}{p} \right)^2 + \ell_{\mathbb{G}}^2(q_i, q_{c,i}). \tag{5.43}$$

Since y_P is time-consuming to compute, we set $K_P = 0$ to decrease simulation time. K_C and K_A were both simply set to 1 in this cost function.

5.5.1 Convergence analysis

Theorems 3.1 and 4.1 provide convergence results for both types of distributed ESC. These theorems show that under some mild assumptions the system converges exponentially to a neighbourhood of the optimum of the cost function. The main assumptions required for the convergence of the consensus algorithm and parameter estimation in the proofs of the two theorems are:

1. *Convexity of the total cost function*, which is guaranteed as the local costs are quadratic,

2. *Lipschitzness of the cost function and vector fields* which is met by the compactness of $\mathbb{S}^2$ and smoothness of $f\{u_i\}(t, q_i)$ and $H(q)$,

3. *Connectedness of the communication network*, which is enforced by choice of network, and

4. *Persistence of excitation in the input signals*, which is provided by the unique dither signals.

Since these assumptions are met, we can use the results from Theorems 3.1 and 4.1 to show that $\widehat{J}_i$ and $\widehat{\boldsymbol{\theta}}_i$ converge to a small neighbourhood of $\frac{1}{p}J$ and $\boldsymbol{\theta}_i$.

The optimization portion of distributed ESC requires an additional assumption to be met:

5. *Stabilizability of the system's dynamics to the steady-state manifold*, which is met when we use the simple linear wind model in (5.11)–(5.12) with a rotating reference frame. However, when using an arbitrary nonlinear, realistic wind model, this assumption is not met as the balloon formation is dynamic and there is no steady-state manifold.

Without this last result holding, we cannot guarantee that the system moves into an optimal formation. However, a simple modification of either proof can be used to show that while $J(t)$ does not always decrease, the distributed ESC minimizes $\dot{J}(t)$ or $\Delta J[k+1]$ even though this derivative or difference may occasionally be positive as there is not always a choice of u_i which results in the total cost decreasing. Furthermore, when J is large, it is more likely that there exists u_i which causes J to decrease. Therefore the distributed ESC tends to move balloons away from bad formations but may not quite reach the optimal formation.

Since the balloons cannot fly actively, they continue to move so it is unreasonable to expect that the balloons would reach the steady-state optimum, as they would have to remain statically in one formation. Since the balloons will be blown out of their formation by unavoidable wind currents, the best we can expect for a controller is one that is able to constantly improve the formation despite the wind currents naturally blowing the balloons into a less favourable formation.

5.6 Simulation results

5.6.1 Continuous-time simulation results

First, we consider a simple, ideal simulation example in continuous-time. In this example, the balloons move on very simple winds which only move east-west, which results in the system having a steady-state in an appropriate rotating reference frame. Thus, in this example, the system meets all of the assumptions required in Chapter 3. The balloons begin at initial conditions that are close to the known optimal formation of the vertices of a dodecahedron and can reach this formation simply by following the appropriate wind currents. We apply the continuous-time distributed ESC from (5.25)–(5.33) and use this example simply to ensure that this technique can indeed solve a simple, ideal formation control problem with a known solution.

Consider a system of 20 balloons implementing distributed ESC while floating on wind currents described by (5.11)–(5.12). From Figure 5.4, winds in the stratosphere have velocities ranging between 40–100 m/s which vary linearly with altitude. Therefore between altitudes of 0–50 km, we use a linear model of wind velocity:

$$\begin{aligned} u_{\text{wind}} &= \left(\frac{60\,\text{m/s}}{40\,\text{km}}\right) a + 25\,\text{m/s} \\ v_{\text{wind}} &= 0 \end{aligned}$$

where a is the altitude. These data are from 60° south where Earth's circumference is 20,000 km. Using $u_i = a_i - 30\,\text{km}$ as the input parameter, we can write the dynamics in coordinates as

$$\begin{aligned}\dot{\lambda}_i &= 4.55 + 0.0975 u_i \\ \dot{\varphi}_i &= 0\end{aligned}$$

where λ_i and φ_i are the longitude and latitude of the i^{th} balloon in degrees and t is in hours. Note that all balloons have identical dynamics which are not coupled, but that each balloon's local cost is influenced by the other balloons.

The objective is to spread the balloons out equally over the globe. In their optimal formation, the areas of all Voronoi cells should be equal with each balloon at the centroid of its Voronoi cell. Therefore, the local costs are

$$y_i = \left(A_i - \frac{\pi R^2}{5}\right)^2 + \ell^2_{\mathbb{G}}(q_i, q_{c,i}).$$

The first term is equalizes the Voronoi areas; the second term moves each balloon to the centroid of its Voronoi cell. The total cost function $J = \sum_{i=1}^{20} y_i$ is minimized when the balloons are at the vertices of a dodecahedron.

Note that using this cost function results in balloons having equal areas of coverage but not necessarily an equal number of users as in requirement 7. This cost function could be improved by replacing the area with r_i; however doing so requires numerical integration which drastically increases simulation time. Using area in the cost also has the advantage that the optimal formation is known exactly.

In this simulation, the wind currents have no north-south components. Therefore, balloons cannot change their latitude. The only way that the balloons can reach the optimal dodecahedral arrangement is if they begin at the correct latitudes. When

Table 5.1: Initial conditions of the 20 balloons, $(\lambda_{0,i}, \varphi_{0,i})$ compared to the locations of the dodecahedral vertices, $(\lambda_{d,i}, \varphi_{d,i})$.

Balloon	$\lambda_{0,i}$	$\varphi_{0,i}$	$\lambda_{d,i}$	$\varphi_{d,i}$
1	117.15	69.09	90.00	69.09
2	−59.69	69.09	−90.00	69.09
3	74.73	35.26	45.00	35.26
4	29.31	35.26	−45.00	35.26
5	161.22	35.26	135.00	35.26
6	−128.15	35.26	−135.00	35.26
7	28.24	20.91	0.00	20.91
8	−178.73	20.91	180.00	20.91
9	80.17	0.00	69.09	0.00
10	−67.25	0.00	−69.09	0.00
11	114.79	0.00	110.91	0.00
12	−77.97	0.00	−110.91	0.00
13	27.79	−20.91	0.00	−20.91
14	−167.32	−20.91	180.00	−20.91
15	83.01	−35.26	45.00	−35.26
16	−43.62	−35.26	−45.00	−35.26
17	152.55	−35.26	135.00	−35.26
18	−119.74	−35.26	−135.00	−35.26
19	120.62	−69.09	90.00	−35.26
20	−58.19	−69.09	−90.00	−35.26

initializing the simulation, we place all of the balloons at the correct latitude but at a longitude which deviates from the location of the dodecahedral vertex by a random amount (Table 5.1). This initialization ensures that balloons are able to reach the optimal formation.

The balloons communicate with their nearest neighbours. The initial formation resembles the vertices of distorted dodecahedron and the communication network can be represented by the edges of this distorted dodecahedron. Using this network, each balloon can communicate with exactly three other balloons which are the three balloons which are closest to it in the optimal formation.

Table 5.2: Tuning parameters used by the balloons implementing the continuous-time distributed ESC.

κ_0	κ_P	κ_I	K	σ_1	K_T	τ_I	K_g	D
100	100	100	50	1×10^{-7}	50	10	100	10

Each controller uses identical tuning parameters (Table 5.2). The only parameter which differs between the balloons is the frequency ω_i of its dither signal. The dithers must be unique so that each balloon can estimate its own parameters while ignoring the effects of other balloons which have different frequency dithers. While there are many choices of dither frequencies which work, we have chosen to use prime frequencies for each balloon.

The system of 20 balloons was simulated for 21 days using distributed ESC to control the formation. During this time, the local costs all decreased and the total cost decreased to close to 20% of its original value (Figure 5.9). Once the system reached this neighbourhood of the optimal cost it is stable around that point. Plots of the Voronoi cells (Figure 5.10) show that the balloons move from the initial random formation to a more evenly distributed formation where the Voronoi cells are similarly sized triangles. This simulation shows that distributed ESC can indeed by applied to formation control problems and that the choice of cost function is effective.

5.6.2 Discrete-time simulation results

Next we consider a more realistic wind model. In this case, there is not necessarily a steady-state and so the problem of steady-state cost function minimization is not well-defined. Nevertheless, we apply distributed ESC as a method of continuously improving the formation of balloons. Since we now consider real wind models and a

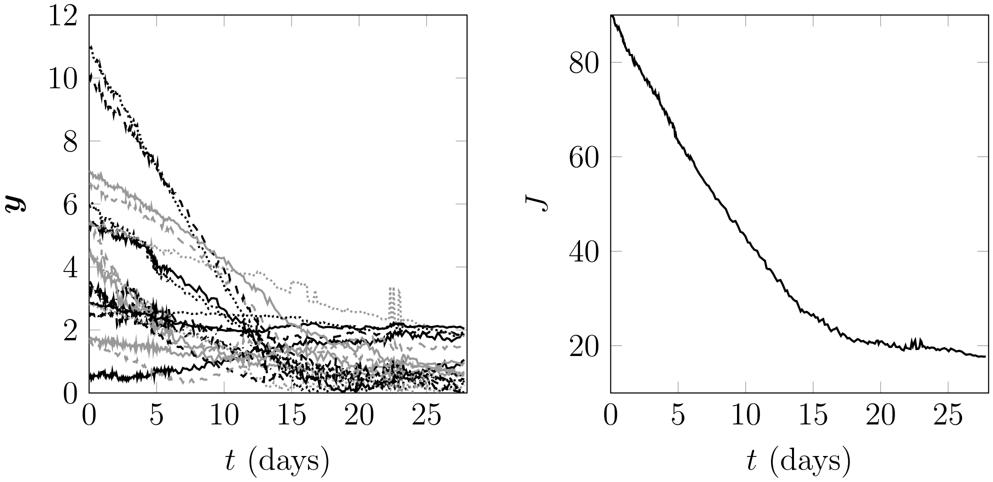
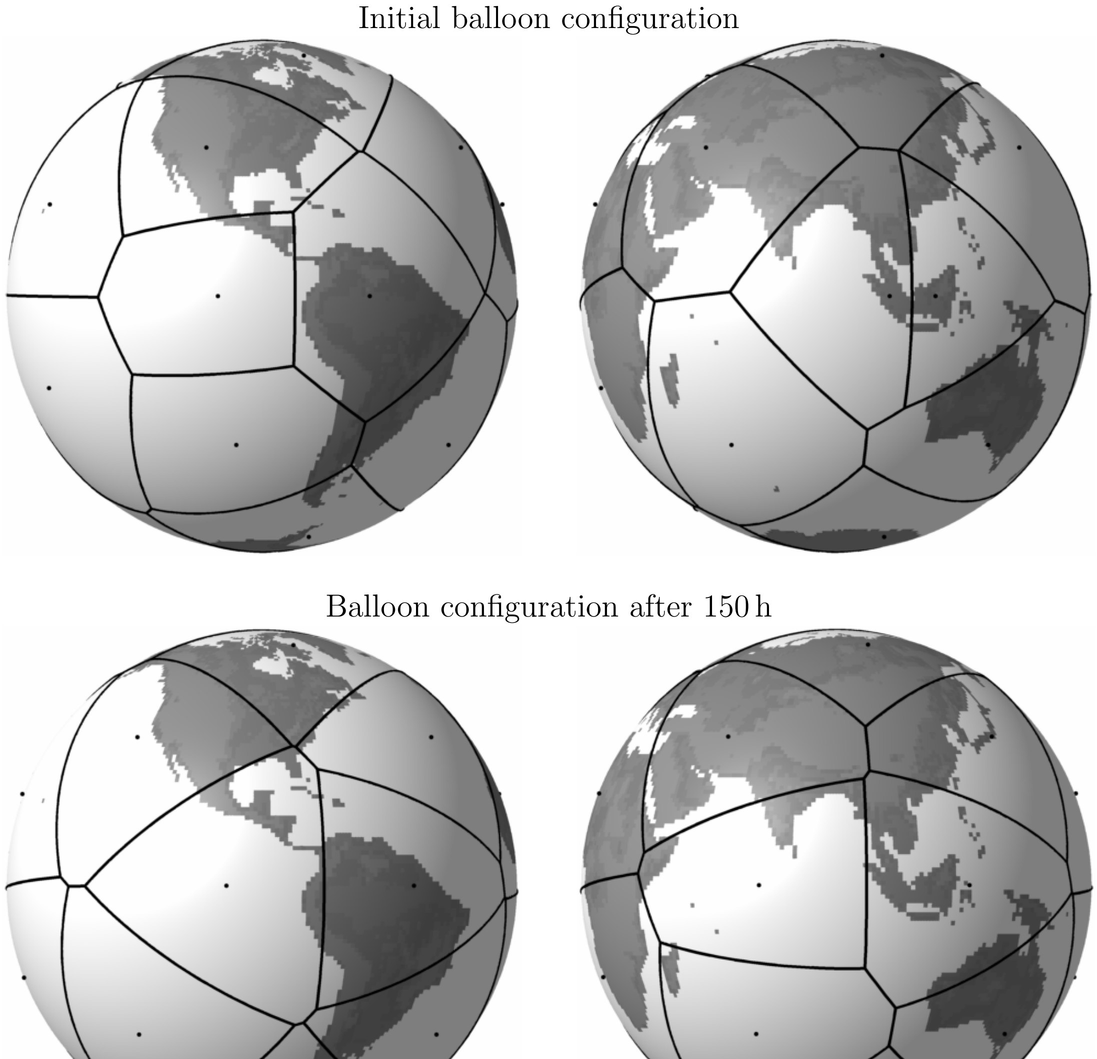


Figure 5.9: Local costs (left) and total cost (right) for the system of 20 balloons under distributed ESC in continuous-time.

system of 1200 balloons, discrete-time simulations are much simpler to perform, so we use the discrete-time extremum-seeking controller (5.34)–(5.42).

Suppose the balloons start in a good formation where they are spread out over the globe. They continue to move with the wind, but must stay in a similar formation to provide good coverage over the entire Earth. Since balloons tend to follow prevailing winds, all balloons naturally end up following the same patterns and may congregate at a specific place, such as in a hurricane, or along one dominant wind current. The first simulation considers the problem of maintaining the balloons in a good formation when they start in a good one.

In any real life situation, balloons must be launched from a fixed location. This location must be equipped with the necessary launch equipment and the balloons leaving this location have to travel through the troposphere to reach their usual altitude in the stratosphere. Since the balloons could disrupt aircrafts or birds in the troposphere, the company performing the launch must obtain the relevant permits to



Initial balloon configuration

Balloon configuration after 150 h

Figure 5.10: Initial randomized balloon configurations and balloon configuration after 150 h of distributed ESC. The dots represent the balloons. The lines represent the boundaries of Voronoi cells.

launch at each location. The requirements of launch equipment and permits result in a limited number of launch sites with many balloons being launched from each site. The second simulation example considers the problem of steering the balloons from a few initial launch locations to an evenly distributed formation.

These simulations were performed using wind data from March 8, 2016 at 17:00 UTC which was obtained from the NOAA [95]. The data was obtained as u- and v-components of the wind velocity on a three-dimensional grid of longitude, latitude, and isobaric pressure. The longitude and latitude grids have resolutions of $0.25°$ and the isobaric pressure grid is not uniform and has values at 1, 2, 3, 5, 7, and 10 kPa. Since isobaric pressure is a monotonic function of altitude, this grid could be converted into an altitude grid. Linear interpolation was used between grid points. Since the grid is high-resolution, linear interpolation is a good approximation.

The simulations run for a total of 1000 h. After such a long time, the wind currents can change significantly. While a more realistic simulation would use a time-varying wind model, we have opted for a simpler time-invariant model. A dynamic wind model can be created from historic wind data from the NOAA. These data are available at 6 hour increments, however a time-varying model uses significantly more data requiring many gigabytes of hard drive space and interpolation must be performed over 4 dimensions, which increases simulation time. Since prevailing wind currents do not change, the advantage of using a time-varying model is small and we have opted to use a time-invariant wind model which is still realistic enough to demonstrate the effectiveness of distributed ESC.

Table 5.3: Tuning parameters used by the balloons implementing discrete-time distributed ESC.

Δt	κ_P	κ_I	K	α	τ_I	K_g	D	γ_θ
0.1 h	1	0.5	0.8	0.8	10	1	0.1	1

5.6.3 Maintaining a formation

In this simulation, we consider a fleet of 1200 balloons which start at 1200 random locations. These locations were chosen from a uniform distribution on the sphere by choosing the (x, y, z)-coordinates of each balloon uniformly on $[-1, 1]^3$, normalizing the position vector, and then converting it into a longitude-latitude pair.

Each balloon is equipped with a distributed extremum-seeking controller which uses isobaric pressure as an input and measures a convex local cost given by (5.43). The total cost achieves its minimum when each Voronoi cell has the same area and each balloon is at the centroid of its Voronoi cell. Each balloon uses identical tuning parameters (Table 5.3) with the exception that each balloon uses a unique dither frequency. The dither frequencies are the consecutive primes between $121,369$ and $135,497$ multiplied by a common rational factor of $\frac{15}{135497}$. The balloons communicate over a time-varying connected network with balloons i and j communicating whenever $j \in \mathbb{J}_i$. The resulting communication network is equivalent to the Delaunay triangulation of the balloons and it is guaranteed to be connected.

The system was simulated from launch for 1000 h using two control schemes (Figure 5.11). The first control scheme is the distributed ESC which we have described. The second control scheme is to simply fly the balloons at 5 kPa, near the middle of the range of acceptable isobaric pressures (1–10 kPa), and not implement any other control action. The second scenario is how the balloons would naturally move without a controller. Without control, the balloons slowly congregate in some specific areas. After two weeks (336 h) there are very few balloons over the equator as most balloons have moved into stronger prevailing wind currents in both hemispheres. After 1000 h most of the balloons have entered into a few dominant wind patterns, either traveling over the Southern Ocean cycling around the south pole, or are traveling over Greenland and Eurasia cycling around the north pole. Additionally, many balloons are trapped in small cyclones above Australia, Mexico, the Yukon, the Atlantic south of Nigeria, the Pacific north of Australia, and the Pacific west of Chile. Large portions of North America, South America, Africa, India, and Southeast Asia receive no coverage.

Distributed ESC performs much better. While the balloons are all moving and their formation is changing, the final formation provides a similar level of coverage to the initial formation. There are no large areas that are lacking coverage and there are no areas with a very high balloon density. The balloons remain relatively well spread out when controlled by distributed ESC.

Another way to see the effectiveness of distributed ESC is by examining the total cost function (Figure 5.12). This cost function is a single positive number which represents the overall performance of the system. It reaches a minimum at $J = 0$ when the balloons are evenly distributed over the Earth. The cost can be thought of as a measure of how spread out or clumped together the balloons are. When J is small, the balloons are spread out; when J is large, the balloons are clumped together. The cost functions for both control schemes start at the same value since the balloons all start in the same locations. When no control is used, the cost increases rapidly as the balloons move from a relatively evenly distributed formation to a poorly

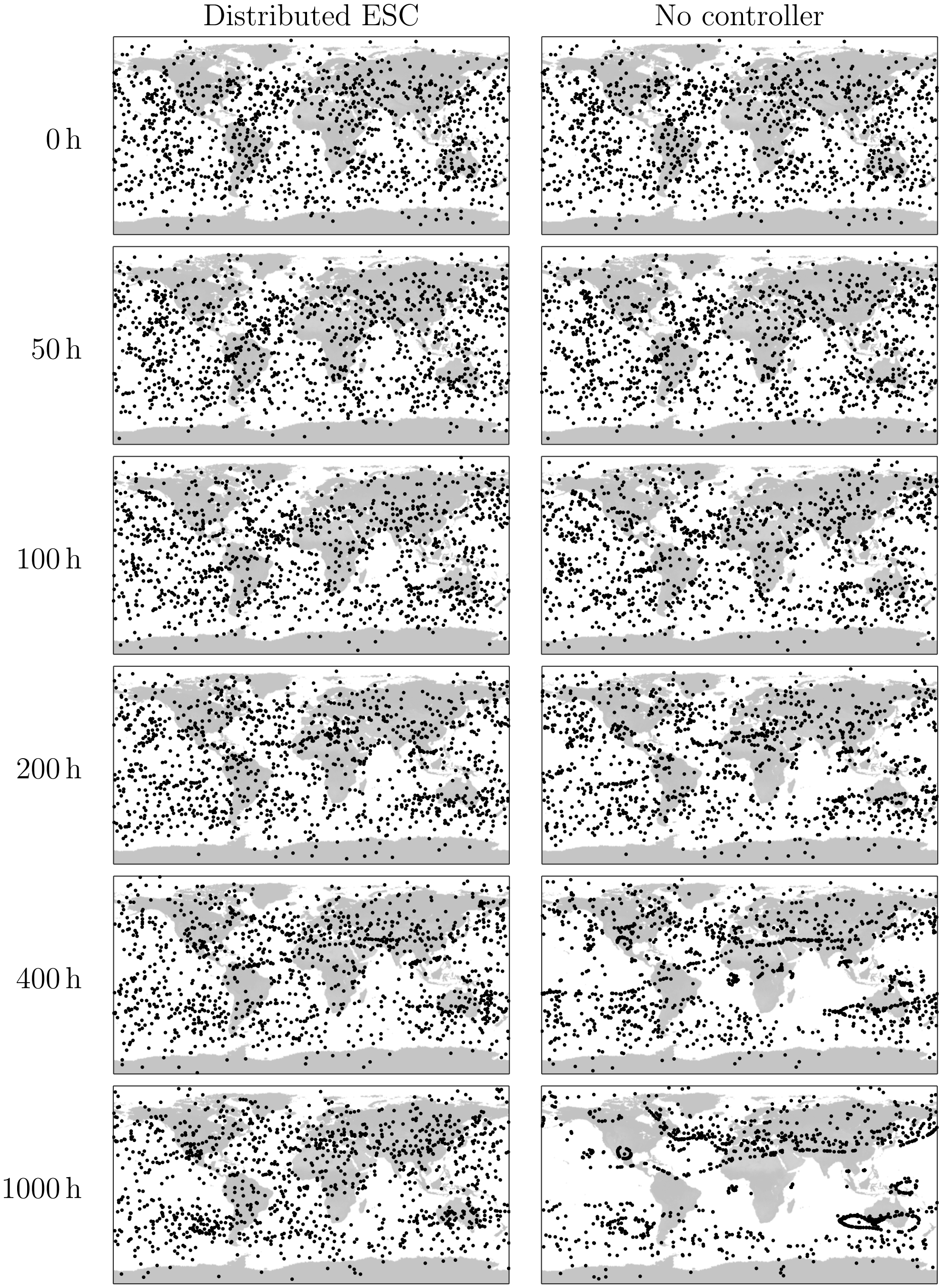


Figure 5.11: Position of 1200 balloons launched from random locations when controlled by distributed ESC (left) and when flown at 5 kPa with no controller (right).

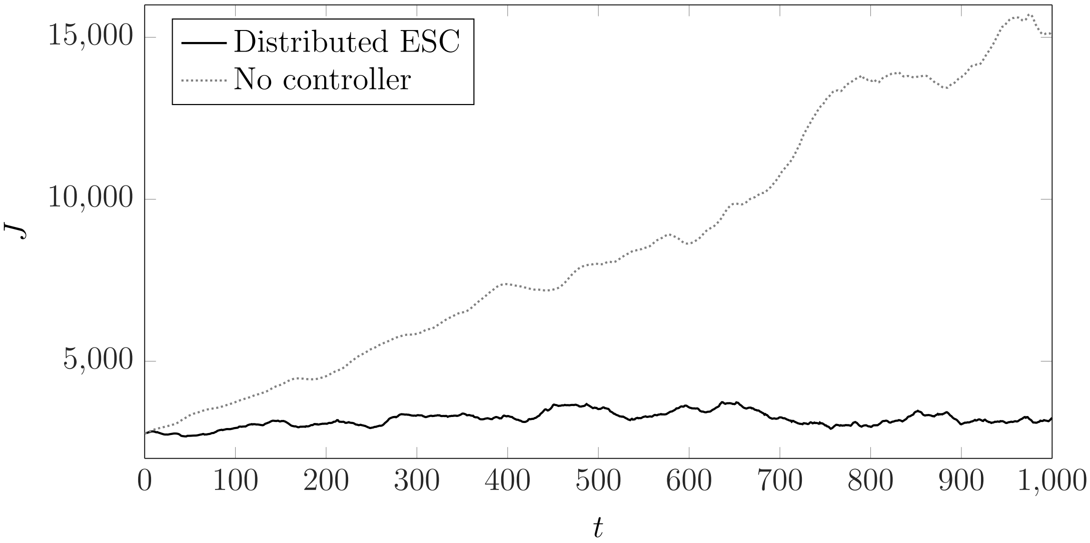


Figure 5.12: Total cost trajectories for a system of 1200 balloons launched from random positions when under no control (dashed) and when under distributed ESC (solid).

distributed formation as the balloons aggregate. When distributed ESC is used, the cost remains relatively constant, although it does have some variance. Since the cost does not increase or decrease significantly, the ESC is keeping the balloons in a formation which provides as good of coverage as when the balloons are randomly placed according to a uniform distribution. The minor variations are due to the non-uniformity of the wind currents which sometimes result in balloons naturally moving to slightly worse formations despite the control, as there are no available currents which move the balloons to a better formation.

From the map and the cost trajectories, we can see that distributed ESC is effective at maintaining a good formation, while using no controller is not. Without control, the balloons tend to move into dominant wind patterns; with ESC, the balloons do not get stuck in these weather features and are able to remain spread out.

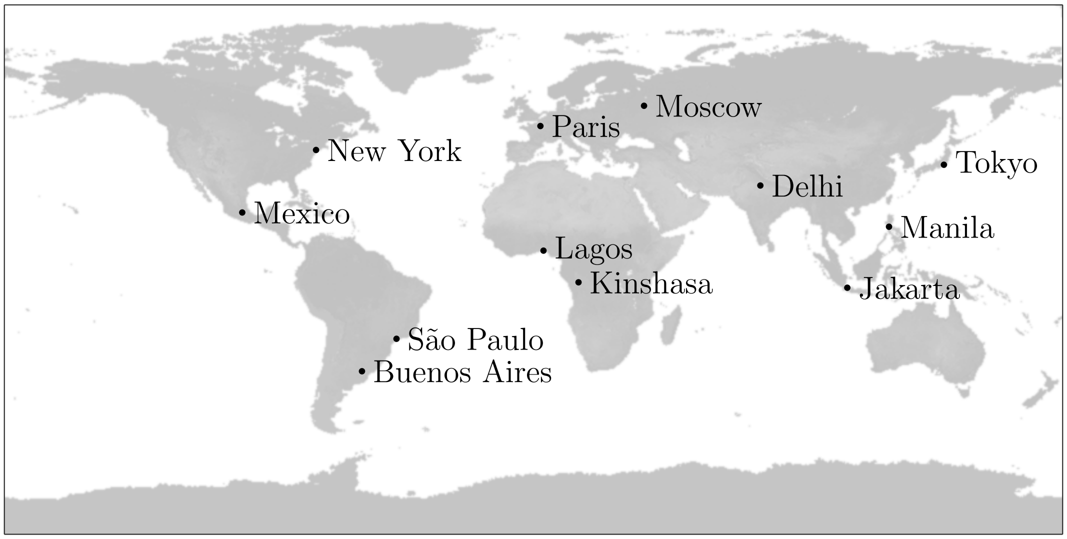


Figure 5.13: Locations of the twelve balloon launch sites.

5.6.4 Launching the balloons

In the next simulation, we again consider a fleet of 1200 balloons. This time the balloons are launched from 12 launch sites with 100 balloons at each site. There are two launch sites for each inhabited continent, located at the continent's two largest urban areas by population (Figure 5.13, Table 5.4). Since Oceania is small, maritime Southeast Asia was grouped with Oceania when selecting the launch sites for the simulation.

From these launch sites, we considered the trajectories of balloons when controlled by distributed ESC and without any control (Figure 5.14). Since 100 balloons start at each launch site, when flown at 5 kPa without control these balloons tend to follow the same wind currents and do not spread out much. After more than a month (744 h), the balloons have formed several closely spaced lines which follow the same routes. Furthermore, at the end of the simulation there are only 10 distinct groups, meaning that two pairs of initial groups of balloons have merged into a single group.

Table 5.4: Coordinates of the twelve launch sites.

City	Country	Continent	λ	φ
New York	USA	North America	−74.01	40.71
Mexico City	Mexico	North America	−99.13	19.43
São Paulo	Brazil	South America	−46.63	−23.55
Buenos Aires	Argentina	South America	−58.38	−34.60
Paris	France	Europe	2.35	48.86
Moscow	Russia	Europe	37.62	55.75
Lagos	Nigeria	Africa	3.40	6.45
Kinshasa	DR Congo	Africa	15.32	−4.33
Tokyo	Japan	Asia	139.68	35.68
Delhi	India	Asia	77.21	28.61
Jakarta	Indonesia	Oceania/SE Asia	106.80	−6.17
Manila	Philippines	Oceania/SE Asia	121.00	14.58

On the other hand, when distributed ESC is used, the balloons are able to spread out much more effectively. After only two days (48 h), the initial groups of balloons have spread out enough that it is difficult to distinguish which balloons started where. After two weeks (336 h), the balloons have almost completely covered all of the Earth with the only significant gaps being over the Pacific Ocean and Antarctica, which both do not have significant populations. After a month (744 h) all of the Earth except Antarctica is covered by balloons. No balloons move over Antarctica as the Antarcticic Circumpolar Current over the Southern Ocean creates strong prevailing westerly winds isolating the Antarctica air system and preventing balloons from floating over the continent.

Again, we can also examine the performance of the control system by looking at the total cost trajectories (Figure 5.15). The cost function for the ESC simulation drops rapidly and gets close to zero indicating the balloons are spreading out as desired. When no controller is used, the cost function varies widely and generally

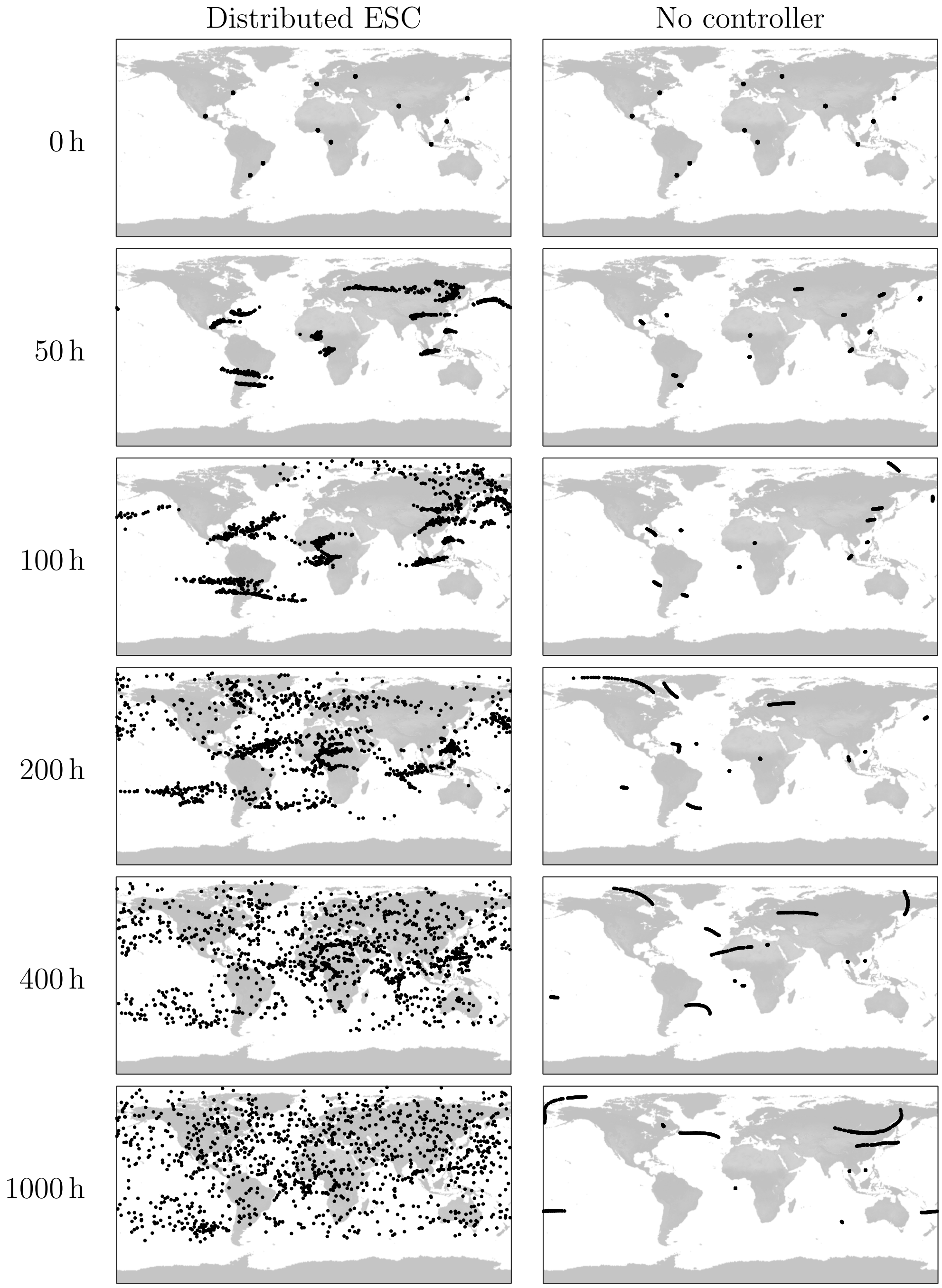


Figure 5.14: Position of 1200 balloons launched from 12 cities when controlled by distributed ESC (left) and when flown at 5 kPa with no controller (right).

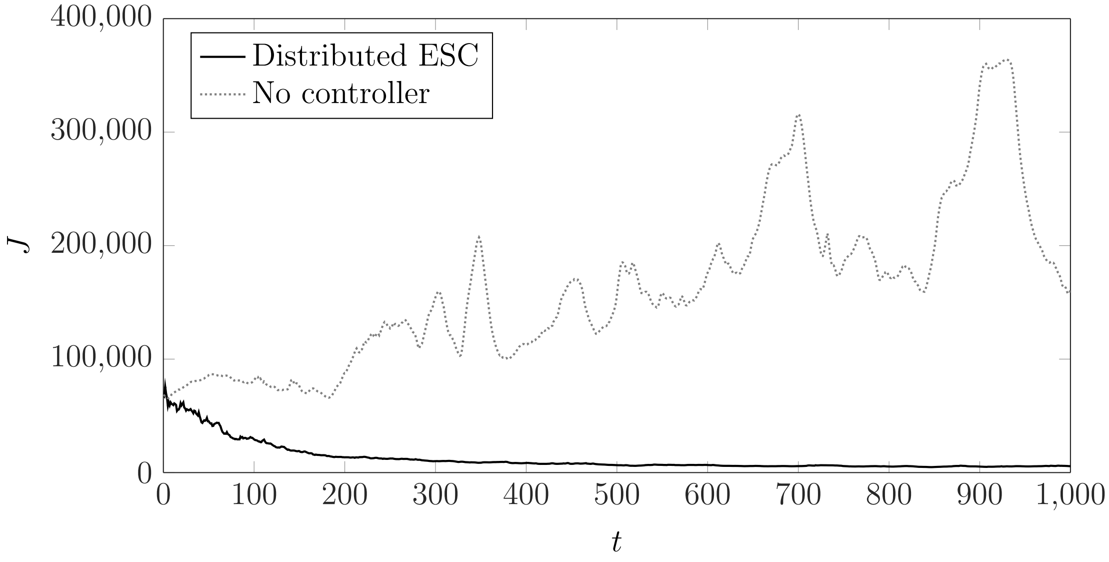


Figure 5.15: Total cost trajectories for a system of 1200 balloons launched from 12 cities when under no control (dotted) and when under distributed ESC (solid).

increases. Since it is increasing, the formation actually tends to get worse as time goes on. It has a high variance as the wind is non-uniform over the globe and may naturally blow balloons closer together or further apart.

Both the individual maps showing the simulation results and the performance metric of the cost function show that distributed ESC is an effective tool for launching balloons. While the balloons would naturally tend to stay together, distributed ESC is able to steer them apart into a formation that provides adequate coverage to the entire Earth. This process takes less than one month. Furthermore, these results indicate that even if all the balloons are released from the same place at the same time, distributed ESC can efficiently distribute them over the entire Earth.

5.7 Conclusion

A fleet of several thousand high-altitude balloons are a promising technology for providing high-speed internet to the entire Earth. One challenge in implementing such a network of balloons is controlling their formation. Google's Project Loon intends to solve this problem using complex wind models and lots of computing power. Other approaches requiring less computing power have been successful for simpler models with unrealistic linear wind currents.

We have shown that discrete-time distributed ESC is an effective method for formation control of balloons when subjected to realistic wind currents. Each balloon implements the same algorithm using only its own measurements and information received from nearby balloons. The algorithm does not require a central coordinator and is computationally efficient. It does not require a model and can adapt to changing, nonlinear wind currents. Distributed ESC relies on measurements of a cost function and constantly improves the formation of the balloons to move them towards a feasible optimum. Several cost functions based on distances between balloons, Voronoi partions, internet bandwidth, and number of internet users connected to a balloon can all be used. While each cost function results in slightly different optimal formations, they all can be used to spread out the balloons over Earth.

In continuous-time, a simulation involving 20 balloons was used to show the effectiveness of this strategy. The optimal formation for 20 balloons on a sphere is for each balloon to be at the vertex of a dodecahedron. Using distributed ESC, the system was able to drastically reduce the total cost over 3 weeks and the total cost was stabilized to a neighbourhood of the optimal cost by moving towards this optimal dodecahedral formation.

In discrete-time, a simulation involving 1200 balloons was used to show the effectiveness of this strategy. The balloons were subjected to realistic wind currents based on meteorological data from the NOAA. Each balloon implemented a PI ESC and communicated with nearby balloons resulting in a distributed ESC control scheme. This scheme was implemented in discrete-time with a time step of six minutes. In the simulation, 100 balloons were launched from each of 12 of the largest cities on Earth. Within a month's time, the balloons moved from being concentrated in 12 locations to being spread out over the entire Earth. This scheme vastly outperformed using no controller. A second simulation showed that when the balloons start in a good formation, distributed ESC can maintain this formation even though the balloons would naturally drift into a suboptimal formation.

Chapter 6

Conclusion

6.1 Summary

The real-time optimization of large-scale systems with complex or poorly understood dynamics is a challenging problem relevant to many diverse industries. Current RTO techniques use centralized approaches which require significant computational effort, and are not feasible for large-scale systems. Furthermore, modeling large-scale systems is a time-consuming and often futile procedure, especially when many parameters are unknown.

Distributed extremum-seeking control provides a simple, effective way to optimize many real-world systems of arbitrary size, while eliminating the need to develop a model of the system. In this thesis, distributed ESC was developed in both continuous- and discrete-time. These controllers were applied to several mathematical examples show to illustrate the types of problems which can be solved with these types of controllers. They were also shown to be an effective way to control the formation of a large fleet of over one thousand high-altitude balloons—a problem which could not be solved by any other existing technique—without requiring a wind model.

A continuous-time distributed extremum-seeking control algorithm was developed in Chapter 3. This controller consists of p agents which each measure a local cost and have the collective objective of minimizing the sum of these costs. Each agent implements an identical ESC. This controller consists of three parts: a dynamic average consensus, time-varying parameter estimation, and a gradient descent. The consensus algorithm is a second-order linear system which converges to the average of the time-varying local costs. It provides each agent with a convergent estimate of the total cost. Using the total cost estimate, two time-varying parameters are estimated. The first parameter represents all changes to the cost that the agent cannot control and is the sum of of the Lie derivatives of the cost along the drift vector field and along all other agents' control vector fields. The second parameter is the Lie derivative of the cost along the agent's control vector field, and describes how that agent's inputs affect the total cost. This second parameter is used in a proportional-integral control law, resulting in a gradient descent algorithm. With each agent implementing a gradient descent based on how its own input affects the total cost, the entire system converges to a neighbourhood of the optimum total cost. This approach only requires measurement of local costs and communication between nearby agents. It does not require a model of the system, a priori knowledge of parameters, or a measurement of the total cost.

In Chapter 4, an analogous discrete-time distributed ESC was developed. This controller consists of the same components—consensus, parameter estimation, and gradient descent—as the continuous-time controller; however, different algorithms must be used for convergence in discrete-time. In discrete-time, the parameter estimation routine estimates parameters which are not simply Lie derivatives, but instead depend on the mean-values of the Lie derivatives over the entire time step.

This parameterization results in a direct relationship between the time step of the discrete-time system and the parameters and hence convergence of the distributed ESC.

Chapter 5 contains a large-scale real-world simulation example showing a valuable application of distributed ESC. This application is the formation control of high-altitude balloons. These balloons are an emerging technology with the potential to bring high-speed wireless internet to people anywhere on Earth. The formation control problem is the problem of spreading the balloons out in such a way that every internet user on Earth is close to at least one balloon. As the balloons float passively in the stratosphere, they naturally follow Earth's wind currents, and control actuation is provided by pumping helium in or out of the balloon to move it up or down into a different wind current. Distributed ESC was an appropriate control algorithm for this system as several thousand balloons would be required, and accurate wind models are not available. We discussed several different cost functions which could be used to steer the balloon into a formation that provides adequate coverage to the entire Earth. For simulations, we used a cost function based on the areas and centroids of Voronoi cells which is optimized when each balloon is at the centroid of its Voronoi cell and the cells are all the same size.

We performed simulations involving 1200 balloons floating on wind currents simulated using data from the United States National Oceanic and Atmospheric Administration. In the first set of simulations, balloons began in a random formation that provided good coverage; in the second set of simulations, balloons began at one of twelve designated launch sites. When no controller was used, balloons starting in a random formation were slowly blown towards prevailing winds and no longer provided good coverage after one month. Distributed ESC, on the other hand, was able to maintain a formation which provided equivalent coverage to the random formation. Furthermore, when launched from launch sites, balloons under control of distributed ESC moved into a favourable formation within one month. This approach used a discrete-time controller with a time step of six minutes and is completely scalable to arbitrarily many balloons.

The distributed extremum-seeking control techniques developed in this thesis are able to solve real-time optimization tasks for a large class of systems. They provide scalable techniques that are computationally efficient and do not require a model. They provide good performance and can be used for nonlinear, time-varying, and slowly unstable systems. The techniques can thus be applied to many real-world problems, such as the formation control of balloons, which cannot be optimized by any other technique.

6.2 Future work

There are several challenges in distributed ESC that have yet to be addressed. The current techniques, while valid for a large class of systems, are not valid for all systems. In particular, the techniques are not valid for nonminimum phase systems or systems with relative degree greater than one. Future work on expanding the techniques to these important classes of systems would result in distributed ESC being applicable to even more real-world problems. Another interesting problem is developing a framework for including models into distributed ESC. Significant effort has been put into modeling of real-world systems and using the results of these modeling efforts could potentially result in more effective extremum-seeking techniques.

Additionally, there are some interesting problems that arise due to the use of a network in distributed ESC. One extension is asynchronous distributed ESC where individual agents operate on different clocks and do not send information at the same time. Another topic to consider is minimal communication, where each agent's goal is to optimize the system while communicating with other agents as infrequently as possible. A main concern when implementing control systems over networks is security. A malicious node which actively works against the efforts of the network could cause poor performance or even instability of the entire system. Future work should consider techniques to detect attacks and prevent malicious agents from making the entire system ineffective or unsafe.

There are several important future steps that should be taken when applying distributed ESC to fleets of high-altitude balloons. Time-varying simulations should be performed using real wind data collected over a period of several weeks to simulate a more realistic scenario. If it is possible to partner with a company, such as Google, then real experiments involving a few balloons should be conducted. These tests would be necessary to tune the distributed ESC and also to determine the best choice of cost function. The Voronoi-based cost functions were used in simulation as they are easy to compute, but a cost function involving internet bandwidth or number of connected users may be more effective at providing coverage.

As distributed control and ESC are relatively young fields, the future for distributed ESC is bright. There remain many interesting theoretical developments to be made in this area. Furthermore, as it is applicable to a large class of systems, there are many potential applications of distributed ESC that have yet to be explored.

Bibliography

[1] V. Adetola and M. Guay, "Parameter convergence in adaptive extremum-seeking control," *Automatica*, vol. 43, no. 1, pp. 105–110, 2007.

[2] V. Adetola and M. Guay, "Finite-time parameter estimation in adaptive control of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 807–811, 2008.

[3] V. Adetola and M. Guay, "Integration of real-time optimization and model predictive control," *Journal of Process Control*, vol. 20, no. 2, pp. 125–133, 2010.

[4] K. B. Ariyur and M. Krstić, "Analysis and design of multivariable extremum seeking," in *Proceedings of the 2002 American Control Conference (ACC)*. IEEE, Anchorage, Alaska; May 2002, pp. 2903–2908.

[5] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 6, pp. 926–939, 1998.

[6] G. Bao, C. Lu, Z. Yuan, and Y. Han, "Load shift real-time optimization strategy of battery energy storage system based on dynamic programming," *Dianli Xitong Zidonghua (Automation of Electric Power Systems)*, vol. 36, no. 12, pp. 11–16, 2012.

[7] L. Becker and W. W. G. Yeh, "Optimization of real time operation of a multiple-reservoir system," *Water Resources Research*, vol. 10, no. 6, pp. 1107–1112, 1974.

[8] R. Becker, R. King, R. Petz, and W. Nitsche, "Adaptive closed-loop separation control on a high-lift configuration using extremum seeking," *AIAA journal*, vol. 45, no. 6, pp. 1382–1392, 2007.

[9] P. Binetti, K. B. Ariyur, M. Krstić, and F. Bernelli, "Control of formation flight via extremum seeking," in *Proceedings of the 2002 American Control Conference (ACC)*, vol. 4. IEEE, Anchorage, Alaska; May 2002, pp. 2848–2853.

[10] P. Binetti, K. B. Ariyur, M. Krstić, and F. Bernelli, "Formation flight optimization using extremum seeking feedback," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 1, pp. 132–142, 2003.

[11] G. D. Birkhoff, "General mean value and remainder theorems with applications to mechanical differentiation and quadrature," *Transactions of the American Mathematical Society*, vol. 7, no. 1, pp. 107–136, 1906.

[12] A. N. Bishop and M. Basiri, "Bearing-only triangular formation control on the plane and the sphere," in *Proceedings of the 18th Mediterranean Conference on Control & Automation (MED).* IEEE, Marrakesh, Morocco; Jun. 2010, pp. 790–795.

[13] N. Bizon, "On tracking robustness in adaptive extremum seeking control of the fuel cell power plants," *Applied Energy*, vol. 87, no. 10, pp. 3115–3130, 2010.

[14] P. F. Blackman, "Extremum-seeking regulators," in *Proceedings of An Exposition of Adaptive Control*, London, England; 1961, pp. 36–50.

[15] J. A. Bondy and U. S. R. Murty, *Graph theory with applications.* Macmillan London, 1976, vol. 290.

[16] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, "Voronoi coverage of non-convex environments with a group of networked robots," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2010, pp. 4982–4989.

[17] S. L. Brunton, C. W. Rowley, S. R. Kulkarni, and C. Clarkson, "Maximum power point tracking for photovoltaic optimization using ripple-based extremum seeking control," *IEEE Transactions on Power Electronics*, vol. 25, no. 10, pp. 2531–2540, 2010.

[18] C. Cabal, C. Alonso, A. Cid-Pastor, B. Estibals, L. Seguier, R. Leyva, G. Schweitz, and J. Alzieu, "Adaptive digital mppt control for photovoltaic applications," in *Proceedings of the 2007 IEEE International Symposium on Industrial Electronics (ISIE).* IEEE, Vigo, Spain; Jun. 2007, pp. 2414–2419.

[19] Y. A. Chang and S. J. Moura, "Air flow control in fuel cell systems: an extremum seeking approach," in *Proceedings of the 2009 American Control Conference (ACC).* IEEE, St. Louis, Missouri; Jun. 2009, pp. 1052–1059.

[20] Z. Chen, L. Wu, and Y. Fu, "Real-time price-based demand response management for residential appliances via stochastic optimization and robust optimization," *IEEE transactions on Smart grid*, vol. 3, no. 4, pp. 1822–1831, 2012.

[21] D. F. Chichka, J. L. Speyer, C. Fanti, and C.-G. Park, "Peak-seeking control for drag reduction in formation flight," *Journal of guidance, control, and dynamics*, vol. 29, no. 5, pp. 1221–1230, 2006.

[22] J.-Y. Choi, M. Krstić, K. B. Ariyur, and J. S. Lee, "Extremum seeking control for discrete-time systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 2, pp. 318–323, 2002.

[23] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader–follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, no. 5, pp. 1343–1349, 2008.

[24] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2. IEEE, Washington, D.C.; May 2002, pp. 1327–1332.

[25] P. Cougnon, D. Dochain, M. Guay, and M. Perrier, "On-line optimization of fedbatch bioreactors by adaptive extremum seeking control," *Journal of Process Control*, vol. 21, no. 10, pp. 1526–1532, 2011.

[26] J. Creaby, Y. Li, and J. Seem, "Maximizing wind turbine energy capture using multivariable extremum seeking control," *Wind Engineering*, vol. 33, no. 4, pp. 361–387, 2009.

[27] A. Dalvi and M. Guay, "Control and real-time optimization of an automotive hybrid fuel cell power system," *Control Engineering Practice*, vol. 17, no. 8, pp. 924–938, 2009.

[28] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 813–825, 2002.

[29] D. Dehaan and M. Guay, "Extremum-seeking control of state-constrained nonlinear systems," *Automatica*, vol. 41, no. 9, pp. 1567–1574, 2005.

[30] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes

governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, no. 4, pp. 577–585, 2002.

[31] C. S. Draper and Y.-T. Li, *Principles of optimalizing control systems and an application to the internal combustion engine.* American Society of Mechanical Engineers, 1951.

[32] S. E. Dreyfus, "Dynamic programming and the calculus of variations," *Journal of Mathematical Analysis and Applications*, vol. 1, no. 2, pp. 228–239, 1960.

[33] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: applications and algorithms," *SIAM Review*, vol. 41, no. 4, pp. 637–676, 1999.

[34] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 947–951, 2001.

[35] P. Eykhoff, "Adaptive and optimalizing control systems," *IRE Transactions on Automatic Control*, vol. 5, no. 2, pp. 148–151, 1960.

[36] P. Eykhoff and O. J. M. Smith, "Optimalizing control with process-dynamics identification," *IRE Transactions on Automatic Control*, vol. 7, no. 2, pp. 140–155, 1962.

[37] Federal Communications Commission, "Types of broadband connections," 2016. [Online]. Available: http://www.broadband.gov/broadband_types.html

[38] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *Proceedings of the 45th IEEE Conference on Decision and Control (CDC)*, San Diego, California; Dec. 2006, pp. 398–403.

[39] P. Frihauf, M. Krstić, and T. Başar, "Nash equilibrium seeking with infinitely-many players," in *Proceedings of the 2011 American Control Conference (ACC).* IEEE, San Francisco, California; Jul. 2011, pp. 3059–3064.

[40] P. Frihauf, M. Krstić, and T. Başar, "Nash equilibrium seeking in noncooperative games," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1192–1207, 2012.

[41] A. Ghaffari, M. Krstić, and D. Nešić, "Multivariable newton-based extremum seeking," *Automatica*, vol. 48, no. 8, pp. 1759–1767, 2012.

[42] A. Ghaffari, M. Krstić, and S. Seshagiri, "Power optimization and control in wind energy conversion systems using extremum seeking," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1684–1695, 2014.

[43] N. Ghods, P. Frihauf, and M. Krstić, "Multi-agent deployment in the plane using stochastic extremum seeking," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC).* IEEE, Atlanta, Georgia; Dec. 2010, pp. 5505–5510.

[44] Google, "Loon for all – project loon," 2016. [Online]. Available: http://www.google.com/loon/

[45] M. Guay, "A time-varying extremum-seeking control approach for discrete-time systems," *Journal of Process Control*, vol. 24, no. 3, pp. 98–112, 2014.

[46] M. Guay and D. J. Burns, "A comparison of extremum seeking algorithms applied to vapor compression system optimization," in *Proceedings of the 2014 American Control Conference (ACC).* IEEE, Portland, Oregon; Jun. 2014, pp. 1076–1081.

[47] M. Guay, S. Dhaliwal, and D. Dochain, "A time-varying extremum-seeking control approach," in *Proceedings of the 2013 American Control Conference (ACC).* IEEE, Washington, D.C.; Jun. 2013, pp. 2643–2648.

[48] M. Guay and D. Dochain, "A proportional-integral extremum-seeking controller design technique," in *Proceedings of the 19th IFAC World Congress*, Aug. 2014, pp. 377–382.

[49] M. Guay, D. Dochain, and M. Perrier, "Adaptive extremum seeking control of continuous stirred tank bioreactors with unknown growth kinetics," *Automatica*, vol. 40, no. 5, pp. 881–888, 2004.

[50] M. Guay, I. Vandermeulen, S. Dougherty, and P. J. Mclellan, "Distributed extremum-seeking control over networks of dynamic agents," in *Proceedings of the 2015 American Control Conference (ACC).* IEEE, Chicago, Illinois; Jul. 2015, pp. 159–164.

[51] M. Guay, I. Vandermeulen, S. Dougherty, and P. J. McLellan, "Distributed extremum-seeking control over networks of unstable dynamic agents," in *Proceedings of the 9th International Symposium on Advanced Control of Chemical Process (ADCHEM).* IFAC, Whistler, British Columbia; Jun. 2015, pp. 693–697.

[52] M. Guay and T. Zhang, "Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainties," *Automatica*, vol. 39, no. 7, pp. 1283–1293, 2003.

[53] A. Gusrialdi, S. Hirche, T. Hatanaka, and M. Fujita, "Voronoi based coverage control with anisotropic sensors," in *Proceedings of the 2008 American Control Conference (ACC).* IEEE, Seattle, Washington; Jun. 2008, pp. 736–741.

[54] T. R. Henderson and R. H. Katz, "Transport protocols for internet-compatible satellite networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 2, pp. 326–344, 1999.

[55] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS)*, IEEE. Springer, Fukuoka, Japan; Jun. 2002, pp. 299–308.

[56] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 519–528, 2005.

[57] I. I. Hussein and D. M. Stipanović, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 642–657, 2007.

[58] Internet Live Stats, "Internet users," 2016. [Online]. Available: http://www.internetlivestats.com/internet-users/

[59] Internet.org by Facebook, "Connectivity lab," 2016. [Online]. Available: https://internet.org/projects

[60] K. Johnson and G. Fritsch, "Assessment of extremum seeking control for wind farm energy production," *Wind Engineering*, vol. 36, no. 6, pp. 701–716, 2012.

[61] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu, "The analysis of a simple k-means clustering algorithm," in *Proceedings of the 16th ACM Symposium on Computational Geometry.* ACM, Hong Kong, China; Jun. 2000, pp. 100–109.

[62] V. V. Kazakevich, "On extremum seeking," Ph.D. dissertation, Moscow High Technical University, 1944.

[63] V. V. Kazakevich, "On the process of extremum control of systems having inertia in the presence of perturbations," *Soviet Physics Doklady*, vol. 4, p. 578, 1959.

[64] V. V. Kazakevich, "Theory of ideal model of extremum controller," *Avtomatika i Telemekhanika*, vol. 21, no. 4, pp. 489–505, 1960.

[65] V. V. Kazakevich, "On algorithms of quickest search in the pulsed extremal control systems," *Avtomatika i Telemekhanika*, no. 12, pp. 71–80, 1966.

[66] S. S. Kia, J. Cortés, and S. Martínez, "Dynamic average consensus under limited control authority and privacy requirements," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 13, pp. 1941–1966, 2015.

[67] N. J. Killingsworth, S. M. Aceves, D. L. Flowers, F. Espinosa-Loza, and M. Krstić, "HCCI engine combustion-timing control: Optimizing gains and fuel consumption via extremum seeking," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 6, pp. 1350–1361, 2009.

[68] M. Komatsu, H. Miyamoto, H. Ohmori, and A. San, "Output maximization control of wind turbine based on extremum control strategy," in *Proceedings of the 2001 American Control Conference (ACC)*, vol. 2. IEEE, Arlington, Virginia; Jun. 2001, pp. 1739–1740.

[69] M. Krstić, P. Frihauf, J. Krieger, and T. Başar, "Nash equilibrium seeking with finitely and infinitely-many players," in *Nonlinear Control Systems*, 2010, pp. 1086–1091.

[70] M. Krstić, I. Kanellakopoulos, and P. Kokotović, *Nonlinear and Adaptive Control Design.* Toronto: John Wiley and Sons Inc, 1995.

[71] M. Krstić and H.-H. Wang, "Stability of extremum seeking feedback for general nonlinear dynamic systems," *Automatica*, vol. 36, no. 4, pp. 595–601, 2000.

[72] K. Kvaternik and L. Pavel, "An analytic framework for decentralized extremum seeking control," in *Proceedings of the 2012 American Control Conference (ACC).* IEEE, Montréal, Québec; Jun. 2012, pp. 3371–3376.

[73] M. Leblanc, "Sur lélectrification des chemins de fer au moyen de courants alternatifs de fréquence élevée," *Revue Générale de l'Électricité*, vol. 12, pp. 275–277, 1922.

[74] G. Leibon and D. Letscher, "Delaunay triangulations and voronoi diagrams for riemannian manifolds," in *Proceedings of the 16th ACM Symposium on Computational geometry.* ACM, Hong Kong, China; Jun. 2000, pp. 341–349.

[75] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proceedings of the 40th IEEE Conference on Decision and Control (CDC)*, vol. 3. IEEE, Orlando, Florida; Dec. 2001, pp. 2968–2973.

[76] M. A. Lewis and K.-H. Tan, “High precision formation control of mobile robots using virtual structures,” *Autonomous Robots*, vol. 4, no. 4, pp. 387–403, 1997.

[77] R. Leyva, C. Alonso, I. Queinnec, A. Cid-Pastor, D. Lagrange, and L. Martínez-Salamero, “MPPT of photovoltaic systems using extremum-seeking control,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 249–258, 2006.

[78] C. Li, Z. Qu, and M. A. Ingram, “Distributed extremum seeking and cooperative control for mobile communication,” in *Proceedings of the 50th IEEE Conference on Decision and Control (CDC).* IEEE, Orlando, Florida; Dec. 2011, pp. 4510–4515.

[79] P. Li, Y. Li, and J. E. Seem, “Efficient operation of air-side economizer using extremum seeking control,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 132, no. 3, pp. 1–10, 2010.

[80] T. Li and J. Zhang, “Consensus conditions of multi-agent systems with time-varying topologies and stochastic communication noises,” *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2043–2057, 2010.

[81] W. Li and C. G. Cassandras, “Distributed cooperative coverage control of sensor networks,” in *Proceedings of the 44th IEEE Conference on Decision and Control (CDC).* IEEE, Seville, Spain; Dec. 2005, pp. 2542–2547.

[82] Y. Li, M. A. Rotea, G. T. C. Chiu, L. G. Mongeau, and I.-S. Paek, “Extremum seeking control of a tunable thermoacoustic cooler,” *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 527–536, 2005.

[83] P. Lin and Y. Jia, “Consensus of second-order discrete-time multi-agent systems with nonuniform time-delays and dynamically changing topologies,” *Automatica*, vol. 45, no. 9, pp. 2154–2158, 2009.

[84] L. Ljung and T. Söderström, “Theory and practice of recursive identification,” 1983.

[85] S. P. Lloyd, “Least squares quantization in pcm,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[86] N. A. Lynch, *Distributed algorithms.* Morgan Kaufmann, 1996.

[87] N. I. Marcos, M. Guay, D. Dochain, and T. Zhang, “Adaptive extremum-seeking control of a continuous stirred tank bioreactor with Haldane’s kinetics,” *Journal of Process Control*, vol. 14, no. 3, pp. 317–328, 2004.

[88] S. M. Meerkov, "Asymptotic methods for investigating a class of forced states in extremal systems," *Automation and Remote Control*, vol. 12, no. 1916, p. 1920, 1967.

[89] A. Michel and D. Porter, "Practical stability and finite-time stability of discontinuous systems," *IEEE transactions on circuit theory*, vol. 19, no. 2, pp. 123–129, 1972.

[90] W. H. Moase, C. Manzie, and M. J. Brear, "Newton-like extremum-seeking for the control of thermoacoustic instability," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2094–2105, 2010.

[91] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[92] I. S. Morosanov, "Method of extremum control," *Automatic and Remote Control*, vol. 18, pp. 1077–1092, 1957.

[93] E. Moshksar and M. Guay, "Invariant manifold approach for time-varying extremum-seeking control problem," in *Proceedings of the 19th IFAC World Congress*. IFAC, Cape Town, South Africa; Aug. 2014, pp. 9129–9134.

[94] NASA Socioeconomic Data and Applications Center, "Population density grid, v1 (1990, 1995, 2000)," 2016. [Online]. Available: http://sedac.ciesin.columbia.edu/data/set/grump-v1-population-density

[95] National Oceanic and Atmospheric Administration, "NOAA operation model archive and distribution system," 2016. [Online]. Available: http://nomads.ncep.noaa.gov/

[96] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[97] D. Nešić, A. Mohammadi, and C. Manzie, "A systematic approach to extremum seeking based on parameter estimation," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*. IEEE, Dec. 2010.

[98] Official Google Blog, "Introducing project loon: Balloon-powered internet access," 2013. [Online]. Available: http://googleblog.blogspot.ca/2013/06/introducing-project-loon.html

[99] R. Olfati-Saber, A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[100] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[101] T. Pan, Z. Ji, and Z. Jiang, "Maximum power point tracking of wind energy conversion systems based on sliding mode extremum seeking control," in *Proceedings of the IEEE Energy 2030 Conference*, Atlanta, Georgia; Nov. 2008, pp. 279–283.

[102] Y. Pan, T. Acarman, and Ü. Özgüner, "Nash solution by extremum seeking control approach," in *Proceedings of the 41st IEEE Conference on Decision and Control (CDC)*, vol. 1. IEEE, Las Vegas, Nevada; Dec. 2002, pp. 329–334.

[103] K. S. Peterson and A. G. Stefanopoulou, "Extremum seeking control for soft landing of an electromechanical valve actuator," *Automatica*, vol. 40, no. 6, pp. 1063–1069, 2004.

[104] D. Popović, M. Jankovic, S. Magner, and A. Teel, "Extremum seeking methods for optimization of variable cam timing engine operation," in *Proceedings of the 2003 American Control Conference (ACC)*, vol. 4. IEEE, Denver, Colorado; Jun. 2003, pp. 3136–3141.

[105] J. Poveda and N. Quijano, "Distributed extremum seeking for real-time resource allocation," in *Proceedings of the 2013 American Control Conference (ACC)*. IEEE, Washington, D.C.; Jun. 2013, pp. 2772–2777.

[106] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proceedings of the 2005 American Control Conference (ACC)*. IEEE, Portland, Oregon; Jun. 2005, pp. 1859–1864.

[107] R. J. Renka, "Algorithm 772: STRIPACK: Delaunay triangulation and voronoi diagram on the surface of a sphere," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 3, pp. 416–434, 1997.

[108] M. A. Rotea, "Analysis of multivariable extremum seeking algorithms," in *Proceedings of the 2000 American Control Conference (ACC)*, no. 6. IEEE, Chicago, Illinois; Jun. 2000, pp. 433–437.

[109] L. Rundqwist, "Anti-reset windup for PID controllers," Ph.D. dissertation, Department of Automatic Control, Lund Institute of Technology, May 1991.

[110] R. Scattolini, “Architectures for distributed and hierarchical model predictive control—a review,” *Journal of Process Control*, vol. 19, no. 5, pp. 723–731, 2009.

[111] M. Schwager, F. Bullo, D. Skelly, and D. Rus, “A ladybug exploration strategy for distributed adaptive coverage control,” in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, Pasadena, California; May 2008, pp. 2346–2353.

[112] M. Schwager, D. Rus, and J.-J. Slotine, “Decentralized, adaptive coverage control for networked robots,” *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.

[113] J. Shao, G. Xie, and L. Wang, “Leader-following formation control of multiple mobile vehicles,” *IET Control Theory & Applications*, vol. 1, no. 2, pp. 545–552, 2007.

[114] A. C. Sniderman, M. E. Broucke, and G. M. T. D’Eleuterio, “Formation control of balloons: A block circulant approach,” in *Proceedings of the 2015 American Control Conference (ACC).* IEEE, Chicago, Illinois; Jul. 2015, pp. 1463–1468.

[115] O. Solon, “Google installing hundreds of internet-enabled balloons in Indonesia,” The Guardian, Oct. 2015. [Online]. Available: https://www.theguardian.com/technology/2015/oct/28/google-installing-20000-internet-enabled-balloons-in-indonesia

[116] M. S. Stanković and D. M. Stipanović, “Stochastic extremum seeking with applications to mobile sensor networks,” in *Proceedings of the 2009 American Control Conference (ACC).* IEEE, St. Louis, Missouri; Jun. 2009, pp. 5622–5627.

[117] J. Sternby, “Extremum control systems?? an area for adaptive control?” in *Proceedings of the 1980 Joint Automatic Control Conference (JACC)*, no. 17, San Francisco, California; Aug. 1980, p. 8.

[118] P. Stone and M. Veloso, “Multiagent systems: A survey from a machine learning perspective,” *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, 2000.

[119] Y. Tan, W. H. Moase, C. Manzie, D. Nešić, and I. Mareels, “Extremum seeking from 1922 to 2010,” in *Proceedings of the 29th Chinese Control Conference (CCC).* IEEE, Beijing, China; Jul. 2010, pp. 14–26.

[120] Y. Tan, D. Nešić, and I. Mareels, “On non-local stability properties of extremum seeking control,” *Automatica*, vol. 42, no. 6, pp. 889–903, 2006.

[121] M. Tanelli, A. Astolfi, and S. M. Savaresi, "Non-local extremum seeking control for active braking control systems," in *Proceedings of the 2006 IEEE International Conference on Control Applications (CCA).* IEEE, Munich, Germany; Oct. 2006, pp. 891–896.

[122] H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443–455, 2004.

[123] Y. Tian and C. Liu, "Consensus of multi-agent systems with diverse input and communication delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2122–2128, 2008.

[124] M. Titica, D. Dochain, and M. Guay, "Adaptive extremum seeking control of fed-batch bioreactors," *European Journal of Control*, vol. 9, no. 6, pp. 618–631, 2003.

[125] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Cambridge lab for Information and Decision Systems, Massachusetts Institute of Technology, Dec. 1984.

[126] I. Vandermeulen, M. Guay, and P. J. McLellan, "Discrete-time distributed extremum-seeking control over networks with unstable dynamics," in *Proceedings of the 10th IFAC Symposium on Nonlinear Control Systems (NOLCOS).* IFAC, Monterey, California; Aug. 2016.

[127] I. Vandermeulen, M. Guay, and P. J. McLellan, "Formation control of high-altitude balloons by distributed extremum seeking control," in *Proceedings of the 2016 American Control Conference (ACC).* IEEE, Boston, Massachusetts; Jul. 2016.

[128] I. Vandermeulen, M. Guay, and P. J. McLellan, "Formation control of high-altitude balloons experiencing real wind currents by discrete-time distributed extremum seeking control," in *Proceedings of the 55th IEEE Conference on Decision and Control (CDC).* IEEE, Las Vegas, Nevada; Dec. 2016.

[129] G. Voronoi, "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites." *Journal für die reine und angewandte Mathematik*, vol. 133, pp. 97–178, 1908.

[130] D. D. Šiljak, *Decentralized control of complex systems.* Courier Corporation, 2011.

[131] H.-H. Wang, M. Krstić, and G. Bastin, "Optimizing bioreactors by extremum seeking," *International Journal of Adaptive Control and Signal Processing*, vol. 13, no. 8, pp. 651–669, 1999.

[132] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[133] W. K. Weiss, D. J. Burns, and M. Guay, "Realtime optimization of mpc setpoints using time-varying extremum seeking control for vapor compression machines," in *Proceedings of the 15th International Refrigeration and Air Conditioning Conference*, Purdue University; Jul. 2014, paper 2273, pp. 1–10.

[134] D. S. Wilks and R. L. Wilby, "The weather generation game: a review of stochastic weather models," *Progress in Physical Geography*, vol. 23, no. 3, pp. 329–357, 1999.

[135] Wired, "Internet by satellite is a space race with no winers," 2015. [Online]. Available: http://www.wired.com/2015/06/elon-musk-space-x-satellite-internet/

[136] Worldometers, "World population by year," 2016. [Online]. Available: http://www.worldometers.info/world-population/world-population-by-year/

[137] X. Xia, D. Li, C. Sun, and L. Ruan, "Transient thermal behavior of stratospheric balloons at float conditions," *Advances in Space Research*, vol. 46, no. 9, pp. 1184–1190, 2010.

[138] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1804–1816, 2008.

[139] H.-T. Yau and C.-H. Wu, "Comparison of extremum-seeking control techniques for maximum power point tracking in photovoltaic systems," *Energies*, vol. 4, no. 12, pp. 2180–2195, 2011.

[140] H. Yu and Ü. Özgüner, "Extremum-seeking control strategy for abs system with time delay," in *Proceedings of the 2002 American Control Conference (ACC)*, vol. 5. IEEE, Anchorage, Alaska; May 2002, pp. 3753–3758.

[141] C. Zhang and R. Ordóñez, "Numerical optimization-based extremum seeking control with application to ABS design," *IEEE Transactions on Automatic Control*, vol. 52, no. 3, pp. 454–467, 2007.

[142] C. Zhang and R. Ordóñez, "Robust and adaptive design of numerical optimization-based extremum seeking control," *Automatica*, vol. 45, no. 3, pp. 634–646, 2009.

[143] T. Zhang, M. Guay, and D. Dochain, "Adaptive extremum seeking control of continuous stirred-tank bioreactors," *AIChE Journal*, vol. 49, no. 1, pp. 113–123, 2003.

[144] Y. Zhang, D. Monder, and J. F. Forbes, "Real-time optimization under parametric uncertainty: a probability constrained approach," *Journal of Process control*, vol. 12, no. 3, pp. 373–389, 2002.

[145] M. Zhong and C. G. Cassandras, "Distributed coverage control and data collection with mobile sensor networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2445–2455, 2011.

[146] Z. Zhong, H. Huo, X. Zhu, G. Cao, and Y. Ren, "Adaptive maximum power point tracking control of fuel cell power plants," *Journal of Power Sources*, vol. 176, no. 1, pp. 259–269, 2008.

[147] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

Appendix A

Proof of Theorem 3.1

In the first part of the proof, we consider the consensus dynamics over the network. We first write the consensus algorithm as

$$\begin{bmatrix} \epsilon\dot{\widehat{\boldsymbol{J}}} \\ \epsilon\dot{\boldsymbol{\rho}} \end{bmatrix} = \begin{bmatrix} -\kappa_0\boldsymbol{I} - \kappa_P\boldsymbol{L} & \kappa_I\boldsymbol{L} \\ -\kappa_I\boldsymbol{L} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{J}} \\ \boldsymbol{\rho} \end{bmatrix} + \begin{bmatrix} \kappa_0\boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{h}(\boldsymbol{u}) \tag{A.1}$$

where ϵ is a small positive singular perturbation parameter. We let $\widetilde{J}_i = J - p\widehat{J}_i$. By Assumption Assumption 3.3, the graph associated with the network is undirected and connected, so $\operatorname{rank}(\boldsymbol{L}) = p - 1$. Then using a theorem from [38], the consensus algorithm achieves an exponential rate of convergence. That is, each local consensus estimate converges to $\frac{1}{p}J$ and there exist constants $\nu_0, k_c > 0$ such that:

$$\left\|\widetilde{J}_i\right\| \leq \nu_0 \exp\left(-k_c \frac{t}{\epsilon}\right) \tag{A.2}$$

Thus, if $\epsilon \to 0$ then $p\widehat{J}_i \to J$ for all i and each agent has a local measurement of the total augmented cost. Since the total costs estimates converge, we can express the estimates as

$$p\widehat{J}_i = J(x) + p\nu_i(t) \tag{A.3}$$

where, by the properties of the dynamic consensus, $\nu_i(t)$ is an exponentially decaying measurement error.

In the second part of the proof, we consider parameter estimation using the total cost estimation instead of the true total cost. The proof of convergence for the parameter estimates follows the arguments presented in [47]. Since we are interested in parameter convergence, we use the following Lyapunov function

$$V_{\text{PE}} = \sum_{i=1}^{p} \left(\frac{1}{2}\widetilde{\eta}_i^\top \widetilde{\eta}_i + \frac{1}{2}\widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \widetilde{\boldsymbol{\theta}}_i \right) \tag{A.4}$$

where $\widetilde{\eta}_i = \eta_i - \widehat{\eta}_i$ and $\widetilde{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i - \widehat{\boldsymbol{\theta}}_i$. In this Lyapunov function, the parameter estimate deviations are summed over all of the agents. The Lyapunov function reaches a minimum at $V_{\text{PE}} = 0$ when all p agents have accurate parameter estimates $\widehat{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i$ and $\widehat{\eta}_i = \eta_i$. Since the local agents are only using local estimates of the total cost, we must use the perturbed measurement for each agent $\widehat{J}_i = \frac{1}{p}J(x) + \nu_i(t)$ and consider the error $e_i = \widehat{J}_i - \widehat{y}_i$. Since we must take into account the effect of $\nu_i(t)$, the error dynamics can be written as

$$\dot{e}_i = -Ke_i - \boldsymbol{w}_i^\top \dot{\widehat{\boldsymbol{\theta}}}_i + \widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\phi}_i + \dot{\nu}_i. \tag{A.5}$$

Using (A.5), the corresponding η_i and $\widetilde{\eta}_i$ dynamics are thus given by

$$\dot{\eta}_i = -K\eta_i + \dot{\nu}_i - \boldsymbol{w}_i^\top \dot{\boldsymbol{\theta}}_i \tag{A.6}$$

$$\dot{\widetilde{\eta}}_i = -K\widetilde{\eta}_i + \dot{\nu}_i - \boldsymbol{w}_i^\top \dot{\boldsymbol{\theta}}_i. \tag{A.7}$$

The dynamics of the parameter estimation error $\widetilde{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i - \widehat{\boldsymbol{\theta}}_i$ are

$$\dot{\widetilde{\theta}}_i = \dot{\boldsymbol{\theta}}_i - \boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{w}_i(e_i - \widehat{\eta}_i) - \sigma_1\widehat{\boldsymbol{\theta}}_i\right). \tag{A.8}$$

By differentiating (A.4) and using (3.28), (3.30), (A.7), and (A.8), the derivative of the Lyapunov function along system trajectories is

$$\begin{aligned}\dot{V}_{\mathrm{PE}} \leq \sum_{i=1}^{p}\Bigg(& -\widetilde{\eta}_i^\top K\widetilde{\eta}_i - \widetilde{\eta}_i^\top \boldsymbol{w}_i^\top \dot{\boldsymbol{\theta}}_i + \widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \dot{\boldsymbol{\theta}}_i + \widetilde{\eta}_i^\top \dot{\nu}_i - \frac{K_T}{2}\widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \widetilde{\boldsymbol{\theta}}_i \\ & + \frac{\sigma_1}{2}\widetilde{\boldsymbol{\theta}}_i^\top \widetilde{\boldsymbol{\theta}}_i - \widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{w}_i(e_i - \widehat{\eta}_i) + \frac{1}{2}\widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{w}_i \boldsymbol{w}_i^\top \widetilde{\boldsymbol{\theta}}_i + \sigma_1 \widetilde{\boldsymbol{\theta}}_i^\top \widehat{\boldsymbol{\theta}}_i\Bigg).\end{aligned} \tag{A.9}$$

From the definitions of η_i and $\widetilde{\eta}_i$, we have that $\boldsymbol{w}_i^\top \widetilde{\boldsymbol{\theta}}_i = e_i - \eta_i = e_i - \widehat{\eta}_i - \widetilde{\eta}_i$. Making this substitution and simplifying results in

$$\begin{aligned}\dot{V}_{\mathrm{PE}} \leq \sum_{i=1}^{p}\Bigg(& -\widetilde{\eta}_i^\top K\widetilde{\eta}_i - \widetilde{\eta}_i^\top \boldsymbol{w}_i^\top \dot{\boldsymbol{\theta}}_i + \widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \dot{\boldsymbol{\theta}}_i + \widetilde{\eta}_i^\top \dot{\nu}_i - \frac{K_T}{2}\widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \widetilde{\boldsymbol{\theta}}_i \\ & + \frac{\sigma_1}{2}\widetilde{\boldsymbol{\theta}}_i^\top \widetilde{\boldsymbol{\theta}}_i - \frac{1}{2}(e_i - \widehat{\eta}_i)^2 + \frac{1}{2}\widetilde{\eta}_i^\top \widetilde{\eta}_i + \sigma_1 \widetilde{\boldsymbol{\theta}}_i^\top \widehat{\boldsymbol{\theta}}_i\Bigg).\end{aligned} \tag{A.10}$$

Noting that $\boldsymbol{\theta}_i = \widehat{\boldsymbol{\theta}}_i + \widetilde{\boldsymbol{\theta}}_i$ and that $-\frac{1}{2}(e_i - \widehat{\eta}_i)^2 \leq 0$, one can rewrite the above inequality as

$$\begin{aligned}\dot{V}_{\mathrm{PE}} \leq \sum_{i=1}^{p} \Bigg(&- \widetilde{\eta}_i^\top \left(K - \frac{1}{2}\right) \widetilde{\eta}_i - \widetilde{\eta}_i^\top \boldsymbol{w}_i^\top \dot{\boldsymbol{\theta}}_i + \widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \dot{\boldsymbol{\theta}}_i + \widetilde{\eta}_i^\top \dot{\nu}_i \\ &- \frac{K_T}{2} \widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \widetilde{\boldsymbol{\theta}}_i - \left(\sigma_1 - \frac{\sigma_1}{2}\right) \widetilde{\boldsymbol{\theta}}_i^\top \widetilde{\boldsymbol{\theta}}_i + \sigma_1 \widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\theta}_i \Bigg).\end{aligned} \tag{A.11}$$

We can bound the indefinite terms by completing the squares to obtain

$$\begin{aligned}\dot{V}_{\mathrm{PE}} \leq \sum_{i=1}^{p} \Bigg(&- \widetilde{\eta}_i^\top \left(K - \frac{1}{2}\right) \widetilde{\eta}_i + \frac{a_{17}}{2} \widetilde{\eta}_i^\top \boldsymbol{w}_i^\top \boldsymbol{w}_i \widetilde{\eta}_i + \frac{1}{2a_{17}} \dot{\boldsymbol{\theta}}_i^\top \dot{\boldsymbol{\theta}}_i \\ &+ \frac{a_{18}}{2} \widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \widetilde{\boldsymbol{\theta}}_i + \frac{1}{2a_{18}} \dot{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \dot{\boldsymbol{\theta}}_i + \frac{a_{19}}{2} \widetilde{\eta}_i^\top \widetilde{\eta}_i + \frac{1}{2a_{19}} \dot{\nu}_i^\top \dot{\nu}_i \\ &- \frac{K_T}{2} \widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \widetilde{\boldsymbol{\theta}}_i - \left(\sigma_1 - \frac{\sigma_1}{2}\right) \widetilde{\boldsymbol{\theta}}_i^\top \widetilde{\boldsymbol{\theta}}_i + \frac{\sigma_1}{2} \widetilde{\boldsymbol{\theta}}_i^\top \widetilde{\boldsymbol{\theta}}_i + \frac{\sigma_1}{2} \boldsymbol{\theta}_i^\top \boldsymbol{\theta}_i \Bigg)\end{aligned} \tag{A.12}$$

where a_{17}, a_{18} and a_{19} are strictly positive constants. It is assumed without loss of generality that $K_T > a_{18}$ so $K_T - a_{18} > 0$. Then collecting terms, we have

$$\begin{aligned}\dot{V}_{\mathrm{PE}} \leq \sum_{i=1}^{p} \Bigg(&- \widetilde{\eta}_i^\top \left(K - \frac{1}{2} - \frac{a_{17}}{2} \boldsymbol{w}_i^\top \boldsymbol{w}_i - \frac{a_{19}}{2}\right) \widetilde{\eta}_i + \frac{1}{2a_{17}} \dot{\boldsymbol{\theta}}_i^\top \dot{\boldsymbol{\theta}}_i \\ &+ \frac{1}{2a_{18}} \dot{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \dot{\boldsymbol{\theta}}_i + \frac{1}{2a_{19}} \dot{\nu}_i^\top \dot{\nu}_i - \frac{K_T - a_{18}}{2} \widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \widetilde{\boldsymbol{\theta}}_i + \frac{\sigma_1}{2} \boldsymbol{\theta}_i^\top \boldsymbol{\theta}_i \Bigg).\end{aligned} \tag{A.13}$$

By definition, $\boldsymbol{\Sigma}_i$ is positive definite; however as it is time-varying, it is useful to show that it has upper and lower bounds so that we can bound the terms $+\frac{1}{2a_{17}} \dot{\boldsymbol{\theta}}_i^\top \dot{\boldsymbol{\theta}}_i$ and $-\frac{K_T - a_{18}}{2} \widetilde{\boldsymbol{\theta}}_i^\top \boldsymbol{\Sigma}_i \widetilde{\boldsymbol{\theta}}_i$ using constants. The boundedness of $\boldsymbol{\Sigma}_i$ is shown using the same approach as is presented in [47]. By integrating and using Assumption Assumption 3.4, for $t \geq T$ we have:

$$\boldsymbol{\Sigma}_i = \mathrm{e}^{-K_T t} \boldsymbol{\Sigma}_i(0) + \int_0^t \mathrm{e}^{-K_T (t-\tau)} \left(\boldsymbol{w}_i(\tau) \boldsymbol{w}_i^\top(\tau) + \sigma_1 \boldsymbol{I} \right) \mathrm{d}\tau$$

$$
\begin{aligned}
&\geq \int_{t-T}^{t} \mathrm{e}^{-K_T(t-\tau)} \left(\boldsymbol{w}_i(\tau)\boldsymbol{w}_i^\top(\tau) + \sigma_1 \boldsymbol{I}\right) \mathrm{d}\tau \\
&\geq \mathrm{e}^{-K_T T} \left(\gamma_w^- + \sigma_1 T\right) \boldsymbol{I}. \qquad \text{(A.14)}
\end{aligned}
$$

When $t < T$ we can upper bound $\boldsymbol{\Sigma}_i$ by $\boldsymbol{\Sigma}_i \geq \mathrm{e}^{-K_T T}\sigma_0 \boldsymbol{I}$. Therefore for any $t > 0$,

$$
\boldsymbol{\Sigma}_i \geq \min\{\sigma_0, \gamma_w^- + \sigma_1 T\}\mathrm{e}^{-K_T T}\boldsymbol{I}. \qquad \text{(A.15)}
$$

The Lipschitz projection operator used to define the update law for $\widehat{\boldsymbol{\theta}}_i$ results in the boundedness of $\widehat{\boldsymbol{\theta}}_i$. Since $\widehat{\boldsymbol{\theta}}_i$ is bounded, $\boldsymbol{u}_i$ is bounded and $\boldsymbol{w}_i$ is also bounded. Since $\boldsymbol{w}_i$ is bounded there exists $\gamma_w^+ > 0$ such that $\boldsymbol{w}_i(t)\boldsymbol{w}_i^\top(t) < \gamma_w^+ \boldsymbol{I}$ for all $t > 0$. Therefore we can write:

$$
\begin{aligned}
\boldsymbol{\Sigma}_i &= \mathrm{e}^{-K_T t}\boldsymbol{\Sigma}_i(0) + \int_0^t \mathrm{e}^{-K_T(t-\tau)} \left(\boldsymbol{w}_i(\tau)\boldsymbol{w}_i^\top(\tau) + \sigma_1 \boldsymbol{I}\right) \mathrm{d}\tau \\
&\leq \sigma_0 I + \left(\gamma_w^+ \boldsymbol{I} + \sigma_1 \boldsymbol{I}\right) \int_0^t \mathrm{e}^{-K_T(t-\tau)} \mathrm{d}\tau \\
&= \sigma_0 I + \left(\gamma_w^+ \boldsymbol{I} + \sigma_1 \boldsymbol{I}\right) \left(\frac{1 - \mathrm{e}^{-K_T t}}{K_T}\right) \\
&\leq \left(\sigma_0 + \frac{\gamma_w^+ + \sigma_1}{K_T}\right) \boldsymbol{I}. \qquad \text{(A.16)}
\end{aligned}
$$

Therefore by setting $\gamma_\Sigma^- = \min\{\sigma_0, \gamma_w^- + \sigma_1 T\}\mathrm{e}^{-K_T T}$ and $\gamma_\Sigma^+ = \left(\sigma_0 + \frac{\gamma_w^+ + \sigma_1}{K_T}\right)$ we have

$$
\gamma_\Sigma^- \boldsymbol{I} \leq \boldsymbol{\Sigma}_i \leq \gamma_\Sigma^+ \boldsymbol{I}. \qquad \text{(A.17)}
$$

Substituting these bounds into (A.13) and using the fact that $\boldsymbol{w}_i^\top \boldsymbol{w}_i < \gamma_w^+$, we can write

$$\begin{aligned}\dot{V}_{\text{PE}} \leq \sum_{i=1}^{p} \Bigg(&- \widetilde{\eta}_i^\top \left(K - \frac{1}{2} - \frac{a_{17}\gamma_w^+}{2} - \frac{a_{19}}{2} \right) \widetilde{\eta}_i + \frac{1}{2a_{17}} \dot{\boldsymbol{\theta}}_i^\top \dot{\boldsymbol{\theta}}_i + \frac{\gamma_\Sigma^+}{2a_{18}} \dot{\boldsymbol{\theta}}_i^\top \dot{\boldsymbol{\theta}}_i \\ &+ \frac{1}{2a_{19}} \dot{\nu}_i^\top \dot{\nu}_i - \frac{(K_T - a_{18})\gamma_\Sigma^-}{2} \widetilde{\boldsymbol{\theta}}_i^\top \widetilde{\boldsymbol{\theta}}_i + \frac{\sigma_1}{2} \boldsymbol{\theta}_i^\top \boldsymbol{\theta}_i \Bigg).\end{aligned} \tag{A.18}$$

Then we can finally write $\dot{V}_{\text{PE}}$ as

$$\dot{V}_{\text{PE}} \leq \sum_{i=1}^{p} \left(-c_{45} \left\| \widetilde{\eta}_i \right\|^2 - c_{46} \left\| \widetilde{\boldsymbol{\theta}}_i \right\|^2 + c_{47} \left\| \dot{\boldsymbol{\theta}}_i \right\|^2 + c_{48} \left\| \dot{\nu}_i \right\|^2 + \frac{\sigma_1}{2} \left\| \boldsymbol{\theta}_i \right\|^2 \right) \tag{A.19}$$

where

$$\begin{aligned} c_{45} &= K - \frac{1}{2} - \frac{a_{17}\gamma_w^+}{2} - \frac{a_{19}}{2} & c_{47} &= \frac{1}{2a_{17}} + \frac{\gamma_\Sigma^+}{2a_{18}} \\ c_{46} &= \frac{(K_T - a_{18})\gamma_\Sigma^-}{2} & c_{48} &= \frac{1}{2a_{19}}. \end{aligned}$$

By choosing $K > \frac{1 + a_{17}\gamma_w^+ + a_{19}}{2}$ and $K_T > a_{18}$, we ensure that the first two terms are negative definite. Since $\lim_{t\to\infty} \nu_i(t) = 0$, the $c_{48} \left\| \dot{\nu}_i \right\|^2$ term does not affect parameter estimate convergence. This Lyapunov derivative therefore shows that $\widetilde{\eta}_i$ and $\widetilde{\boldsymbol{\theta}}_i$ are uniformly ultimately bounded to a ball whose size is determined by $\boldsymbol{\theta}_i$ and $\dot{\boldsymbol{\theta}}_i$. Therefore the parameter estimates reach a neighbourhood of their true values.

In the third part of the proof, we consider the stability of the proposed ESC at a constant $\widehat{\boldsymbol{u}}$. We use the Lyapunov function

$$V_{\text{Stab}} = V_{\text{PE}} + \Omega - \Omega(\boldsymbol{x}^*, \boldsymbol{z}^*). \tag{A.20}$$

Using the system dynamics, (3.5) and (3.4), its derivative is

$$\dot{V}_{\text{Stab}} = \dot{V}_{\text{PE}} + \frac{\partial V_z}{\partial \boldsymbol{z}} \boldsymbol{f}_z + \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{f} + \sum_{i=1}^{p} \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{G}_i \boldsymbol{u}_i. \tag{A.21}$$

Using the definition of $\boldsymbol{\theta}_{1,i}$ we have that $\frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{G}_i = p\boldsymbol{\theta}_{1,i}^{\top}$ so

$$\dot{V}_{\text{Stab}} = \dot{V}_{\text{PE}} + \frac{\partial V_z}{\partial \boldsymbol{z}} \boldsymbol{f}_z + \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{f} + \sum_{i=1}^{p} p\boldsymbol{\theta}_{1,i}^{\top} \boldsymbol{u}_i. \tag{A.22}$$

Substituting the proposed ESC, (3.32), and letting $\widehat{\boldsymbol{\theta}}_{1,i} = \boldsymbol{\theta}_{1,i} - \widetilde{\boldsymbol{\theta}}_{1,i}$ gives

$$\begin{aligned} \dot{V}_{\text{Stab}} = \dot{V}_{\text{PE}} &+ \frac{\partial V_z}{\partial \boldsymbol{z}} \boldsymbol{f}_z + \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{f} \\ &+ \sum_{i=1}^{p} \left(-pK_g \boldsymbol{\theta}_{1,i}^{\top} \boldsymbol{\theta}_{1,i} + pK_g \boldsymbol{\theta}_{1,i}^{\top} \widetilde{\boldsymbol{\theta}}_{1,i} + p\boldsymbol{\theta}_{1,i}^{\top} \widehat{\boldsymbol{u}}_i + p\boldsymbol{\theta}_{1,i}^{\top} \boldsymbol{d}_i \right). \end{aligned} \tag{A.23}$$

Using the fact that $\boldsymbol{\theta}_{1,i} = \frac{1}{p} \left(\frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{G}_i \right)^{\top}$, we have

$$\begin{aligned} \dot{V}_{\text{Stab}} = \dot{V}_{\text{PE}} &+ \frac{\partial V_z}{\partial \boldsymbol{z}} \boldsymbol{f}_z + \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{f} - \frac{K_g}{p} \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{G}\boldsymbol{G}^{\top} \left(\frac{\partial J}{\partial \boldsymbol{x}} \right)^{\top} \\ &+ \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{G} \widehat{\boldsymbol{u}} + \sum_{i=1}^{p} \left(pK_g \boldsymbol{\theta}_{1,i}^{\top} \widetilde{\boldsymbol{\theta}}_{1,i} + p\boldsymbol{\theta}_{1,i}^{\top} \boldsymbol{d}_i \right). \end{aligned} \tag{A.24}$$

By Assumption Assumption 3.2, there exists $\beta_{4,x}, \beta_{4,z}, K_g^* > 0$ such that

$$\begin{aligned} \dot{V}_{\text{Stab}} = \dot{V}_{\text{PE}} &- \beta_{4,x} \left\| \boldsymbol{x} - \boldsymbol{\pi}_x(\widehat{\boldsymbol{u}}) \right\|^2 - \beta_{4,z} \left\| \boldsymbol{z} - \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}}) \right\|^2 \\ &- \left(\frac{K_g}{p} - K_g^* \right) \frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{G}\boldsymbol{G}^{\top} \left(\frac{\partial J}{\partial \boldsymbol{x}} \right)^{\top} + \sum_{i=1}^{p} \left(pK_g \boldsymbol{\theta}_{1,i}^{\top} \widetilde{\boldsymbol{\theta}}_{1,i} + p\boldsymbol{\theta}_{1,i}^{\top} \boldsymbol{d}_i \right). \end{aligned} \tag{A.25}$$

Since we can choose the parameter estimation gain, K_g, choose $K_g > pK_g^*$ we ensure that $\frac{K_g}{p} - K_g^* > 0$. Then using by substituting (A.19) and using the fact that $\frac{\partial J}{\partial \boldsymbol{x}}\boldsymbol{G}_i = p\boldsymbol{\theta}_{1,i}^\top$, we have

$$\begin{aligned}\dot{V}_{\text{Stab}} \leq \sum_{i=1}^{p} \Big(&- c_{45}\left\|\widetilde{\eta}_i\right\|^2 - c_{46}\left\|\widetilde{\theta}_{0,i}\right\|^2 + c_{47}\left\|\dot{\boldsymbol{\theta}}_i\right\|^2 \\ &+ c_{48}\left\|\dot{\nu}_i\right\|^2 + \frac{\sigma_1}{2}\left\|\boldsymbol{\theta}_i\right\|^2 + p\boldsymbol{\theta}_{1,i}^\top \boldsymbol{d}_i - \boldsymbol{v}_{\theta,i}^\top \boldsymbol{\Lambda}_\theta \boldsymbol{v}_{\theta,i}\Big) \\ &- \beta_{4,x}\left\|\boldsymbol{x} - \boldsymbol{\pi}_x(\widehat{\boldsymbol{u}})\right\|^2 - \beta_{4,z}\left\|\boldsymbol{z} - \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}})\right\|^2\end{aligned} \tag{A.26}$$

where

$$\boldsymbol{v}_{\theta,i} = \begin{bmatrix} \left\|\widetilde{\boldsymbol{\theta}}_{1,i}\right\| \\ \left\|\boldsymbol{\theta}_{1,i}\right\| \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{\Lambda}_\theta = \begin{bmatrix} c_{46} & -\frac{pK_g}{2} \\ -\frac{pK_g}{2} & p^2\left(\frac{K_g}{p} - K_g^*\right). \end{bmatrix}$$

We must check that $\boldsymbol{\Lambda}_\theta$ is positive definite. Since $c_{46} > 0$, we only need to show that $\det(\boldsymbol{\Lambda}_\theta) > 0$. The determinant is positive when $c_{46} > \frac{K_g^2}{4\left(\frac{K_g}{p} - K_g^*\right)}$ so we want to make c_{46} as large as possible. We cannot directly choose c_{46}; however, it is a function of K_T as

$$c_{46} = \frac{(K_T - a_{18})\min\{\sigma_0, \gamma_w^- + \sigma_1 T\}\mathrm{e}^{-K_T T}}{2}.$$

To maximize c_{46}, we should choose $K_T = a_{18} + \frac{1}{T}$. With this choice of K_T, we can ensure that $\boldsymbol{\Lambda}_\theta > 0$ by choosing $K_g^2 < \frac{2\gamma_\Sigma^-\left(\frac{K_g}{p} - K_g^*\right)}{T}$. Since the right hand side is a function of K_g, we can alternatively write this bound as

$$K_g < \frac{\gamma_\Sigma^-}{pT}\left(1 + \sqrt{\frac{\gamma_\Sigma^- - 2p^2T\left(\frac{K_g}{p} - K_g^*\right)}{\gamma_\Sigma^-}}\right) < \frac{2\gamma_\Sigma^-}{pT}. \tag{A.27}$$

Therefore $\boldsymbol{\Lambda}_\theta > \mathbf{0}$. Let $\gamma_\Lambda^- > 0$ be the minimum eigenvalue of $\boldsymbol{\Lambda}_\theta$. Then (A.26) can be written as

$$\begin{aligned}\dot{V}_{\text{Stab}} \leq \sum_{i=1}^{p} \Big(& - c_{45} \left\|\widetilde{\eta}_i\right\|^2 - c_{46} \left\|\widetilde{\theta}_{0,i}\right\|^2 + c_{47} \left\|\dot{\boldsymbol{\theta}}_i\right\|^2 + c_{48} \left\|\dot{\nu}_i\right\|^2 \\ & + \frac{\sigma_1}{2} \left\|\boldsymbol{\theta}_i\right\|^2 + p\boldsymbol{\theta}_{1,i}^\top \boldsymbol{d}_i - \gamma_\Lambda^- \left\|\widetilde{\boldsymbol{\theta}}_{1,i}\right\|^2 - \gamma_\Lambda^- \left\|\boldsymbol{\theta}_{1,i}\right\|^2 \\ & - \frac{\beta_{4,x}}{p} \left\|\boldsymbol{x} - \boldsymbol{\pi}_x(\widehat{\boldsymbol{u}})\right\|^2 - \frac{\beta_{4,z}}{p} \left\|\boldsymbol{z} - \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}})\right\|^2 \Big). \end{aligned} \tag{A.28}$$

The only indefinite term in (A.28) which was not in (A.19) is the indefinite term $p\boldsymbol{\theta}_{1,i}^\top \boldsymbol{d}_i$. This term is bounded since $\boldsymbol{d}_i(t)$ is a bounded dither signal. Therefore this Lyapunov function shows that the proposed ESC stabilizes the system dynamics to a neighbourhood of the steady-state manifold.

In the final part of the proof, we consider the dynamics of the input bias $\widehat{\boldsymbol{u}}$ to show that it reaches a neighbourhood of the optimum, $\boldsymbol{u}^*$, of the steady-state map, $\ell(\widehat{\boldsymbol{u}})$. To achieve this objective, we focus on proving that there exists a τ_I^* such that for all $\tau_I > \tau_I^*$, $\widehat{\boldsymbol{u}}$ reaches a neighbourhood of the optimum equilibrium input, $\boldsymbol{u}^*$. We consider the equilibrium response for the nonlinear system (3.1), given by $\boldsymbol{x} = \boldsymbol{\pi}(\widehat{\boldsymbol{u}})$, of the system at a specific $\widehat{\boldsymbol{u}}$. Let $\widetilde{\boldsymbol{u}} = \boldsymbol{u}^* - \widehat{\boldsymbol{u}}$ and consider the Lyapunov function

$$V_{\text{ESC}} = V_{\text{Stab}} + \frac{1}{2}\widetilde{\boldsymbol{u}}^\top \widetilde{\boldsymbol{u}}. \tag{A.29}$$

Differentiating and using (3.33) gives

$$\dot{V}_{\text{ESC}} = \dot{V}_{\text{Stab}} + \sum_{i=1}^{p} \frac{1}{\tau_I} \widetilde{\boldsymbol{u}}_i^\top \widehat{\boldsymbol{\theta}}_{1,i}. \tag{A.30}$$

By letting $\widehat{\boldsymbol{\theta}}_{1,i} = \boldsymbol{\theta}_{1,i} - \widetilde{\boldsymbol{\theta}}_{1,i}$ we have

$$\dot{V}_{\text{ESC}} = \dot{V}_{\text{Stab}} + \frac{1}{\tau_I}\widetilde{\boldsymbol{u}}^\top \boldsymbol{\theta}_1 - \sum_{i=1}^{p} \frac{1}{\tau_I}\widetilde{\boldsymbol{u}}_i^\top \widetilde{\boldsymbol{\theta}}_{1,i}. \tag{A.31}$$

The gradient of the steady-state map $\ell(u)$ is

$$\frac{\partial \ell(\widehat{\boldsymbol{u}})}{\partial \boldsymbol{u}} = \frac{\partial J(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}))}{\partial \boldsymbol{x}} \boldsymbol{G}(\boldsymbol{\pi}_x(\widehat{\boldsymbol{u}}), \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}})). \tag{A.32}$$

Then by letting $\widetilde{\boldsymbol{u}} = \boldsymbol{u}^* - \widehat{\boldsymbol{u}}$, Assumption Assumption 3.1 can be written as

$$0 \leq -\beta_1 \left\|\widetilde{\boldsymbol{u}}\right\|^2 - \frac{\partial J(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}))}{\partial \boldsymbol{x}} \boldsymbol{G}(\boldsymbol{\pi}_x(\widehat{\boldsymbol{u}}), \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}}))\widetilde{\boldsymbol{u}}. \tag{A.33}$$

Then since $\widetilde{\boldsymbol{u}}^\top \boldsymbol{\theta}_1 = \frac{1}{p}\frac{\partial J}{\partial \boldsymbol{x}} \boldsymbol{G}\widetilde{\boldsymbol{u}}$, we can use this inequality to write

$$\widetilde{\boldsymbol{u}}^\top \boldsymbol{\theta}_1 \leq -\frac{\beta_1}{p} \left\|\widetilde{\boldsymbol{u}}\right\|^2 - \frac{1}{p}\left(\frac{\partial J}{\partial \boldsymbol{x}}\boldsymbol{G} - \frac{\partial J(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}))}{\partial \boldsymbol{x}} \boldsymbol{G}(\boldsymbol{\pi}_x(\widehat{\boldsymbol{u}}), \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}}))\right)\widetilde{\boldsymbol{u}}. \tag{A.34}$$

Given that $J(x)$ and $\boldsymbol{G}(x)$ are smooth, it follows that there exists Lipschitz constants $L_{G,x}$ and $L_{G,z}$ such that

$$\widetilde{\boldsymbol{u}}^\top \boldsymbol{\theta}_1 \leq -\frac{\beta_1}{p} \left\|\widetilde{\boldsymbol{u}}\right\|^2 + \frac{L_{G,x}}{p}\left\|\boldsymbol{x} - \boldsymbol{\pi}_x(\widehat{\boldsymbol{u}})\right\|\left\|\widetilde{\boldsymbol{u}}\right\| + \frac{L_{G,z}}{p}\left\|\boldsymbol{z} - \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}})\right\|\left\|\widetilde{\boldsymbol{u}}\right\|. \tag{A.35}$$

Substituting this inequality into (A.31), we have

$$\begin{aligned}\dot{V}_{\text{ESC}} \leq \dot{V}_{\text{Stab}} &- \frac{\beta_1}{p\tau_I}\left\|\widetilde{\boldsymbol{u}}\right\|^2 + \frac{L_{G,x}}{p\tau_I}\left\|\boldsymbol{x} - \boldsymbol{\pi}_x(\widehat{\boldsymbol{u}})\right\|\left\|\widetilde{\boldsymbol{u}}\right\| \\ &+ \frac{L_{G,z}}{p\tau_I}\left\|\boldsymbol{z} - \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}})\right\|\left\|\widetilde{\boldsymbol{u}}\right\| - \sum_{i=1}^{p}\frac{1}{\tau_I}\widetilde{\boldsymbol{u}}_i^\top \widetilde{\boldsymbol{\theta}}_{1,i}.\end{aligned} \tag{A.36}$$

We can bound the indefinite term $\frac{1}{\tau_I}\widetilde{\boldsymbol{u}}_i^\top \widetilde{\boldsymbol{\theta}}_{1,i}$ by completing the squares:

$$\begin{aligned}\dot{V}_{\text{ESC}} \leq \dot{V}_{\text{Stab}} + \sum_{i=1}^{p} \Big(&- \frac{\beta_1}{p\tau_I} \left\|\widetilde{\boldsymbol{u}}_i\right\|^2 + \frac{L_{G,x}}{p\tau_I} \left\|\boldsymbol{x} - \boldsymbol{\pi}_x(\widehat{\boldsymbol{u}})\right\| \left\|\widetilde{\boldsymbol{u}}_i\right\| \\ &+ \frac{L_{G,z}}{p\tau_I} \left\|\boldsymbol{z} - \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}})\right\| \left\|\widetilde{\boldsymbol{u}}_i\right\| + \frac{1}{2\tau_I} \left\|\widetilde{\boldsymbol{u}}_i\right\|^2 + \frac{1}{2\tau_I} \left\|\widetilde{\boldsymbol{\theta}}_{1,i}\right\|^2 \Big).\end{aligned} \tag{A.37}$$

By substituting (A.28), we can write

$$\begin{aligned}\dot{V}_{\text{ESC}} \leq \sum_{i=1}^{p} \Big(&- c_{45} \left\|\widetilde{\eta}_i\right\|^2 - c_{46} \left\|\widetilde{\theta}_{0,i}\right\|^2 + c_{47} \left\|\dot{\boldsymbol{\theta}}_i\right\|^2 + c_{48} \left\|\dot{\nu}_i\right\|^2 \\ &+ \frac{\sigma_1}{2} \left\|\theta_{0,i}\right\|^2 + p\boldsymbol{\theta}_{1,i}^\top \boldsymbol{d}_i - \left(\gamma_\Lambda^- - \frac{1}{2\tau_I}\right) \left\|\widetilde{\boldsymbol{\theta}}_{1,i}\right\|^2 \\ &- \left(\gamma_\Lambda^- - \frac{\sigma_1}{2}\right) \left\|\boldsymbol{\theta}_{1,i}\right\|^2 - \boldsymbol{v}_x^\top \boldsymbol{\Lambda}_x \boldsymbol{v}_x - \boldsymbol{v}_z^\top \boldsymbol{\Lambda}_z \boldsymbol{v}_z \Big)\end{aligned} \tag{A.38}$$

where

$$\boldsymbol{v}_x = \begin{bmatrix} \left\|\boldsymbol{x} - \boldsymbol{\pi}_x(\widehat{\boldsymbol{u}})\right\| \\ \left\|\widetilde{\boldsymbol{u}}_i\right\| \end{bmatrix} \qquad \boldsymbol{\Lambda}_{\boldsymbol{x}} = \begin{bmatrix} \frac{\beta_{4,x}}{p} & -\frac{L_{G,x}}{2p\tau_I} \\ -\frac{L_{G,x}}{2p\tau_I} & \frac{2\beta_1 - p}{4p\tau_I} \end{bmatrix}$$

$$\boldsymbol{v}_z = \begin{bmatrix} \left\|\boldsymbol{z} - \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}})\right\| \\ \left\|\widetilde{\boldsymbol{u}}_i\right\| \end{bmatrix} \qquad \boldsymbol{\Lambda}_z = \begin{bmatrix} \frac{\beta_{4,z}}{p} & -\frac{L_{G,z}}{2p\tau_I} \\ -\frac{L_{G,z}}{2p\tau_I} & \frac{2\beta_1 - p}{4p\tau_I} \end{bmatrix}.$$

We must check that $\boldsymbol{\Lambda}_x$ and $\boldsymbol{\Lambda}_z$ are positive definite. Since $\frac{\beta_{4,x}}{p}$ and $\frac{\beta_{4,z}}{p}$ are positive, it remains to check that their determinants are positive. We can ensure that the determinants are positive by choosing τ_I such that

$$\tau_I > \max\left\{ \frac{L_{G,x}^2}{\beta_{4,x}(2\beta_1 - p)}, \frac{L_{G,z}^2}{\beta_{4,z}(2\beta_1 - p)} \right\}. \tag{A.39}$$

Additionally, to ensure that $c_{49} = \gamma_\Lambda^- - \frac{1}{2\tau_I} > 0$, we must choose $\tau_I > \frac{\gamma_\Lambda^-}{2}$. Therefore let

$$\tau_I^* = \max\left\{\frac{L_{G,x}^2}{\beta_{4,x}(2\beta_1 - p)}, \frac{L_{G,z}^2}{\beta_{4,z}(2\beta_1 - p)}, \frac{\gamma_\Lambda^-}{2}\right\} \tag{A.40}$$

and choose $\tau_I > \tau_I^*$. With this choice of τ_I, let $\gamma_v^- = \min\{\lambda_{\min}(\boldsymbol{\Lambda}_x), \lambda_{\min}(\boldsymbol{\Lambda}_z)\}$ and $c_{50} = \gamma_\Lambda^- - \frac{\sigma_1}{2}$. To ensure that $c_{50} > 0$, set $\sigma_1 < 2\gamma_\Lambda^-$. Then we can write $\dot{V}_{\text{ESC}}$ as

$$\begin{aligned}\dot{V}_{\text{ESC}} \leq \sum_{i=1}^{p}\Big(&- c_{45}\left\|\widetilde{\eta}_i\right\|^2 - c_{46}\left\|\widetilde{\theta}_{0,i}\right\|^2 + c_{47}\left\|\dot{\boldsymbol{\theta}}_i\right\|^2 + c_{48}\left\|\dot{\nu}_i\right\|^2 \\ &+ \frac{\sigma_1}{2}\left\|\theta_{0,i}\right\|^2 + p\boldsymbol{\theta}_{1,i}^\top \boldsymbol{d}_i - c_{49}\left\|\widetilde{\boldsymbol{\theta}}_{1,i}\right\|^2 - c_{50}\left\|\boldsymbol{\theta}_{1,i}\right\|^2 \\ &- \gamma_v^-\left\|\boldsymbol{x} - \boldsymbol{\pi}_x(\widehat{\boldsymbol{u}})\right\|^2 - \gamma_v^-\left\|\boldsymbol{z} - \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}})\right\| - 2\gamma_v^-\left\|\widetilde{\boldsymbol{u}}_i\right\|^2\Big).\end{aligned} \tag{A.41}$$

Thus, for every $\tau_I > \tau_I^*$, we have that $\widetilde{\eta}_i$, $\widetilde{\boldsymbol{\theta}}_i$, $\widetilde{\boldsymbol{u}}_i$ and $\boldsymbol{\theta}_{1,i}$ converge to a neighbourhood of the origin and $(\boldsymbol{x}, \boldsymbol{z})$ converges to a neighbourhood of $(\boldsymbol{\pi}_x(\widehat{\boldsymbol{u}}), \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}}))$. As $\boldsymbol{u}$ approaches a neighbourhood of $\boldsymbol{u}^*$, the state $\boldsymbol{x}$ enters a neighbourhood of the steady-state optimum $\boldsymbol{\pi}(\boldsymbol{u}^*)$. This convergence is achieved by using estimation gains K and K_T that are large enough to ensure that all constants multiplying the corresponding norms ($\left\|\widetilde{\eta}\right\|^2$, $\left\|\widetilde{\theta}\right\|^2$, $\left\|\boldsymbol{\theta}_1\right\|^2$, $\left\|\widetilde{\boldsymbol{u}}\right\|^2$, $\left\|\boldsymbol{x} - \boldsymbol{\pi}_x(\widehat{\boldsymbol{u}})\right\|^2$, and $\left\|\boldsymbol{z} - \boldsymbol{\pi}_z(\widehat{\boldsymbol{u}})\right\|^2$) are negative.

To determine the size of the neighbourhood the trajectories converge to, we must examine the positive and indefinite terms in (A.41). These terms are:

1. $c_{47}\left\|\dot{\boldsymbol{\theta}}\right\|^2$ — Since $c_{47} = \frac{1}{2a_{17}} + \frac{\gamma_\Sigma^+}{2a_{18}}$, we can reduce the effect of this term by increasing a_{17} and a_{18}. We can freely increase a_{17} by increasing K. Since a_{18} appears in $K_T - a_{18}$ which must be positive, we can only make a_{18} as large as K_T. Therefore the contribution of this term is $\mathcal{O}(\|\dot{\boldsymbol{\theta}}\|/K_T)$.

   The magnitude of $\dot{\boldsymbol{\theta}}$ depends on the magnitude of a drift term, $\frac{1}{p}L_f J$, and a controlled term, $\frac{1}{p}L_G J\boldsymbol{u}$. Since the drift is unaffected by the controller, we can simply bound its size by a positive constant, χ_0. Since the magnitude of the control is proportional to the gain, K_g, we can bound the size of the controlled term by $\chi_1 K_g$. Therefore

$$\left\|\dot{\boldsymbol{\theta}}\right\| \leq \chi_0 + \chi_1 K_g. \tag{A.42}$$

Since $\left\|\dot{\boldsymbol{\theta}}\right\|$ increases monotonically with K_g, the contribution of this term to the size of the neighbourhood is $\mathcal{O}({}^{K_g}/{}_{K_T})$.

2. $c_{48}\left\|\dot{\boldsymbol{\nu}}\right\|^2$ — Since $\lim_{t\to\infty}\boldsymbol{\nu} = \mathbf{0}$, this term does not contribute to the size of the neighbourhood.

3. $\frac{\sigma_1}{2}\left\|\boldsymbol{\theta}_0\right\|^2$ — Since $\boldsymbol{\theta}_0$ is a drift term, we cannot bound its size using any control parameters. Therefore the contribution of this term can only be expressed as $\mathcal{O}(\sigma_1\left\|\boldsymbol{\theta}_0\right\|)$.

4. $p\boldsymbol{\theta}_1^\top d$ — The dither signal is bounded such that $\left\|\boldsymbol{d}\right\| \leq D$; however the effect of the dither signal can be reduced by increasing K_g. Therefore the contribution of this term is $\mathcal{O}({}^{D}/{}_{K_g})$.

Therefore the proposed ESC converges to an $\mathcal{O}({}^{K_g}/{}_{K_T}, \sigma_1\left\|\boldsymbol{\theta}_0\right\|, {}^{D}/{}_{K_g})$-neighbourhood of the origin.

Appendix B

Proof of Lemma 4.1

Using the system dynamics,

$$\begin{aligned}\|\Delta\boldsymbol{x}[k+1]\| &= \|\boldsymbol{x}[k+1]-\boldsymbol{x}[k]\| \\ &= \|\boldsymbol{f}\left(\boldsymbol{x}[k]\right)+\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\boldsymbol{u}[k]\|\,. \end{aligned} \tag{B.1}$$

By the definition of $\boldsymbol{\pi}$, we have that $\boldsymbol{f}(\boldsymbol{\pi}(\boldsymbol{u}))+\boldsymbol{G}(\boldsymbol{\pi}(\boldsymbol{u}))\boldsymbol{u}=\boldsymbol{0}$ for any $\boldsymbol{u}\in\mathbb{U}$. Then using the triangle inequality, we have

$$\begin{aligned}\|\Delta\boldsymbol{x}[k+1]\| &= \|\boldsymbol{f}\left(\boldsymbol{x}[k]\right)+\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\boldsymbol{u}[k]-\boldsymbol{f}\left(\boldsymbol{\pi}\left(\widehat{\boldsymbol{u}}[k]\right)\right)-\boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\widehat{\boldsymbol{u}}[k]\| \\ &\leq \|\boldsymbol{f}\left(\boldsymbol{x}[k]\right)-\boldsymbol{f}\left(\boldsymbol{\pi}\left(\widehat{\boldsymbol{u}}[k]\right)\right)\| \\ &\quad+\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\boldsymbol{u}[k]-\boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\widehat{\boldsymbol{u}}[k]\|\,. \end{aligned} \tag{B.2}$$

By the Lipschitzness of $\boldsymbol{f}$, we can bound the first term to get

$$\|\Delta\boldsymbol{x}[k+1]\| \leq L_f\left\|\boldsymbol{x}[k]-\boldsymbol{\pi}\left(\widehat{\boldsymbol{u}}[k]\right)\right\|+\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\boldsymbol{u}[k]-\boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\widehat{\boldsymbol{u}}[k]\|$$

$$= L_f \left\|\widetilde{\boldsymbol{x}}[k]\right\| + \left\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\boldsymbol{u}[k] - \boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\widehat{\boldsymbol{u}}[k]\right\|. \tag{B.3}$$

We can simplify the second term by adding and subtracting $\boldsymbol{G}(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k]))\boldsymbol{u}[k]$, and then applying the triangle inequality and Cauchy-Schwarz inequality:

$$\begin{aligned}\left\|\Delta\boldsymbol{x}[k+1]\right\| &\leq L_f \left\|\widetilde{\boldsymbol{x}}[k]\right\| + \Big\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\boldsymbol{u}[k] - \boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\boldsymbol{u}[k] + \ldots\\ &\qquad + \boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\boldsymbol{u}[k] - \boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\widehat{\boldsymbol{u}}[k]\Big\|\\ &\leq L_f \left\|\widetilde{\boldsymbol{x}}[k]\right\| + \left\|\left(\boldsymbol{G}\left(\boldsymbol{x}[k]\right) - \boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\right)\boldsymbol{u}[k]\right\|\\ &\qquad + \left\|\boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\left(\boldsymbol{u}[k] - \widehat{\boldsymbol{u}}[k]\right)\right\|\\ &\leq L_f \left\|\widetilde{\boldsymbol{x}}[k]\right\| + \left\|\boldsymbol{u}[k]\right\| \left\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right) - \boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\right\|\\ &\qquad + \left\|\boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\right\| \left\|\boldsymbol{u}[k] - \widehat{\boldsymbol{u}}[k]\right\|.\end{aligned} \tag{B.4}$$

The second term can be bounded using the Lipschitzness of $\boldsymbol{G}$ and the bound on $\|\boldsymbol{u}\|$ from Assumption 4.5. We also simplify the last term by adding and subtracting $\boldsymbol{G}(\boldsymbol{\pi}(\boldsymbol{u}^*))$ and applying the triangle inequality:

$$\begin{aligned}\left\|\Delta\boldsymbol{x}[k+1]\right\| &\leq L_f \left\|\widetilde{\boldsymbol{x}}[k]\right\| + \sqrt{p\gamma_u^+ T} L_G \left\|\boldsymbol{x}[k] - \boldsymbol{\pi}\left(\widehat{\boldsymbol{u}}[k]\right)\right\|\\ &\qquad + \left\|\boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right) - \boldsymbol{G}\left(\boldsymbol{\pi}(\boldsymbol{u}^*)\right) + \boldsymbol{G}\left(\boldsymbol{\pi}(\boldsymbol{u}^*)\right)\right\| \left\|\boldsymbol{u}[k] - \widehat{\boldsymbol{u}}[k]\right\|\\ &\leq \left(L_f + \sqrt{p\gamma_u^+ T} L_G\right) \left\|\widetilde{\boldsymbol{x}}[k]\right\|\\ &\qquad + \big(\left\|\boldsymbol{G}\left(\boldsymbol{\pi}\left(\widehat{\boldsymbol{u}}[k]\right)\right) - \boldsymbol{G}\left(\boldsymbol{\pi}(\boldsymbol{u}^*)\right)\right\| + \left\|\boldsymbol{G}\left(\boldsymbol{\pi}(\boldsymbol{u}^*)\right)\right\|\big) \times \ldots\\ &\qquad\ \left\|\boldsymbol{u}[k] - \widehat{\boldsymbol{u}}[k]\right\|\\ &\leq c_1 \left\|\widetilde{\boldsymbol{x}}[k]\right\| + \big(\left\|\boldsymbol{G}\left(\boldsymbol{\pi}\left(\widehat{\boldsymbol{u}}[k]\right)\right) - \boldsymbol{G}\left(\boldsymbol{\pi}(\boldsymbol{u}^*)\right)\right\| + \ldots\\ &\qquad + \left\|\boldsymbol{G}\left(\boldsymbol{\pi}(\boldsymbol{u}^*)\right)\right\|\big) \left\|\boldsymbol{u}[k] - \widehat{\boldsymbol{u}}[k]\right\|\end{aligned} \tag{B.5}$$

where $c_1 = L_f + \sqrt{p\gamma_u^+ T} L_G$. Since $\boldsymbol{u}^*$ is constant, $\boldsymbol{G}(\boldsymbol{\pi}(\boldsymbol{u}^*))$ is a constant and we denote it by γ_G. Then using the Lipschitzness of $\boldsymbol{G}$ and $\boldsymbol{\pi}$, we can write the bound as

$$\begin{aligned}\|\Delta \boldsymbol{x}[k+1]\| &\le c_1 \|\widetilde{\boldsymbol{x}}[k]\| + \big(L_G \|\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k]) - \boldsymbol{\pi}(\boldsymbol{u}^*)\| + \gamma_G\big) \|\boldsymbol{u}[k] - \widehat{\boldsymbol{u}}[k]\| \\ &\le c_1 \|\widetilde{\boldsymbol{x}}[k]\| + \big(L_G L_\pi \|\widehat{\boldsymbol{u}}[k] - \boldsymbol{u}^*\| + \gamma_G\big) \|\boldsymbol{u}[k] - \widehat{\boldsymbol{u}}[k]\| \\ &\le c_1 \|\widetilde{\boldsymbol{x}}[k]\| + \big(L_G L_\pi \|\widetilde{\boldsymbol{u}}[k]\| + \gamma_G\big) \|\boldsymbol{u}[k] - \widehat{\boldsymbol{u}}[k]\| . \end{aligned} \tag{B.6}$$

By looking at the controller dynamics (4.24)–(4.25), we have that $\boldsymbol{u}[k] - \widehat{\boldsymbol{u}}[k] = -K_g \widehat{\boldsymbol{\theta}}_1[k] + \boldsymbol{d}[k]$, which we can use with the triangle inequality to simplify the bound as

$$\begin{aligned}\|\Delta \boldsymbol{x}[k+1]\| &\le c_1 \|\widetilde{\boldsymbol{x}}[k]\| + \big(L_G L_\pi \|\widetilde{\boldsymbol{u}}[k]\| + \gamma_G\big) \left\| K_g \widehat{\boldsymbol{\theta}}_1[k] + \boldsymbol{d}[k] \right\| \\ &\le c_1 \|\widetilde{\boldsymbol{x}}[k]\| + \big(L_G L_\pi \|\widetilde{\boldsymbol{u}}[k]\| + \gamma_G\big) \left(K_g \left\| \widehat{\boldsymbol{\theta}}_1[k] \right\| + \|\boldsymbol{d}[k]\| \right) . \end{aligned} \tag{B.7}$$

By the projection algorithm used in (4.22), $\|\widehat{\boldsymbol{\theta}}_{1,i}[k]\| < \gamma_\theta$ and by the choice of dither signals $\|\boldsymbol{d}_i[k]\| \le D$. Since there are p agents, $\|\widehat{\boldsymbol{\theta}}_1[k]\| < \gamma_\theta \sqrt{p}$ and $\|\boldsymbol{d}_i[k]\| \le D\sqrt{p}$. Therefore

$$\begin{aligned}\|\Delta \boldsymbol{x}[k+1]\| &\le c_1 \|\widetilde{\boldsymbol{x}}[k]\| + \big(L_G L_\pi \|\widetilde{\boldsymbol{u}}[k]\| + \gamma_G\big) (K_g \gamma_\theta \sqrt{p} + D\sqrt{p}) \\ &= c_1 \|\widetilde{\boldsymbol{x}}[k]\| + L_G L_\pi \sqrt{p} \left(K_g \gamma_\theta + D\right) \|\widetilde{\boldsymbol{u}}[k]\| \\ &\quad + \gamma_G \gamma_\theta \sqrt{p} K_g + \gamma_G \sqrt{p} D \\ &= c_1 \|\widetilde{\boldsymbol{x}}[k]\| + c_2 \|\widetilde{\boldsymbol{u}}[k]\| + c_3 K_g + c_4 D \end{aligned} \tag{B.8}$$

where $c_2 = L_G L_\pi (K_g \gamma_\theta \sqrt{p} + D\sqrt{p})$, $c_3 = \gamma_G \gamma_\theta \sqrt{p}$, and $c_4 = \gamma_G \sqrt{p}$. While c_1, c_2, and c_3 are constants, they can be changed by suitable choices of K_g and D.

Appendix C

Proof of Lemma 4.2

Using the definition of $\boldsymbol{\theta}_i$ from (4.5)–(4.6) and the triangle inequality, we have

$$
\begin{aligned}
\|\Delta\boldsymbol{\theta}_i[k+1]\| = {} & \|\boldsymbol{\theta}_i[k+1] - \boldsymbol{\theta}_i[k]\| \\
\leq {} & \|\boldsymbol{\theta}_{1,i}[k+1] - \boldsymbol{\theta}_{1,i}[k]\| + \|\boldsymbol{\theta}_{0,i}[k+1] - \boldsymbol{\theta}_{0,i}[k]\| \\
\leq {} & \Big\| H\big(\boldsymbol{x}[k+1] + \boldsymbol{f}\left(\boldsymbol{x}[k+1]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k+1]\right)\widehat{\boldsymbol{u}}[k+1]\big) - \dots \\
& \quad - H\big(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\widehat{\boldsymbol{u}}[k]\big)\Big\| \\
& + \left\| \sum_{j\neq i} \nabla H\left(\overline{\boldsymbol{x}}\right)\boldsymbol{G}_j\left(\boldsymbol{x}[k+1]\right)\left(\boldsymbol{u}_j[k+1] - \widehat{\boldsymbol{u}}_j[k+1]\right) - \dots \right. \\
& \qquad \left. - \sum_{j\neq i} \nabla H\left(\overline{\boldsymbol{x}}\right)\boldsymbol{G}_j\left(\boldsymbol{x}[k]\right)\left(\boldsymbol{u}_j[k] - \widehat{\boldsymbol{u}}_j[k]\right) \right\| \\
& + \left\|-H\left(\boldsymbol{x}[k+1]\right) + H\left(\boldsymbol{x}[k]\right)\right\| \\
& + \left\|\boldsymbol{G}_i^\top\left(\boldsymbol{x}[k+1]\right)\nabla^\top H\left(\overline{\boldsymbol{x}}\right) - \boldsymbol{G}_i^\top\left(\boldsymbol{x}[k]\right)\nabla^\top H\left(\overline{\boldsymbol{x}}\right)\right\|.
\end{aligned}
\tag{C.1}
$$

Since H is Lipschitz, its gradient is also bounded so $\nabla H(\boldsymbol{x}) \leq L_H$ for any $\boldsymbol{x}$. In particular, $\nabla H(\overline{\boldsymbol{x}}) \leq L_H$. Using this fact and the usual Lipschitz inequality, we obtain

$$\begin{aligned}
\|\Delta\boldsymbol{\theta}_i[k+1]\| \leq\, & L_H \Big\| \boldsymbol{x}[k+1] + \boldsymbol{f}\left(\boldsymbol{x}[k+1]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k+1]\right)\widehat{\boldsymbol{u}}[k+1] - \dots \\
& \qquad - \boldsymbol{x}[k] - \boldsymbol{f}\left(\boldsymbol{x}[k]\right) - \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\widehat{\boldsymbol{u}}[k] \Big\| \\
& + L_H \Bigg\| \sum_{j\neq i} \boldsymbol{G}_j\left(\boldsymbol{x}[k+1]\right)\left(\boldsymbol{u}_j[k+1] - \widehat{\boldsymbol{u}}_j[k+1]\right) - \dots \\
& \qquad - \boldsymbol{G}_j\left(\boldsymbol{x}[k]\right)\left(\boldsymbol{u}_j[k] - \widehat{\boldsymbol{u}}_j[k]\right) \Bigg\| \\
& + L_H \left\|\boldsymbol{x}[k+1] - \boldsymbol{x}[k]\right\| + L_H \left\|\boldsymbol{G}_i\left(\boldsymbol{x}[k+1]\right) - \boldsymbol{G}_i\left(\boldsymbol{x}[k]\right)\right\| \\
\leq\, & L_H \left\|\boldsymbol{x}[k+1] - \boldsymbol{x}[k]\right\| + L_H \left\|\boldsymbol{f}\left(\boldsymbol{x}[k+1]\right) - \boldsymbol{f}\left(\boldsymbol{x}[k]\right)\right\| \\
& + L_H \left\|\boldsymbol{G}\left(\boldsymbol{x}[k+1]\right)\widehat{\boldsymbol{u}}[k+1] - \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\widehat{\boldsymbol{u}}[k]\right\| \\
& + L_H \Big\| \boldsymbol{G}\left(\boldsymbol{x}[k+1]\right)\left(\boldsymbol{u}[k+1] - \widehat{\boldsymbol{u}}[k+1]\right) - \dots \\
& \qquad - \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\left(\boldsymbol{u}[k] - \widehat{\boldsymbol{u}}[k]\right) \Big\| \\
& + L_H \left\|\boldsymbol{x}[k+1] - \boldsymbol{x}[k]\right\| + L_H \left\|\boldsymbol{G}_i\left(\boldsymbol{x}[k+1]\right) - \boldsymbol{G}_i\left(\boldsymbol{x}[k]\right)\right\|.
\end{aligned} \tag{C.2}$$

Using the Lipschitzness of $\boldsymbol{f}$ and $\boldsymbol{G}$, we can simplify the second and last terms:

$$\begin{aligned}
\|\Delta\boldsymbol{\theta}_i[k+1]\| \leq\, & L_H \left\|\boldsymbol{x}[k+1] - \boldsymbol{x}[k]\right\| + L_H L_f \left\|\boldsymbol{x}[k+1] - \boldsymbol{x}[k]\right\| \\
& + L_H \left\|\boldsymbol{G}\left(\boldsymbol{x}[k+1]\right)\widehat{\boldsymbol{u}}[k+1] - \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\widehat{\boldsymbol{u}}[k]\right\| \\
& + L_H \Big\| \boldsymbol{G}\left(\boldsymbol{x}[k+1]\right)\left(\boldsymbol{u}[k+1] - \widehat{\boldsymbol{u}}[k+1]\right) - \dots \\
& \qquad - \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\left(\boldsymbol{u}[k] - \widehat{\boldsymbol{u}}[k]\right) \Big\| \\
& + L_H \left\|\boldsymbol{x}[k+1] - \boldsymbol{x}[k]\right\| + L_H L_f \left\|\boldsymbol{x}[k+1] - \boldsymbol{x}[k]\right\|
\end{aligned}$$

$$\begin{aligned}
&\leq L_H\left(2+L_G+L_f\right)\|\Delta \boldsymbol{x}[k+1]\| \\
&\quad + L_H\left\|\boldsymbol{G}\left(\boldsymbol{x}[k+1]\right)\widehat{\boldsymbol{u}}[k+1]-\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\widehat{\boldsymbol{u}}[k]\right\| \\
&\quad + L_H\Big\|\boldsymbol{G}\left(\boldsymbol{x}[k+1]\right)\left(\boldsymbol{u}[k+1]-\widehat{\boldsymbol{u}}[k+1]\right)-\ldots \\
&\qquad\quad -\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\left(\boldsymbol{u}[k]-\widehat{\boldsymbol{u}}[k]\right)\Big\|.
\end{aligned} \tag{C.3}$$

The remaining terms are of the form $\|r_2 s_2 - r_1 s_1\|$, but we would like to bound them by terms of the form $\|r_2 - r_1\|$ or $\|s_2 - s_1\|$. By adding and subtracting $r_1 s_2$, and using the triangle inequality, we can bound these types of terms by $\|r_2 s_2 - r_1 s_1\| = \|r_2 s_2 - r_1 s_2 + r_1 s_2 - r_1 s_1\| \leq \|r_2 s_2 - r_1 s_2\| + \|r_1 s_2 - r_1 s_1\| = \|s_2\|\|r_2 - r_1\| + \|r_1\|\|s_2 - s_1\|$. Using this type of bound for the remaining terms, we have

$$\begin{aligned}
\|\Delta\boldsymbol{\theta}_i[k+1]\| \leq\ & L_H\left(2+L_G+L_f\right)\|\Delta \boldsymbol{x}[k+1]\| \\
&+ L_H\left\|\widehat{\boldsymbol{u}}[k+1]\right\|\left\|\boldsymbol{G}\left(\boldsymbol{x}[k+1]\right)-\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\| \\
&+ L_H\left\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\|\left\|\widehat{\boldsymbol{u}}[k+1]-\widehat{\boldsymbol{u}}[k]\right\| \\
&+ L_H\left\|\boldsymbol{u}[k+1]-\widehat{\boldsymbol{u}}[k+1]\right\|\left\|\boldsymbol{G}\left(\boldsymbol{x}[k+1]\right)-\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\| \\
&+ L_H\left\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\|\left\|\left(\boldsymbol{u}[k+1]-\widehat{\boldsymbol{u}}[k+1]\right)-\left(\boldsymbol{u}[k]-\widehat{\boldsymbol{u}}[k]\right)\right\|.
\end{aligned} \tag{C.4}$$

Terms involving $\boldsymbol{u}$ or $\widehat{\boldsymbol{u}}$ can be simplified by substituting the control laws (4.24)–(4.25) and the triangle inequality:

$$\begin{aligned}
\|\Delta\boldsymbol{\theta}_i[k+1]\| \leq\ & L_H\left(2+L_G+L_f\right)\|\Delta \boldsymbol{x}[k+1]\| \\
&+ L_H\left\|\boldsymbol{u}[k+1]+K_g\widehat{\boldsymbol{\theta}}_1[k+1]-\boldsymbol{d}[k+1]\right\| \times \ldots \\
&\quad \left\|\boldsymbol{G}\left(\boldsymbol{x}[k+1]\right)-\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\| + L_H\left\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\|\left\|\frac{1}{\tau_I}\widehat{\boldsymbol{\theta}}_1[k]\right\| \\
&+ L_H\left\|-K_g\widehat{\boldsymbol{\theta}}_1[k+1]+\boldsymbol{d}[k+1]\right\|\left\|\boldsymbol{G}\left(\boldsymbol{x}[k+1]\right)-\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\|
\end{aligned}$$

$$
\begin{aligned}
&\quad + L_H \left\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\| \times \dots \\
&\qquad \left\| -K_g(\widehat{\boldsymbol{\theta}}_1[k+1] - \widehat{\boldsymbol{\theta}}_1[k]) + (\boldsymbol{d}[k+1] - \boldsymbol{d}[k]) \right\| \\
&\leq L_H \left(2 + L_G + L_f\right) \left\|\Delta \boldsymbol{x}[k+1]\right\| \\
&\quad + L_H \left(\left\|\boldsymbol{u}[k+1]\right\| + K_g \left\|\widehat{\boldsymbol{\theta}}_1[k+1]\right\| + \left\|\boldsymbol{d}[k+1]\right\| \right) \times \dots \\
&\quad \times \left\|\boldsymbol{G}\left(\boldsymbol{x}[k+1]\right) - \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\| + \frac{L_H}{\tau_I} \left\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\| \left\|\widehat{\boldsymbol{\theta}}_1[k]\right\| \\
&\quad + L_H \left(K_g \left\|\widehat{\boldsymbol{\theta}}_1[k+1]\right\| + \left\|\boldsymbol{d}[k+1]\right\| \right) \times \dots \qquad \text{(C.5)} \\
&\qquad \left\|\boldsymbol{G}\left(\boldsymbol{x}[k+1]\right) - \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\| \\
&\quad + L_H \left\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\| \left(K_g \left\|\widehat{\boldsymbol{\theta}}_1[k+1] - \widehat{\boldsymbol{\theta}}_1[k]\right\| + \dots \right. \\
&\qquad \left. \left\|\boldsymbol{d}[k+1] - \boldsymbol{d}[k]\right\| \right).
\end{aligned}
$$

By the projection algorithm in (4.22), $\|\widehat{\boldsymbol{\theta}}_1[k]\| \leq \gamma_\theta$. Using Assumption 4.5, $\boldsymbol{u}[k] \leq \sqrt{p\gamma_u^+ T}$. Since the dither signals are sinusoids with amplitude D, we have that $\|\boldsymbol{d}[k]\| < D$. Using these bounds,

$$
\begin{aligned}
\left\|\Delta \boldsymbol{\theta}_i[k+1]\right\| &\leq L_H \left(2 + L_G + L_f\right) \left\|\Delta \boldsymbol{x}[k+1]\right\| \\
&\quad + L_H \left(\sqrt{p\gamma_u^+ T} + K_g \gamma_\theta + D \right) \left\|\boldsymbol{G}\left(\boldsymbol{x}[k+1]\right) - \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\| \\
&\quad + \frac{L_H}{\tau_I} \left\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\| \gamma_\theta \\
&\quad + L_H \left(K_g \gamma_\theta + D\right) \left\|\boldsymbol{G}\left(\boldsymbol{x}[k+1]\right) - \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\| \\
&\quad + L_H \left\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\| \left(2K_g \gamma_\theta + 2D\right) \\
&= L_H \left(2 + L_G + L_f\right) \left\|\Delta \boldsymbol{x}[k+1]\right\| \\
&\quad + L_H \left(\sqrt{p\gamma_u^+ T} + 2K_g \gamma_\theta + 2D \right) \left\|\boldsymbol{G}\left(\boldsymbol{x}[k+1]\right) - \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\| \qquad \text{(C.6)} \\
&\quad + L_H \left(\frac{\gamma_\theta}{\tau_I} + 2K_g \gamma_\theta + 2D \right) \left\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\right\|.
\end{aligned}
$$

We can simplify the second term using the Lipschitzness of G. For the third term, we add and subtract $\boldsymbol{G}(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k]))$ and $\boldsymbol{G}(\boldsymbol{\pi}(\boldsymbol{u}^*))$ and then use the triangle inequality and Lipschitzness of G and $\boldsymbol{\pi}$ to obtain

$$\begin{aligned}
\|\Delta\boldsymbol{\theta}_i[k+1]\| \leq{} & L_H\left(2+L_G+L_f\right)\|\Delta\boldsymbol{x}[k+1]\| \\
&+ L_H\left(\sqrt{p\gamma_u^+T}+2K_g\gamma_\theta+2D\right)\left(L_G\left\|\boldsymbol{x}[k+1]-\boldsymbol{x}[k]\right\|\right) \\
&+ L_H\left(\frac{\gamma_\theta}{\tau_I}+2K_g\gamma_\theta+2D\right)\big\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)-\boldsymbol{G}(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k]))+\ldots \\
&\quad \boldsymbol{G}(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k]))-\boldsymbol{G}(\boldsymbol{\pi}(\boldsymbol{u}^*))+\boldsymbol{G}(\boldsymbol{\pi}(\boldsymbol{u}^*))\big\| \\
\leq{} & L_H\left(2+L_G\left(1+\sqrt{p\gamma_u^+T}+2K_g\gamma_\theta+2D\right)+L_f\right)\|\Delta\boldsymbol{x}[k+1]\| \\
&+ L_H\left(\frac{\gamma_\theta}{\tau_I}+2K_g\gamma_\theta+2D\right)\big(\left\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right)-\boldsymbol{G}(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k]))\right\|+\ldots \\
&\quad \|\boldsymbol{G}(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k]))-\boldsymbol{G}(\boldsymbol{\pi}(\boldsymbol{u}^*))\|+\|\boldsymbol{G}(\boldsymbol{\pi}(\boldsymbol{u}^*))\|\big) \\
\leq{} & c_5\|\Delta\boldsymbol{x}[k+1]\|+L_H\left(\frac{\gamma_\theta}{\tau_I}+2K_g\gamma_\theta+2D\right)\times\ldots \\
&\big(L_G\|\widetilde{\boldsymbol{x}}[k]\|+L_GL_\pi\|uT[k]\|+\gamma_G\big)
\end{aligned} \tag{C.7}$$

where $c_5 = L_H(2+L_G(1+\sqrt{p\gamma_u^+T}+2K_g\gamma_\theta+2D)+L_f)$. Finally, we can insert the bound for $\|\Delta\boldsymbol{x}[k+1]\|$ from Lemma 4.1 to obtain

$$\begin{aligned}
\|\Delta\boldsymbol{\theta}_i[k+1]\| \leq{} & c_5\left(c_1\|\widetilde{\boldsymbol{x}}[k]\|+c_2\|\widetilde{\boldsymbol{u}}[k]\|+c_3K_g+c_4D\right) \\
&+ L_H\left(L_G\|\widetilde{\boldsymbol{x}}[k]\|+L_GL_\pi\|\widetilde{\boldsymbol{u}}[k]\|+\gamma_G\right)\left(\frac{\gamma_\theta}{\tau_I}+2K_g\gamma_\theta+2D\right) \\
={} & \left(c_1c_5+L_G\left(\frac{\gamma_\theta}{\tau_I}+2L_HK_g\gamma_\theta+2D\right)\right)\|\widetilde{\boldsymbol{x}}[k]\| \\
&+ \left(c_2c_5+L_GL_\pi\left(\frac{\gamma_\theta}{\tau_I}+2L_HK_g\gamma_\theta+2D\right)\right)\|\widetilde{\boldsymbol{u}}[k]\| \\
&+ \left(c_3c_5+2L_HK_g\gamma_G\gamma_\theta\right)K_g+L_H\gamma_G\gamma_\theta\frac{1}{\tau_I}
\end{aligned}$$

$$\begin{aligned} &+ \left(c_4 c_5 + 2 L_H \gamma_G\right) D \\ &= c_6 \left\|\widetilde{\boldsymbol{x}}[k]\right\| + c_7 \left\|\widetilde{\boldsymbol{u}}[k]\right\| + c_8 K_g + c_9 \frac{1}{\tau_I} + c_{10} D \end{aligned} \tag{C.8}$$

where $c_6 = c_1 c_5 + L_H L_G(\frac{\gamma_\theta}{\tau_I} + 2K_g\gamma_\theta + 2D)$, $c_7 = c_2 c_5 + L_H L_G L_\pi(\frac{\gamma_\theta}{\tau_I} + 2K_g\gamma_\theta + 2D)$, $c_8 = c_3 c_5 + 2L_H K_g \gamma_G \gamma_\theta)$, $c_9 = L_H \gamma_G \gamma_\theta$, and $c_{10} = c_4 c_5 + 2L_H \gamma_G$ are positive constants.

Appendix D

Proof of Lemma 4.3

We are using the consensus algorithm from [66]. In corollary 4.2 from their paper, they prove that this algorithm is ultimately bounded when its average input has bounded first differences. The inputs to our consensus algorithm are the local costs, y_i, and so the average input to the consensus algorithm is simply the average cost $\frac{1}{p}J$. Therefore to apply the results of [66], we must show that $\|\Delta J\|$ is bounded.

Using the Lipschitzness of H and the bound for $\|\Delta \boldsymbol{x}[k+1]\|$ from Lemma 4.1, we can bound $\|\Delta J\|$ by

$$\begin{aligned} \|\Delta J[k+1]\| &= \|J[k+1] - J[k]\| \\ &= \|H(\boldsymbol{x}[k+1]) - H(x[k])\| \\ &\leq L_H \|\boldsymbol{x}[k+1] - \boldsymbol{x}[k]\| \\ &\leq L_H c_1 \|\widetilde{\boldsymbol{x}}[k]\| + L_H c_2 \|\widetilde{\boldsymbol{u}}[k]\| + L_H c_3 K_g + L_H c_4 D. \end{aligned} \tag{D.1}$$

Therefore the average input to the consensus algorithm is bounded, and so we can use the ultimate bound from [66]. This bound is

$$\lim_{k\to\infty}\left\|\widetilde{J}_i[k]\right\| \leq \frac{\max_{k\in\mathbb{Z}_{>0}}\left\{\left\|\frac{1}{p}\Delta J[k]\right\|\right\}}{\Delta t\kappa_I\lambda_2} \tag{D.2}$$

where λ_2 is the second smallest eigenvalue of $\boldsymbol{L}$. By Assumption 4.4, $\lambda_2 \neq 0$ so this bound is well-defined. We use the tuning parameters of $\kappa_P = \frac{1}{\Delta t}$ and $\kappa_I = \frac{1}{\Delta t \deg^+(\boldsymbol{L})}$ where $\deg^+(\boldsymbol{L})$ is the maximum degree of the network. Since the consensus error is ultimately bounded, for any $\epsilon > 0$, there exists $N(\epsilon) \in \mathbb{Z}_{>0}$ such that for any $k > N$, we have that

$$\left\|\widetilde{J}_i[k]\right\| \leq \frac{\deg^+(\boldsymbol{L})\max_{k\in\mathbb{Z}_{>0}}\{\|\Delta J[k]\|\}}{\lambda_2} + \epsilon. \tag{D.3}$$

In particular by letting $\epsilon = \frac{1}{2}\frac{\deg^+(\boldsymbol{L})\max_{k\in\mathbb{Z}_{>0}}\{\|\Delta J[k]\|\}}{\lambda_2}$, there exists N such that for all $k > N$, we have

$$\left\|\widetilde{J}_i[k]\right\| \leq \frac{3}{2}\frac{\deg^+(\boldsymbol{L})\max_{k\in\mathbb{Z}_{>0}}\{\|\Delta J[k]\|\}}{\lambda_2}. \tag{D.4}$$

We already have a bound for $\|\Delta J[k]\|$ and since this bound holds for all k, it holds for the argmax. Therefore the consensus error is bounded by

$$\begin{aligned}
\left\|\Delta\widetilde{J}_i[k+1]\right\| &= \left\|\widetilde{J}[k+1] - \widetilde{J}[k]\right\| \\
&\leq \left\|\widetilde{J}_i[k+1]\right\| + \left\|\widetilde{J}[k]\right\| \\
&\leq \frac{3}{2}\frac{\deg^+(\boldsymbol{L})L_H\left(c_1\left\|\widetilde{\boldsymbol{x}}[k]\right\| + c_2\left\|\widetilde{\boldsymbol{u}}[k]\right\| + c_3K_g + c_4D\right)}{\widehat{\lambda}_2} \\
&\quad + \frac{3}{2}\frac{\deg^+(\boldsymbol{L})L_H\left(c_1\left\|\widetilde{\boldsymbol{x}}[k]\right\| + c_2\left\|\widetilde{\boldsymbol{u}}[k]\right\| + c_3K_g + c_4D\right)}{\widehat{\lambda}_2}
\end{aligned}$$

$$\begin{aligned}
&= \frac{3\deg^{+}(\boldsymbol{L})L_H c_1}{\widehat{\lambda}_2}\left\|\widetilde{\boldsymbol{x}}[k]\right\| + \frac{3\deg^{+}(\boldsymbol{L})L_H c_2}{\widehat{\lambda}_2}\left\|\widetilde{\boldsymbol{u}}[k]\right\| \\
&\quad + \frac{3\deg^{+}(\boldsymbol{L})L_H c_3}{\widehat{\lambda}_2}K_g + \frac{3\deg^{+}(\boldsymbol{L})L_H c_4}{\widehat{\lambda}_2}D \\
&= c_{11}\left\|\widetilde{\boldsymbol{x}}[k]\right\| + c_{12}\left\|\widetilde{\boldsymbol{u}}[k]\right\| + c_{13}K_g + c_{14}D \qquad \text{(D.5)}
\end{aligned}$$

where $c_{11} = \frac{3\deg^{+}(\boldsymbol{L})L_H c_1}{\widehat{\lambda}_2}$, $c_{12} = \frac{3\deg^{+}(\boldsymbol{L})L_H c_2}{\widehat{\lambda}_2}$, $c_{13} = \frac{3\deg^{+}(\boldsymbol{L})L_H c_3}{\widehat{\lambda}_2}$ and $c_{14} = \frac{3\deg^{+}(\boldsymbol{L})L_H c_4}{\widehat{\lambda}_2}$ are positive constants.

Appendix E

Proof of Lemma 4.4

By iteratively applying (4.21), we have

$$\begin{aligned}\boldsymbol{\Sigma}_i[k+1] &= \alpha\boldsymbol{\Sigma}_i[k] + \boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k] \\ &= \alpha\left(\alpha\boldsymbol{\Sigma}_i[k-1] + \boldsymbol{w}_i[k-1]\boldsymbol{w}_i^\top[k-1]\right) + \boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k] \\ &= \alpha^2\boldsymbol{\Sigma}_i[k-1] + \alpha\boldsymbol{w}_i[k-1]\boldsymbol{w}_i^\top[k-1] + \boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k] \\ &= \alpha^{r+1}\boldsymbol{\Sigma}_i[k-r] + \sum_{j=k-r}^{k}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j] \\ &= \alpha^{k+1}\boldsymbol{\Sigma}_i[0] + \sum_{j=0}^{k}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]. \qquad \text{(E.1)}\end{aligned}$$

In this way, we can write $\boldsymbol{\Sigma}_i[k+1]$ in terms of $\boldsymbol{w}_i[j]$ for $j \in \{1, \ldots, k\}$ and $\boldsymbol{\Sigma}_i[0]$. Since we can arbitrarily assign $\boldsymbol{\Sigma}_i[0]$, we set $\boldsymbol{\Sigma}_i[0] = \sigma_0\boldsymbol{I}$ for some $\sigma_0 \in \mathbb{R}_{>0}$ and then we can explicitly write

$$\boldsymbol{\Sigma}_i[k+1] = \alpha^{k+1}\sigma_0\boldsymbol{I} + \sum_{j=0}^{k}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]. \qquad \text{(E.2)}$$

By Assumption 4.5, we have several bounds for sums of $\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]$. By adding or subtracting appropriate amounts of positive semidefinite terms of the form $\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]$, we can bound $\boldsymbol{\Sigma}_i[k+1]$ by an expression involving sums over T time steps of $\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]$. Then these bounds can be simplified using Assumption 4.5 to obtain a bound which is a geometric sequence in α which can easily be simplified.

First, we find a lower bound for $\boldsymbol{\Sigma}_i[k+1]$. We consider the cases where $k \leq T$ and $k > T$ separately. First, assume $k \leq T$. Then

$$\begin{aligned}\boldsymbol{\Sigma}_i[k+1] &= \alpha^{k+1}\sigma_0\boldsymbol{I} + \sum_{j=0}^{k}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]\\ &\geq \alpha^{k+1}\sigma_0\boldsymbol{I}\\ &\geq \alpha^{T+1}\sigma_0\boldsymbol{I}. \end{aligned} \tag{E.3}$$

Next consider the case where $k > T$. Since the first term is positive, we can lower bound it by zero:

$$\begin{aligned}\boldsymbol{\Sigma}_i[k+1] &= \alpha^{k+1}\sigma_0\boldsymbol{I} + \sum_{j=0}^{k}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]\\ &\geq \frac{1}{T}\sum_{j=0}^{k}T\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]. \end{aligned} \tag{E.4}$$

We would like to bound the sum using Assumption 4.5 and therefore need to write this sum as several sums which resemble (4.28). These sums need to be over a horizon of T time steps. Since $k > T$, there is at least one such sum. The exact number of these sums is $k-T+1$. Additionally, we have several sums over shorter time horizons for the first $T-2$ terms and last $T-2$ terms. When we rearrange the sum in this way, we obtain

$$\begin{aligned}\boldsymbol{\Sigma}_i[k+1] &\geq \frac{1}{T}\left(\sum_{j=0}^{T-1}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j] + \cdots + \sum_{j=k-T+1}^{k}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]\right)\\ &\quad + \frac{1}{T}\left(\sum_{j=0}^{0}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j] + \cdots + \sum_{j=0}^{T-2}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]\right)\\ &\quad + \frac{1}{T}\left(\sum_{j=k-T+2}^{k}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j] + \cdots + \sum_{j=k}^{k}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]\right)\\ &= \frac{1}{T}\sum_{s=0}^{k-T+1}\sum_{j=s}^{s+T-1}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j] + \frac{1}{T}\sum_{s=0}^{T-2}\sum_{j=0}^{s}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]\\ &\quad + \frac{1}{T}\sum_{s=0}^{T-2}\sum_{j=k-s}^{k}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]\\ &\geq \frac{1}{T}\sum_{s=0}^{k-T+1}\sum_{j=s}^{s+T-1}\alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]. \end{aligned} \tag{E.5}$$

Since $\alpha \in (0,1)$, for any $j \in \{s, \ldots, s+T-1\}$ we have that $\alpha^{k-s} \geq \alpha^{k-j}$. We can factor the α terms from the second sum using this bound.

$$\boldsymbol{\Sigma}_i[k+1] \geq \sum_{s=0}^{k-T+1}\alpha^{k-s}\left(\frac{1}{T}\sum_{j=s}^{s+T-1}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]\right). \tag{E.6}$$

The second sum is over a time horizon of T time steps as in (4.28) and we can apply Assumption 4.5 to bound these sums by

$$\begin{aligned}\boldsymbol{\Sigma}_i[k+1] &> \sum_{s=0}^{k-T+1}\alpha^{k-s}\gamma_w^-\boldsymbol{I}\\ &= \alpha^k\sum_{s=0}^{k-T+1}\left(\frac{1}{\alpha}\right)^s\gamma_w^-\boldsymbol{I}. \end{aligned} \tag{E.7}$$

The resulting sum is geometric in $\frac{1}{\alpha}$. Using the well-known formula for the first several terms of a geometric series, we have

$$\begin{aligned}\boldsymbol{\Sigma}_i[k+1] &> \alpha^k \left(\frac{1 - (1/\alpha)^{k-T+2}}{1 - 1/\alpha} \right) \gamma_w^- \boldsymbol{I} \\ &= \frac{\alpha^{k+1} - \alpha^{T-1}}{\alpha - 1} \gamma_w^- \boldsymbol{I} \\ &= \frac{\alpha^{T-1}\left(1 - \alpha^{k-T+2}\right)}{1 - \alpha} \gamma_w^- \boldsymbol{I}. \end{aligned} \tag{E.8}$$

Since $k > T$, $k - T + 2 > 1$ so $\alpha^{k-T+2} < \alpha$ and $1 - \alpha^{k-T+2} > 1 - \alpha$. Therefore

$$\begin{aligned}\boldsymbol{\Sigma}_i[k+1] &> \frac{\alpha^{T+1}(1-\alpha)}{1-\alpha} \gamma_w^- \boldsymbol{I} \\ &= \alpha^{T+1} \gamma_w^- \boldsymbol{I}. \end{aligned} \tag{E.9}$$

We now have lower bounds for $\boldsymbol{\Sigma}$ for any value of k:

$$\boldsymbol{\Sigma}_i[k+1] > \begin{cases} \alpha^{T+1} \sigma_0 \boldsymbol{I} & \text{if } k \leq T \\ \alpha^{T+1} \gamma_w^- \boldsymbol{I} & \text{if } k > T. \end{cases} \tag{E.10}$$

Alternatively, we can write this bound as a single equation valid for all $k \in \mathbb{Z}_{\geq 0}$ by taking the minimum of the two cases to obtain

$$\boldsymbol{\Sigma}_i[k+1] > \min\{\sigma_0, \gamma_w^-\} \alpha^{T+1} \boldsymbol{I}. \tag{E.11}$$

Next, we find an upper bound for $\boldsymbol{\Sigma}_i[k+1]$. We start with the definition of $\boldsymbol{\Sigma}_i[k+1]$

in terms of $\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]$ in (E.2). Since $\alpha^{k+1} \leq 1$ for any $k \in \mathbb{Z}_{\geq 0}$, we have

$$\begin{aligned}\boldsymbol{\Sigma}_i[k+1] &= \alpha^{k+1}\sigma_0\boldsymbol{I} + \sum_{j=0}^{k} \alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j] \\ &\leq \sigma_0\boldsymbol{I} + \sum_{j=0}^{k} \alpha^{k-j}\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]. \end{aligned} \tag{E.12}$$

Now we selectively add more terms of the form $\boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]$ so that we can get sums of the form (4.28) and apply Assumption 4.5:

$$\begin{aligned}\boldsymbol{\Sigma}_i[k+1] &\leq \sigma_0\boldsymbol{I} + \sum_{s=0}^{k} \alpha^{k-s}\left(\sum_{j=s}^{s+T-1} \boldsymbol{w}_i[j]\boldsymbol{w}_i^\top[j]\right) \\ &< \sigma_0\boldsymbol{I} + \sum_{s=0}^{k} \alpha^{k-s}T\gamma_w^+\boldsymbol{I} \\ &= \sigma_0\boldsymbol{I} + \sum_{s=0}^{k} \alpha^{s}T\gamma_w^+\boldsymbol{I}. \end{aligned} \tag{E.13}$$

Again, we have a geometric series, which we can simplify using the well known formula to get

$$\boldsymbol{\Sigma}_i[k+1] < \sigma_0\boldsymbol{I} + \frac{1-\alpha^{k+1}}{1-\alpha}T\gamma_w^+\boldsymbol{I}. \tag{E.14}$$

Then since $1-\alpha^k < 1$ for any $k \in \mathbb{Z}_{>0}$, we can finally write this bound as

$$\boldsymbol{\Sigma}_i[k+1] = \left(\sigma_0 + \frac{\gamma_w^+T}{1-\alpha}\right)\boldsymbol{I}. \tag{E.15}$$

We have now shown that $\boldsymbol{\Sigma}_i[k+1]$ has both an upper and lower bound. Therefore

$$\min\{\sigma_0, \gamma_w^-\}\alpha^{T+1}\boldsymbol{I} < \boldsymbol{\Sigma}_i[k+1] < \left(\sigma_0 + \frac{\gamma_w^+ T}{1-\alpha}\right)\boldsymbol{I} \tag{E.16}$$

and since both bounds are positive definite, we can conclude that $\boldsymbol{\Sigma}_i[k+1]$ is always positive definite and thus invertible and there exists $\gamma_\Sigma^-, \gamma_\Sigma^+ \in \mathbb{R}{\geq}\, 0$ such that $\gamma_\Sigma^-\boldsymbol{I} < \boldsymbol{\Sigma}_i[k+1] < \gamma_\Sigma^+\boldsymbol{I}$.

Appendix F

Proof of Theorem 4.1

The proof of this theorem uses a Lyapunov approach. Rather than writing a Lyapunov function for the entire system right away, we consider some simpler candidate Lyapunov functions, which only relate to one aspect of the controller, and then add them together to get an overall Lyapunov function. The first differences of each Lyapunov function are not necessarily negative definite in all variables, but once added together the final Lyapunov function has the necessary properties. This approach has the advantage of being easer to follow as expressions are only written down when relevant. We consider separate Lyapunov functions for parameter estimation, stabilization, and extremum-seeking. Note that throughout this proof, $a_1, a_2, \ldots$ denote arbitrary constants from completing the squares and $c_1, c_2, \ldots$ denote positive coefficients.

The parameter estimation Lyapunov functions are written in terms of the deviation variables, $\widetilde{\boldsymbol{\theta}}_i[k] = \boldsymbol{\theta}_i[k] - \widehat{\boldsymbol{\theta}}_i[k]$ and $\widetilde{\eta}_i[l] = \eta_i[k] - \widehat{\eta}_i[k]$. The dynamics of the parameter estimation error are

$$\widetilde{\boldsymbol{\theta}}_i[k+1] \leq \widetilde{\boldsymbol{\theta}}_i[k] + \Delta\boldsymbol{\theta}_i[k+1] - \frac{\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]\left(e_i[k] - \widehat{\eta}[k]\right)}{\alpha + \boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]}. \tag{F.1}$$

Note that these dynamics are represented by an inequality due to the projection algorithm on the dynamics of $\widehat{\boldsymbol{\theta}}_i[k]$ from (4.22). The dynamics of the auxiliary variable estimation error are

$$\widetilde{\eta}_i[k+1 = \widetilde{\eta}_i[k] - K\widetilde{\eta}_i[k] - \boldsymbol{w}_i^\top[k+1]\left(\boldsymbol{\theta}_i[k+1] - \boldsymbol{\theta}_i[k]\right) + \Delta\widetilde{J}_i[k+1]. \tag{F.2}$$

We start by finding a Lyapunov function to show that the parameter estimation error is practically stable. An obvious candidate is

$$V_\theta[k] = \sum_{i=1}^{p} \widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k]. \tag{F.3}$$

By Lemma 4.4, $\boldsymbol{\Sigma}_i[k]$ is positive definite for all $k \in \mathbb{Z}_{\geq 0}$ so this function is positive definite and radially unbounded and hence a valid candidate Lyapunov function. Using (4.21) and (F.1), the dynamics of $\boldsymbol{\Sigma}_i$ and $\widetilde{\boldsymbol{\theta}}_i$, the first difference is

$$\begin{aligned}\Delta V_\theta[k+1] &= V_\theta[k+1] - V_\theta[k] \\ &= \sum_{i=1}^{p} \widetilde{\boldsymbol{\theta}}_i^\top[k+1]\boldsymbol{\Sigma}_i[k+1]\widetilde{\boldsymbol{\theta}}_i[k+1] - \sum_{i=1}^{p} \widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k]\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{p} \left(\widetilde{\boldsymbol{\theta}}_i[k] + \Delta\boldsymbol{\theta}_i[k+1] - \frac{\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]\left(e_i[k] - \widehat{\eta}[k]\right)}{\alpha + \boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]} \right)^\top \times \dots \\
&\quad \left(\alpha\boldsymbol{\Sigma}_i[k] + \boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k]\right) \times \dots \\
&\quad \left(\widetilde{\boldsymbol{\theta}}_i[k] + \Delta\boldsymbol{\theta}_i[k+1] - \frac{\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]\left(e_i[k] - \widehat{\eta}[k]\right)}{\alpha + \boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]} \right) \\
&\quad - \sum_{i=1}^{p} \widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k].
\end{aligned}
\tag{F.4}
$$

To simplify notation, let $\mu_{1,i}[k] = \widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{w}_i[k]$, $\mu_{2,i}[k] = \Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k]$, $\mu_{3,i}[k] = \boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]$, and $\mu_{4,i}[k] = e_i[k] - \widehat{\eta}_i[k]$. Using this notation, the Lyapunov difference is

$$
\begin{aligned}
\Delta V_\theta[k+1] = &\sum_{i=1}^{p} \left(\widetilde{\boldsymbol{\theta}}_i^\top[k] + \Delta\boldsymbol{\theta}_i^\top[k+1] - \frac{\boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\mu_{4,i}[k]}{\alpha + \mu_{3,i}[k]} \right) \times \dots \\
&\left(\alpha\boldsymbol{\Sigma}_i[k] + \boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k]\right) \times \dots \\
&\left(\widetilde{\boldsymbol{\theta}}_i[k] + \Delta\boldsymbol{\theta}_i[k+1] - \frac{\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]\mu_{4,i}[k]}{\alpha + \mu_{3,i}[k]} \right) \\
&- \widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k].
\end{aligned}
\tag{F.5}
$$

Since $\boldsymbol{\Sigma}_i[0] = \sigma_0\boldsymbol{I}$ and $\boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k]$ are symmetric, it is easy to inductively show that

$\boldsymbol{\Sigma}_i[k]$ is symmetric. Using this symmetry, we can expand the quadratic term to get

$$
\begin{aligned}
\Delta V_\theta[k+1] = \sum_{i=1}^{p} & \alpha \widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] + 2\alpha\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
& - 2\frac{\alpha\mu_{4,i}[k]}{\alpha+\mu_{3,i}[k]}\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k] \\
& + \alpha\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
& - 2\frac{\alpha\mu_{4,i}[k]}{\alpha+\mu_{3,i}[k]}\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k] \\
& + \frac{\alpha\mu_{4,i}^2[k]}{\left(\alpha+\mu_{3,i}[k]\right)^2}\boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{\Sigma}_i[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k] \\
& + \widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k]\widetilde{\boldsymbol{\theta}}_i[k] + 2\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k]\Delta\boldsymbol{\theta}_i[k+1] \\
& - 2\frac{\mu_{4,i}[k]}{\alpha+\mu_{3,i}[k]}\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k] \\
& + \Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k]\Delta\boldsymbol{\theta}_i[k+1] \\
& - 2\frac{\mu_{4,i}[k]}{\alpha+\mu_{3,i}[k]}\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k] \\
& + \frac{\mu_{4,i}^2[k]}{\left(\alpha+\mu_{3,i}[k]\right)^2}\boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k] \\
& - \widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k].
\end{aligned}
\tag{F.6}
$$

Next, we simplify this expression by letting $\boldsymbol{\Sigma}_i[k]\boldsymbol{\Sigma}_i^{-1}[k] = \boldsymbol{I}$, collecting terms, and grouping together squared terms where appropriate:

$$
\begin{aligned}
\Delta V_\theta[k+1] = \sum_{i=1}^{p} & -(1-\alpha)\,\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] + 2\alpha\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
& + \alpha\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] - 2\alpha\frac{\mu_{4,i}[k]}{\alpha+\mu_{3,i}[k]}\left(\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{w}_i[k]\right) \\
& - 2\alpha\frac{\mu_{4,i}[k]}{\alpha+\mu_{3,i}[k]}\left(\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k]\right) \\
& + \alpha\frac{\mu_{4,i}^2[k]}{(\alpha+\mu_{3,i}[k])^2}\left(\boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]\right) + \left(\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{w}_i[k]\right)^2 \\
& + 2\left(\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{w}_i[k]\right)\left(\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k]\right) \\
& - 2\frac{\mu_{4,i}[k]}{\alpha+\mu_{3,i}[k]}\left(\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{w}_i[k]\right)\left(\boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]\right) \\
& + \left(\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k]\right)^2 \\
& - 2\frac{\mu_{4,i}[k]}{\alpha+\mu_{3,i}[k]}\left(\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k]\right)\left(\boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]\right) \\
& + \frac{\mu_{4,i}^2[k]}{(\alpha+\mu_{3,i}[k])^2}\left(\boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]\right)^2.
\end{aligned}
\tag{F.7}
$$

The terms $\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{w}_i[k]$, $\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k]$, and $\boldsymbol{w}_i^\top[k]\boldsymbol{\Sigma}_i^{-1}[k]\boldsymbol{w}_i[k]$ appear several times in this expression. Conveniently, these terms are equal to the scalars $\mu_{1,i}[k]$, $\mu_{2,i}[k]$, and $\mu_{3,i}[k]$ which were recently defined. Making this substitution, we can write

$$\begin{aligned}\Delta V_\theta[k+1] = \sum_{i=1}^{p} & -(1-\alpha)\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] + 2\alpha\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\ & + \alpha\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] - 2\alpha\frac{\mu_{4,i}[k]}{\alpha+\mu_{3,i}[k]}\mu_{1,i}[k] \\ & - 2\alpha\frac{\mu_{4,i}[k]}{\alpha+\mu_{3,i}[k]}\mu_{2,i}[k] + \alpha\frac{\mu_{4,i}^2[k]}{(\alpha+\mu_{3,i}[k])^2}\mu_{3,i}[k] + \mu_{1,i}^2[k] \\ & + 2\mu_{1,i}[k]\mu_{2,i}[k] - 2\frac{\mu_{4,i}[k]}{\alpha+\mu_{3,i}[k]}\mu_{1,i}[k]\mu_{3,i}[k] + \mu_{2,i}^2[k] \\ & - 2\frac{\mu_{4,i}[k]}{\alpha+\mu_{3,i}[k]}\mu_{2,i}[k]\mu_{3,i}[k] + \frac{\mu_{4,i}^2[k]}{(\alpha+\mu_{3,i}[k])^2}\mu_{3,i}^2[k].\end{aligned} \tag{F.8}$$

By regrouping and canceling terms, this expression becomes

$$\begin{aligned}\Delta V_\theta[k+1] = \sum_{i=1}^{p} & -(1-\alpha)\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] + 2\alpha\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\ & + \alpha\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] + \mu_{1,i}^2[k] \\ & + 2\mu_{1,i}[k]\mu_{2,i}[k] + \mu_{2,i}^2[k] - 2\frac{\mu_{4,i}[k]\left(\alpha+\mu_{3,i}[k]\right)}{\alpha+\mu_{3,i}[k]}\mu_{1,i}[k] \\ & - 2\frac{\mu_{4,i}[k]\left(\alpha+\mu_{3,i}[k]\right)}{\alpha+\mu_{3,i}[k]}\mu_{2,i}[k] + \frac{\mu_{4,i}^2[k]\left(\alpha+\mu_{3,i}[k]\right)}{(\alpha+\mu_{3,i}[k])^2}\mu_{3,i}[k] \\ = \sum_{i=1}^{p} & -(1-\alpha)\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] + 2\alpha\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\ & + \alpha\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] + \left(\mu_{1,i}[k]+\mu_{2,i}[k]\right)^2 \\ & - 2\mu_{4,i}[k]\left(\mu_{1,i}[k]+\mu_{2,i}[k]\right) + \frac{\mu_{3,i}[k]}{\alpha+\mu_{3,i}[k]}\mu_{4,i}^2[k].\end{aligned} \tag{F.9}$$

Since $\mu_{3,i}[k]$ is a quadratic term, it is always positive. Therefore $\frac{\mu_{3,i}[k]}{\alpha+\mu_{3,i}[k]} < 1$ and we can bound the Lyapunov difference by

$$\begin{aligned}
\Delta V_\theta[k+1] < &\sum_{i=1}^{p} -(1-\alpha)\,\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] + 2\alpha\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
&+ \alpha\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] + \left(\mu_{1,i}[k]+\mu_{2,i}[k]\right)^2 \\
&- 2\mu_{4,i}[k]\left(\mu_{1,i}[k]+\mu_{2,i}[k]\right) + \mu_{4,i}^2[k] \\
= &\sum_{i=1}^{p} -(1-\alpha)\,\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] + 2\alpha\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
&+ \alpha\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] + \left(\mu_{1,i}[k]+\mu_{2,i}[k]-\mu_{4,i}[k]\right)^2.
\end{aligned} \tag{F.10}$$

Next, we replace $\mu_{1,i}[k]$, $\mu_{2,i}[k]$, and $\mu_{4,i}[k]$ by their definitions to obtain

$$\begin{aligned}
\Delta V_\theta[k+1] \leq &\sum_{i=1}^{p} -(1-\alpha)\,\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] + 2\alpha\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
&+ \alpha\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
&+ \left(\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{w}_i[k] + \Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k] - e_i[k] + \widehat{\eta}_i[k]\right)^2.
\end{aligned} \tag{F.11}$$

By rearranging the definition of $\eta_i[k]$, we have $e_i[k] = \eta_i[k] + \widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{w}_i[k]$. Substituting this expression for $e_i[k]$ gives

$$\begin{aligned}
\Delta V_\theta[k+1] \leq &\sum_{i=1}^{p} -(1-\alpha)\,\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] + 2\alpha\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
&+ \alpha\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
&+ \Bigg(\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{w}_i[k] + \Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k] - \eta_i[k] \\
&\quad - \widetilde{\boldsymbol{\theta}}^\top[k]\boldsymbol{w}_i[k] + \widehat{\eta}_i[k]\Bigg)^2.
\end{aligned} \tag{F.12}$$

Let $\widetilde{\eta}_i[k] = \eta_i[k] - \widehat{\eta}_i[k]$. Then the first difference is bounded by

$$\begin{aligned}
\Delta V_\theta[k+1] \leq& \sum_{i=1}^{p} -(1-\alpha)\,\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] + 2\alpha\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
&+ \alpha\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] + \left(\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k] - \widetilde{\eta}_i[k]\right)^2 \\
=& \sum_{i=1}^{p} -(1-\alpha)\,\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] + 2\alpha\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
&+ \alpha\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] - 2\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k]\widetilde{\eta}_i[k] \\
&+ \Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k]\Delta\boldsymbol{\theta}_i[k+1] + \widetilde{\eta}_i^2[k].
\end{aligned} \tag{F.13}$$

There is one remaining positive definite term. It can be bounded by completing the squares. Since $\boldsymbol{\Sigma}_i[k]$ is positive definite, for any $a_1 \in \mathbb{R}_{>0}$ we have that

$$\left(\widetilde{\boldsymbol{\theta}}_i[k] - a_1\Delta\boldsymbol{\theta}_i[k+1]\right)^\top \boldsymbol{\Sigma}_i[k]\left(\widetilde{\boldsymbol{\theta}}_i[k] - a_1\Delta\boldsymbol{\theta}_i[k+1]\right) \geq 0. \tag{F.14}$$

By rearranging this equation we can write an upper bound for the indefinite term as the sum of two positive definite terms. In this step, we additionally use the fact that $\boldsymbol{\Sigma}_i[k]$ is symmetric. This process gives

$$\begin{aligned}
0 \leq& \left(\widetilde{\boldsymbol{\theta}}_i[k] - a_1\Delta\boldsymbol{\theta}_i[k+1]\right)^\top \boldsymbol{\Sigma}_i[k]\left(\widetilde{\boldsymbol{\theta}}_i[k] - a_1\Delta\boldsymbol{\theta}_i[k+1]\right) \\
0 \leq& \widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] - 2a_1\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
&+ a_1^2\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \leq& \frac{1}{2a_1}\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] \\
&+ \frac{a_1}{2}\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1].
\end{aligned} \tag{F.15}$$

Similarly, for any $a_2 \in \mathbb{R}_{>0}$, we can complete the squares to bound the $\Delta\boldsymbol{\theta}_i^\top[k+$

$1]\boldsymbol{w}_i[k]\widetilde{\eta}_i[k]$ term by $\frac{a_2}{2}\left\|\Delta\boldsymbol{\theta}_i[k+1]\right\|^2\left\|\boldsymbol{w}_i[k]\right\| + \frac{1}{2a_2}\widetilde{\eta}_i^2[k]$. Using these two bounds, we can bound the Lyapunov difference in terms of definite functions only:

$$\begin{aligned}
\Delta V_\theta[k+1] \leq\ & \sum_{i=1}^{p} -\left(1-\alpha\right)\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] + \frac{\alpha}{a_1}\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] \\
& + \alpha a_1 \Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
& + \alpha\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] + a_2\left\|\Delta\boldsymbol{\theta}_i[k+1]\right\|^2\left\|\boldsymbol{w}_i[k]\right\| \\
& + \frac{1}{a_2}\widetilde{\eta}_i^2[k] + \Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{w}_i[k]\boldsymbol{w}_i^\top[k]\Delta\boldsymbol{\theta}_i[k+1] + \widetilde{\eta}_i^2[k] \\
=\ & \sum_{i=1}^{p} -\left(1-\alpha-\frac{\alpha}{a_1}\right)\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k] \\
& + \left(\alpha a_1+\alpha\right)\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1] \\
& + \left(1+a_2\right)\left\|\Delta\boldsymbol{\theta}_i[k+1]\right\|^2\left\|\boldsymbol{w}_i[k]\right\|^2 \\
& + \left(1+\frac{1}{a_2}\right)\widetilde{\eta}_i^2[k].
\end{aligned} \tag{F.16}$$

The quadratic terms $\widetilde{\boldsymbol{\theta}}_i^\top[k]\boldsymbol{\Sigma}_i[k]\widetilde{\boldsymbol{\theta}}_i[k]$ and $\Delta\boldsymbol{\theta}_i^\top[k+1]\boldsymbol{\Sigma}_i[k]\Delta\boldsymbol{\theta}_i[k+1]$ can be further bounded using Lemma 4.4:

$$\begin{aligned}
\Delta V_\theta[k+1] \leq\ & \sum_{i=1}^{p} -\left(1-\alpha-\frac{\alpha}{a_1}\right)\gamma_\Sigma^-\widetilde{\boldsymbol{\theta}}_i^\top[k]\widetilde{\boldsymbol{\theta}}_i[k] \\
& + \left(\alpha a_1+\alpha\right)\gamma_\Sigma^+\left\|\Delta\boldsymbol{\theta}_i[k+1]\right\|^2 \\
& + \left(1+a_2\right)\left\|\Delta\boldsymbol{\theta}_i[k+1]\right\|^2\left\|\boldsymbol{w}_i[k]\right\|^2 + \left(1+\frac{1}{a_2}\right)\widetilde{\eta}_i^2[k] \\
\leq\ & \sum_{i=1}^{p} -c_{15}\left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 + \left(\alpha a_1+\alpha\right)\gamma_\Sigma^+\left\|\Delta\boldsymbol{\theta}_i[k+1]\right\|^2 \\
& + \left(1+a_2\right)\left\|\Delta\boldsymbol{\theta}_i[k+1]\right\|^2\left\|\boldsymbol{w}_i[k]\right\|^2 + \left(1+\frac{1}{a_2}\right)\widetilde{\eta}_i^2[k]
\end{aligned} \tag{F.17}$$

where $c_{15} = (1 - \alpha - \frac{\alpha}{a_1})\gamma_\Sigma^-$. This constant is positive whenever $a_1 > \frac{\alpha}{1-\alpha}$, however since a_1 is arbitrary, it can be chosen to ensure that c_{15} is positive and the first term is therefore negative. Using the bound for $\|\boldsymbol{w}_i[k]\|$ from Assumption 4.5, we can further simplify this expression:

$$\begin{aligned}
\Delta V_\theta[k+1] \leq& \sum_{i=1}^{p} -c_{15} \left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 + (\alpha a_1 + \alpha)\,\gamma_\Sigma^+ \|\Delta\boldsymbol{\theta}_i[k+1]\|^2 \\
&+ (1 + a_2)\|\Delta\boldsymbol{\theta}_i[k+1]\|^2 \left(\sqrt{\gamma_w^+ T}\right)^2 + \left(1 + \frac{1}{a_2}\right)\widetilde{\eta}_i^2[k] \\
=& \sum_{i=1}^{p} -c_{15} \left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 + \left(1 + \frac{1}{a_2}\right)\widetilde{\eta}_i^2[k] \\
&+ \big((\alpha a_1 + \alpha)\,\gamma_\Sigma^+ + (1 + a_2)\gamma_w^+ T\big)\,\|\Delta\boldsymbol{\theta}_i[k+1]\|^2 \\
=& \sum_{i=1}^{p} -c_{15} \left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 + c_{16}\|\Delta\boldsymbol{\theta}_i[k+1]\|^2 + \left(1 + \frac{1}{a_2}\right)\widetilde{\eta}_i^2[k] \qquad \text{(F.18)}
\end{aligned}$$

where $c_{16} = (\alpha a_1 + \alpha)\gamma_\Sigma^+ + (1 + a_2)\gamma_w^+ T$. Since we know a bound for $\|\Delta\boldsymbol{\theta}_i[k+1]\|$, we are not concerned with the second term at this point. We do not know anything about the behaviour of $\widetilde{\eta}_i$ yet. Therefore, we consider the Lyapunov candidate

$$V_\eta[k] = \sum_{i=1}^{p} \widetilde{\eta}_i^2[k]. \qquad \text{(F.19)}$$

Since this function is the sum of quadratic scalar terms, it is positive definite and radially unbounded. Using (F.2), the dynamics of $\widetilde{\eta}_i[k]$, the first difference is

$$\begin{aligned}
\Delta V_\eta[k+1] &= V_\eta[k+1] - V_\eta[k] \\
&= \sum_{i=1}^{p} \widetilde{\eta}_i^2[k+1] - \sum_{i=1}^{p} \widetilde{\eta}_i^2[k]
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{p} \Big((1-K)\widetilde{\eta}_i[k] - \boldsymbol{w}_i^\top[k+1]\Delta\boldsymbol{\theta}_i[k+1] + \Delta\widetilde{J}_i[k+1]\Big)^2 \\
&\quad - \widetilde{\eta}_i^2[k].
\end{aligned} \tag{F.20}
$$

Next, we expand the squared term and collect terms to simplify this expression:

$$
\begin{aligned}
\Delta V_\eta[k+1] = &\sum_{i=1}^{p}(1-K)^2\widetilde{\eta}_i^2[k] - 2(1-K)\widetilde{\eta}_i[k]\boldsymbol{w}_i^\top[k+1]\Delta\boldsymbol{\theta}_i[k+1] \\
&+ 2(1-K)\widetilde{\eta}_i[k]\Delta\widetilde{J}_i[k+1] + \big(\boldsymbol{w}_i^\top[k+1]\Delta\boldsymbol{\theta}_i[k+1]\big)^2 \\
&- 2\boldsymbol{w}_i^\top[k+1]\Delta\boldsymbol{\theta}_i[k+1]\Delta\widetilde{J}_i[k+1] + \Delta\widetilde{J}_i^2[k+1] - \widetilde{\eta}_i^2[k] \\
= &\sum_{i=1}^{p} - \left(2K-K^2\right)\widetilde{\eta}_i^2[k] - 2(1-K)\widetilde{\eta}_i[k]\boldsymbol{w}_i^\top[k+1]\Delta\boldsymbol{\theta}_i[k+1] \\
&+ 2(1-K)\widetilde{\eta}_i[k]\Delta\widetilde{J}_i[k+1] + \big(\boldsymbol{w}_i^\top[k+1]\Delta\boldsymbol{\theta}_i[k+1]\big)^2 \\
&+ 2\left\|\boldsymbol{w}_i[k+1]\right\|\left\|\Delta\boldsymbol{\theta}_i[k+1]\right\|\left\|\Delta\widetilde{J}_i[k+1]\right\| + \Delta\widetilde{J}_i^2[k+1].
\end{aligned} \tag{F.21}
$$

We can remove the indefinite terms by completing the squares. For any arbitrary $a_3, \ldots, a_5 \in \mathbb{R}_{>0}$, we have

$$
\begin{aligned}
\Delta V_\eta[k+1] \leq &\sum_{i=1}^{p} - \left(2K-K^2\right)\widetilde{\eta}_i^2[k] + a_3(1-K)\widetilde{\eta}_i^2[k] \\
&+ \frac{1-K}{a_3}\big(\boldsymbol{w}_i^\top[k+1]\Delta\boldsymbol{\theta}_i[k+1]\big)^2 + a_4(1-K)\widetilde{\eta}_i^2[k] \\
&+ \frac{1-K}{a_4}\Big(\Delta\widetilde{J}_i[k+1]\Big)^2 + \big(\boldsymbol{w}_i^\top[k+1]\Delta\boldsymbol{\theta}_i[k+1]\big)^2 \\
&+ a_5\left\|\boldsymbol{w}_i[k+1]\right\|^2\left\|\Delta\boldsymbol{\theta}_i[k+1]\right\|^2 + \frac{1}{a_5}\left\|\Delta\widetilde{J}_i[k+1]\right\|^2 \\
&+ \Big(\Delta\widetilde{J}_i[k+1]\Big)^2
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{i=1}^{p} - \left(2K - K^2 - (a_3 + a_4)(1-K)\right) \widetilde{\eta}_i^2[k] \\
&\quad + \left(\frac{1-K}{a_3} + 1 + a_5\right) \|\boldsymbol{w}_i[k+1]\|^2 \|\Delta\boldsymbol{\theta}_i[k+1]\|^2 \\
&\quad + \left(1 + \frac{1-K}{a_4} + \frac{1}{a_5}\right) \left\|\Delta\widetilde{J}_i[k+1]\right\|^2 .
\end{aligned} \tag{F.22}
$$

Using the bound on $\|\boldsymbol{w}_i[k+1]\|$ from Assumption 4.5, we can further simplify this expression as

$$
\begin{aligned}
\Delta V_\eta[k+1] &\leq \sum_{i=1}^{p} -c_{17}\widetilde{\eta}_i^2[k] + \left(\frac{1-K}{a_3} + 1 + a_5\right) \gamma_w^+ T \|\Delta\boldsymbol{\theta}_i[k+1]\|^2 \\
&\quad + \left(1 + \frac{1-K}{a_4} + \frac{1}{a_5}\right) \left\|\Delta\widetilde{J}_i[k+1]\right\|^2 \\
&= \sum_{i=1}^{p} -c_{17}\widetilde{\eta}_i^2[k] + c_{18} \|\Delta\boldsymbol{\theta}_i[k+1]\|^2 + c_{19} \left\|\Delta\widetilde{J}_i[k+1]\right\|^2 .
\end{aligned} \tag{F.23}
$$

where $c_{17} = (2K - K^2 - (a_3 + a_4)(1-K))$, $c_{18} = (\frac{1-K}{a_3} + 1 + a_5)\gamma_w^+ T$, and $c_{19} = 1 + \frac{1-K}{a_4} + \frac{1}{a_5}$. Note that since $K \in (0,1)$, we can always choose a_3 and a_4 such that $a_3 + a_4 < K$ which ensures that $c_{17} > 0$. Note that all of the terms are either negative definite or have already been shown to be bounded whenever the ESC stabilizes and optimizes the system. Therefore, we can interpret V_η as a Lyapunov function that shows that the $\widetilde{\eta}$ dynamics are uniformly asymptotically bounded. The size of the invariant set that $\widetilde{\eta}$ converges to is inversely proportional to a_3 and a_4. Therefore, we want to make a_3 and a_4 as large as possible. Since these are constants obtained by completing the square, they can be chosen arbitrarily. However, to ensure the positivity of c_{17}, we have the additional constraint that $a_3 + a_4 < K$. Therefore the size of the set that $\widetilde{\eta}$ converges to can be made smaller by increasing K. This set cannot be made arbitrarily small as K must be less than one. This limitation is a fundamental limitation of a discrete-time algorithm.

The size of the set could also be made arbitrarily small by reducing the size of the time step which would effectively cause $\boldsymbol{f}$ and $\boldsymbol{G}$ to scale proportional to the time step. The scaled vector fields would have proportionally scaled Lipschitz constants, L_f and L_G which would ultimately result in an arbitrarily small bounds on $\|\Delta\boldsymbol{\theta}_i[k+1]\|$ and $\|\Delta\widetilde{J}_i[k+1]\|$. Furthermore, the bound on $\|\Delta\widetilde{J}_i[k+1]\|$ can be made arbitrarily small by reducing the time step towards zero.

Now that we have candidate Lyapunov functions for the $\widetilde{\boldsymbol{\theta}}_i$ and $\widetilde{\eta}_i$ dynamics, we can add them to get the overall parameter estimation Lyapunov function:

$$V_{\mathrm{PE}}[k] = V_\theta[k] + \zeta V_\eta[k] \tag{F.24}$$

where $\zeta \in \mathbb{R}_{>0}$ is a positive constant which will be chosen later to make terms negative definite. Using the differences we have already calculated for V_θ and V_η, it is straightforward to compute the first difference for the parameter estimation Lyapunov function:

$$\begin{aligned}
\Delta V_{\mathrm{PE}}[k+1] &= \Delta V_\theta[k+1] + \zeta \Delta V_\eta[k+1] \\
&\leq \sum_{i=1}^{p} -c_{15}\left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 + c_{16}\left\|\Delta\boldsymbol{\theta}_i[k+1]\right\|^2 + \left(1+\frac{1}{a_2}\right)\widetilde{\eta}_i^2[k] \\
&\quad + \zeta\left(\sum_{i=1}^{p} -c_{17}\widetilde{\eta}_i^2[k] + c_{18}\left\|\Delta\boldsymbol{\theta}_i[k+1]\right\|^2 + c_{19}\left\|\Delta\widetilde{J}_i[k+1]\right\|^2\right) \\
&= \sum_{i=1}^{p} -c_{15}\left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 - \left(\zeta c_{17} - 1 - \frac{1}{a_2}\right)\widetilde{\eta}_i^2[k] \\
&\quad + (c_{16}+\zeta c_{18})\left\|\Delta\boldsymbol{\theta}_i[k+1]\right\|^2 + \zeta c_{19}\left\|\Delta\widetilde{J}_i[k+1]\right\|^2
\end{aligned}$$

$$\begin{aligned} &= \sum_{i=1}^{p} -c_{15} \left\| \widetilde{\boldsymbol{\theta}}_i[k] \right\|^2 - c_{20} \widetilde{\eta}_i^2[k] + c_{21} \left\| \Delta \boldsymbol{\theta}_i[k+1] \right\|^2 \\ &\quad + \zeta c_{19} \left\| \Delta \widetilde{J}_i[k+1] \right\|^2 \end{aligned} \tag{F.25}$$

where $c_{20} = \zeta c_{17} - 1 - \frac{1}{a_2}$ and $c_{21} = c_{16} + \zeta c_{18}$. Since c_{17} is positive, we can ensure that $c_{20} > 0$ by choosing $\zeta > \frac{a_2+1}{c_{17}a_2}$. Such a ζ always exists since the only restriction on ζ is that it must be positive to ensure the V_{PE} is indeed positive definite. Note that if we choose ζ to be very large, the coefficients multiplying the other terms in the Lyapunov difference become large, which results in a more conservative estimate of the neighbourhood that the parameter estimates converge to.

At this point, we can interpret what this Lyapunov function means. The first two terms are negative definite terms involving $\widetilde{\boldsymbol{\theta}}$ and $\widetilde{\eta}$. The third and fourth terms are positive terms involving $\Delta\boldsymbol{\theta}$ and $\Delta\widetilde{J}$. Suppose that the system does not change too fast. Then $\Delta\boldsymbol{\theta}$ and $\Delta\widetilde{J}$ are bounded and we can treat the last three terms as a constant term. Whenever $\widetilde{\boldsymbol{\theta}}$ or $\widetilde{\eta}$ are large, the negative definite terms dominate this positive constant, and the Lyapunov function decreases. A decreasing Lyapunov function means that $\widetilde{\boldsymbol{\theta}}$ and $\widetilde{\eta}$ must decrease. Eventually $\widetilde{\boldsymbol{\theta}}$ and $\widetilde{\eta}$ get small enough that the negative definite terms no longer bound the positive term. At this point, $\widetilde{\boldsymbol{\theta}}$ and $\widetilde{\eta}$ cannot increase, or else the Lyapunov difference would be negative and they would decrease. Therefore $\widetilde{\boldsymbol{\theta}}$ and $\widetilde{\eta}$ must remain in some neighbourhood of the origin. The system has converged to a level set of the Lyapunov function which it cannot leave. Therefore, if $\Delta\boldsymbol{\theta}$ and $\Delta\widetilde{J}$ are bounded, then the parameter estimation algorithm is uniformly asymptotically bounded.

We have discovered that the convergence of the parameter estimates depends on the assumption that $\boldsymbol{\theta}$ and $\widetilde{J}$ do not change too rapidly. Fortunately, we have already shown that for a system controlled by ESC, when k is sufficiently large, $\Delta\boldsymbol{\theta}$ and $\Delta\widetilde{J}$ can be bounded by a function of $\widetilde{\boldsymbol{x}}$, $\widetilde{\boldsymbol{u}}$ and a constant. Using these bounds, for a large enough k we can write the first difference of the parameter estimation Lyapunov function as

$$\begin{aligned}\Delta V_{\mathrm{PE}}[k+1] \leq \sum_{i=1}^{p} & -c_{15}\left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 - c_{20}\widetilde{\eta}_i^2[k] \\ & + c_{21}\left(c_6\left\|\widetilde{\boldsymbol{x}}[k]\right\| + c_7\left\|\widetilde{\boldsymbol{u}}[k]\right\| + c_8K_g + c_9\frac{1}{\tau_I} + c_{10}D\right)^2 \\ & + \zeta c_{19}\left(c_{11}\left\|\widetilde{\boldsymbol{x}}[k]\right\| + c_{12}\left\|\widetilde{\boldsymbol{u}}[k]\right\| + c_{13}K_g + c_{14}D\right)^2.\end{aligned} \tag{F.26}$$

By expanding the squared terms, and collecting similar terms, we can simplify this bound as

$$\begin{aligned}\Delta V_{\mathrm{PE}}[k+1] \leq \sum_{i=1}^{p} & -c_{15}\left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 - c_{20}\widetilde{\eta}_i^2[k] + \left(c_6^2c_{21} + \zeta c_{11}^2c_{19}\right)\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \\ & + \left(c_7^2c_{21} + \zeta c_{12}^2c_{19}\right)\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 \\ & + 2\left(c_6c_7c_{21} + \zeta c_{11}c_{12}c_{19}\right)\left\|\widetilde{\boldsymbol{x}}[k]\right\|\left\|\widetilde{\boldsymbol{u}}[k]\right\| \\ & + 2\left(c_6c_8c_{21} + \zeta c_{11}c_{13}c_{19}\right)K_g\left\|\widetilde{\boldsymbol{x}}[k]\right\| \\ & + 2c_6c_9c_{21}\frac{1}{\tau_I}\left\|\widetilde{\boldsymbol{x}}[k]\right\| + 2\left(c_6c_{10}c_{21} + \zeta c_{11}c_{14}c_{19}\right)D\left\|\widetilde{\boldsymbol{x}}[k]\right\| \\ & + 2\left(c_7c_8c_{21} + \zeta c_{12}c_{13}c_{19}\right)K_g\left\|\widetilde{\boldsymbol{u}}[k]\right\| + 2c_7c_9c_{21}\frac{1}{\tau_I}\left\|\widetilde{\boldsymbol{u}}[k]\right\| \\ & + 2\left(c_7c_{10}c_{21} + \zeta c_{12}c_{14}c_{19}\right)D\left\|\widetilde{\boldsymbol{u}}[k]\right\| \\ & + \left(c_8c_{21}\left(c_8K_g + c_9\frac{1}{\tau_I} + c_{10}D\right) + \zeta c_{13}c_{19}\left(c_{13}K_g + c_{14}D\right)\right)K_g \\ & + c_9c_{21}\left(c_8K_g + c_9\frac{1}{\tau_I} + c_{10}D\right)\frac{1}{\tau_I} \\ & + \left(c_{10}c_{21}\left(c_8K_g + c_9\frac{1}{\tau_I} + c_{10}D\right) + \zeta c_{14}c_{19}\left(c_{13}K_gc_{14}D\right)\right)D\end{aligned}$$

$$\begin{aligned}= \sum_{i=1}^{p} -c_{15} \left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 - c_{20}\widetilde{\eta}_i^2[k] + c_{22}\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 + c_{23}\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 \\ + c_{24}\left\|\widetilde{\boldsymbol{x}}[k]\right\|\left\|\widetilde{\boldsymbol{u}}[k]\right\| + c_{25}K_g\left\|\widetilde{\boldsymbol{x}}[k]\right\| + c_{26}\frac{1}{\tau_I}\left\|\widetilde{\boldsymbol{x}}[k]\right\| \\ + c_{27}D\left\|\widetilde{\boldsymbol{x}}[k]\right\| + c_{28}K_g\left\|\widetilde{\boldsymbol{u}}[k]\right\| + c_{29}\frac{1}{\tau_I}\left\|\widetilde{\boldsymbol{u}}[k]\right\| \\ + c_{30}D\left\|\widetilde{\boldsymbol{u}}[k]\right\| + c_{31}K_g + c_{32}\frac{1}{\tau_I} + c_{33}D.\end{aligned} \tag{F.27}$$

The various positive constants are defined by

$$\begin{aligned}
c_{22} &= c_6^2c_{21} + \zeta c_{11}^2c_{19} \\
c_{23} &= c_7^2c_{21} + \zeta c_{12}^2c_{19} \\
c_{24} &= 2\left(c_6c_7c_{21} + \zeta c_{11}c_{12}c_{19}\right) \\
c_{25} &= 2\left(c_6c_8c_{21} + \zeta c_{11}c_{13}c_{19}\right) \\
c_{26} &= 2c_7c_9c_{21} \\
c_{27} &= 2\left(c_6c_{10}c_{21} + \zeta c_{11}c_{14}c_{19}\right) \\
c_{28} &= 2\left(c_7c_8c_{21} + \zeta c_{12}c_{13}c_{19}\right) \\
c_{29} &= 2c_7c_9c_{21} \\
c_{30} &= 2\left(c_7c_{10}c_{21} + \zeta c_{12}c_{14}c_{19}\right) \\
c_{31} &= c_8c_{21}\left(c_8K_g + c_9\frac{1}{\tau_I} + c_{10}D\right) + \zeta c_{13}c_{19}\left(c_{13}K_g + c_{14}D\right) \\
c_{32} &= c_9c_{21}\left(c_8K_g + c_9\frac{1}{\tau_I} + c_{10}D\right) \\
c_{33} &= c_{10}c_{21}\left(c_8K_g + c_9\frac{1}{\tau_I} + c_{10}D\right) + \zeta c_{14}c_{19}\left(c_{13}K_gc_{14}D\right).
\end{aligned}$$

Next we complete the squares and simplify the resulting expression to write the bound entirely in terms of linear functions of K_g, $\frac{1}{\tau_I}$, and D and quadratic functions of the other variables:

$$\begin{aligned}
\Delta V_{\mathrm{PE}}[k+1] \leq\ & \sum_{i=1}^{p} -c_{15}\left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 - c_{20}\widetilde{\eta}_i^2[k] + c_{22}\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 + c_{23}\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 \\
& + \frac{a_6 c_{24}}{2}\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 + \frac{c_{24}}{2a_6}\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 + \frac{a_7 c_{25}}{2}K_g K_g + \frac{c_{25}}{2a_7}\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \\
& + \frac{a_8 c_{26}}{2}\frac{1}{\tau_I}\frac{1}{\tau_I} + \frac{c_{26}}{2a_8}\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 + \frac{a_9 c_{27}}{2}DD + \frac{c_{27}}{2a_9}\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \\
& + \frac{a_{10} c_{28}}{2}K_g K_g + \frac{c_{28}}{2a_{10}}\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 + \frac{a_{11} c_{29}}{2}\frac{1}{\tau_I}\frac{1}{\tau_I} + \frac{c_{29}}{2a_{11}}\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 \\
& + \frac{a_{12} c_{30}}{2}DD + \frac{c_{30}}{2a_{12}}\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 + c_{31}K_g + c_{32}\frac{1}{\tau_I} + c_{33}D \\
\leq\ & \sum_{i=1}^{p} -c_{15}\left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 - c_{20}\widetilde{\eta}_i^2[k] \\
& + \left(c_{22} + \frac{a_6 c_{24}}{2} + \frac{c_{25}}{2a_7} + \frac{c_{26}}{2a_8} + \frac{c_{27}}{2a_9}\right)\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \\
& + \left(c_{23} + \frac{c_{24}}{2a_6} + \frac{c_{28}}{2a_{10}} + \frac{c_{29}}{2a_{11}} + \frac{c_{30}}{2a_{12}}\right)\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 \\
& + \left(c_{31} + \frac{a_7 c_{25}}{2}K_g + \frac{a_{10} c_{28}}{2}K_g\right)K_g \\
& + \left(c_{32} + \frac{a_8 c_{26}}{2}\frac{1}{\tau_I} + \frac{a_{11} c_{29}}{2}\frac{1}{\tau_I}\right)\frac{1}{\tau_I} \\
& + \left(c_{33} + \frac{a_9 c_{27}}{2}D + \frac{a_{12} c_{30}}{2}D\right)D \\
=\ & \sum_{i=1}^{p} -c_{15}\left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 - c_{20}\widetilde{\eta}_i^2[k] + c_{34}\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 + c_{35}\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 \\
& + c_{36}K_g + c_{37}\frac{1}{\tau_I} + c_{38}D
\end{aligned} \tag{F.28}$$

where $a_6, \ldots, a_{12} \in \mathbb{R}$ are arbitrary and the positive constants are defined by

$$\begin{aligned}
c_{34} &= c_{22} + \frac{a_6 c_{24}}{2} + \frac{c_{25}}{2a_7} + \frac{c_{26}}{2a_8} + \frac{c_{27}}{2a_9} \\
c_{35} &= c_{23} + \frac{c_{24}}{2a_6} + \frac{c_{28}}{2a_{10}} + \frac{c_{29}}{2a_{11}} + \frac{c_{30}}{2a_{12}}
\end{aligned}$$

$$
\begin{aligned}
c_{36} &= c_{31} + \frac{a_7 c_{25}}{2} K_g + \frac{a_{10} c_{28}}{2} K_g \\
c_{37} &= c_{32} + \frac{a_8 c_{26}}{2} \frac{1}{\tau_I} + \frac{a_{11} c_{29}}{2} \frac{1}{\tau_I} \\
c_{38} &= c_{33} + \frac{a_9 c_{27}}{2} D + \frac{a_{12} c_{30}}{2} D.
\end{aligned}
$$

Now suppose that $\widetilde{\boldsymbol{x}}[k]$ and $\widetilde{\boldsymbol{u}}[k]$ converge to zero. Then since K_g, $\frac{1}{\tau_I}$, and D are all constants, these Lyapunov difference is negative whenever $\widetilde{\boldsymbol{\theta}}_i[k]$ and $\widetilde{\eta}_i[k]$ are large. Therefore this function tells us that when $\widetilde{\boldsymbol{x}}[k]$ and $\widetilde{\boldsymbol{u}}[k]$ converge, then the parameter estimates converge to a neighbourhood of the origin whose size is $\mathcal{O}(K_G + \frac{1}{\tau_I} + D)$. It remains to show that $\widetilde{\boldsymbol{x}}[k]$ and $\widetilde{\boldsymbol{u}}[k]$ converge.

In the second part of the proof, we show that the controller is able to stabilize $\boldsymbol{x}$ to the steady-state manifold $\boldsymbol{\pi}(\widehat{\boldsymbol{u}})$. Consider the Lyapunov function

$$
V_x[k] = V_z\left(\boldsymbol{z}[k]\right) + H\left(\boldsymbol{x}[k]\right). \tag{F.29}
$$

Using the state dynamics, (4.1), the first difference of this Lyapunov function is

$$
\begin{aligned}
\Delta V_x[k+1] &= V_x[k+1] - V_x[k] \\
&= V_z\left(\boldsymbol{z}[k+1]\right) + H\left(\boldsymbol{x}[k+1]\right) - V_z\left(\boldsymbol{z}[k]\right) - H\left(\boldsymbol{x}[k]\right) \\
&= V_z\left(\boldsymbol{z}[k+1]\right) + H\left(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\boldsymbol{u}[k]\right) \\
&\quad - V_z\left(\boldsymbol{z}[k]\right) - H\left(\boldsymbol{x}[k]\right).
\end{aligned} \tag{F.30}
$$

Substituting in the control law (4.24) for $\boldsymbol{u}[k]$ into this expression results in

$$
\begin{aligned}
\Delta V_x[k+1] = {} & V_z\left(\boldsymbol{z}[k+1]\right) - V_z\left(\boldsymbol{z}[k]\right) - H\left(\boldsymbol{x}[k]\right) \\
& + H\Big(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\Big(-K_g\widehat{\boldsymbol{\theta}}_1[k] + \widehat{\boldsymbol{u}}[k] + \boldsymbol{d}[k]\Big)\Big).
\end{aligned} \tag{F.31}
$$

We would like to be able to bound this expression using Assumption 4.2. However, a stabilizing controller should be written in terms of $\boldsymbol{\theta}_1[k]$ instead of $\widehat{\boldsymbol{\theta}}_1[k]$. Therefore we add and subtract $K_g\boldsymbol{\theta}_1[k]$ and use $\widetilde{\boldsymbol{\theta}}_1[k] = \boldsymbol{\theta}_1[k] - \widehat{\boldsymbol{\theta}}_1[k]$ to simplify the resulting expression:

$$\begin{aligned}
\Delta V_x[k+1] &= V_z\left(\boldsymbol{z}[k+1]\right) - V_z\left(\boldsymbol{z}[k]\right) - H\left(\boldsymbol{x}[k]\right) \\
&\quad + H\Big(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right) \times \dots \\
&\qquad \Big(-K_g\boldsymbol{\theta}_1[k] + K_g\boldsymbol{\theta}_1[k] - K_g\widehat{\boldsymbol{\theta}}_1[k] + \widehat{\boldsymbol{u}}[k] + \boldsymbol{d}[k]\Big)\Big) \\
&= V_z\left(\boldsymbol{z}[k+1]\right) - V_z\left(\boldsymbol{z}[k]\right) - H\left(\boldsymbol{x}[k]\right) \\
&\quad + H\Big(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right) \times \dots \\
&\qquad \Big(-K_g\boldsymbol{\theta}_1[k] + K_g\widetilde{\boldsymbol{\theta}}_1[k] + \widehat{\boldsymbol{u}}[k] + \boldsymbol{d}[k]\Big)\Big).
\end{aligned} \tag{F.32}$$

Furthermore, the stabilizability assumption uses a gain of K_g^* instead of K_g. Therefore we add and subtract $K_g^*\boldsymbol{\theta}_1[k]$ as well:

$$\begin{aligned}
\Delta V_x[k+1] &= V_z\left(\boldsymbol{z}[k+1]\right) - V_z\left(\boldsymbol{z}[k]\right) - H\left(\boldsymbol{x}[k]\right) \\
&\quad + H\Big(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\big(-K_g^*\boldsymbol{\theta}_1[k] + \dots \\
&\qquad + K_g^*\boldsymbol{\theta}_1[k] - K_g\boldsymbol{\theta}_1[k] + K_g\widetilde{\boldsymbol{\theta}}_1[k] + \widehat{\boldsymbol{u}}[k] + \boldsymbol{d}[k]\big)\Big) \\
&= V_z\left(\boldsymbol{z}[k+1]\right) - V_z\left(\boldsymbol{z}[k]\right) - H\left(\boldsymbol{x}[k]\right) \\
&\quad + H\Big(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\big(\widehat{\boldsymbol{u}}[k] - K_g^*\boldsymbol{\theta}_1[k]\big) + \dots \\
&\qquad + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\Big(\big(K_g^* - K_g\big)\boldsymbol{\theta}_1[k] + K_g\widetilde{\boldsymbol{\theta}}_1[k] + \boldsymbol{d}[k]\Big)\Big).
\end{aligned} \tag{F.33}$$

Now, we would like to separate the last term into two terms so that we can apply

Assumption 4.2. By the mean-value theorem, there exists $\overline{\boldsymbol{x}} \in \mathbb{X}$ such that

$$\begin{aligned}\Delta V_x[k+1] = {}& V_z\left(\boldsymbol{z}[k+1]\right) - V_z\left(\boldsymbol{z}[k]\right) - H\left(\boldsymbol{x}[k]\right) \\ & + H\left(\boldsymbol{x}[k] + \boldsymbol{f}\left(\boldsymbol{x}[k]\right) + \boldsymbol{G}\left(\boldsymbol{x}[k]\right)\left(\widehat{\boldsymbol{u}}[k] - K_g^*\boldsymbol{\theta}_1[k]\right)\right) \\ & + \nabla H(\overline{\boldsymbol{x}})\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\Big(\left(K_g^* - K_g\right)\boldsymbol{\theta}_1[k] + K_g\widetilde{\boldsymbol{\theta}}_1[k] + \boldsymbol{d}[k]\Big).\end{aligned} \tag{F.34}$$

Then by Assumption 4.2, the system is stabilizable, so we can bound this expression as

$$\begin{aligned}\Delta V_x[k+1] \leq {}& -\beta_4\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \\ & + \nabla H(\overline{\boldsymbol{x}})\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\Big(\left(K_g^* - K_g\right)\boldsymbol{\theta}_1[k] + K_g\widetilde{\boldsymbol{\theta}}_1[k] + \boldsymbol{d}[k]\Big).\end{aligned} \tag{F.35}$$

The last term resembles (4.6), the definition of $\boldsymbol{\theta}_{1,i}[k]$. By writing this term as a sum, we can then apply this definition to obtain

$$\begin{aligned}\Delta V_x[k+1] \leq {}& -\sum_{i=1}^{p} -\frac{\beta_4}{p}\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \\ & + \nabla H(\overline{\boldsymbol{x}})\boldsymbol{G}_i\left(\boldsymbol{x}[k]\right)\Big(\left(K_g^* - K_g\right)\boldsymbol{\theta}_{1,i}[k] + K_g\widetilde{\boldsymbol{\theta}}_{1,i}[k] + \boldsymbol{d}_i[k]\Big) \\ = {}& -\sum_{i=1}^{p} -\frac{\beta_4}{p}\beta_4\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \\ & + p\boldsymbol{\theta}_{1,i}^{\top}[k]\Big(-\left(K_g - K_g^*\right)\boldsymbol{\theta}_{1,i}[k] + K_g\widetilde{\boldsymbol{\theta}}_{1,i}[k] + \boldsymbol{d}_i[k]\Big).\end{aligned} \tag{F.36}$$

Next, we expand the last term to obtain

$$\begin{aligned}\Delta V_x[k+1] \leq {}& \sum_{i=1}^{p} -\frac{\beta_4}{p}\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 - p\left(K_g - K_g^*\right)\boldsymbol{\theta}_{1,i}^{\top}[k]\boldsymbol{\theta}_{1,i}[k] \\ & + pK_g\boldsymbol{\theta}_{1,i}^{\top}[k]\widetilde{\boldsymbol{\theta}}_{1,i}[k] + p\boldsymbol{\theta}_{1,i}^{\top}[k]\boldsymbol{d}_i[k].\end{aligned} \tag{F.37}$$

We bound the indefinite term involving the dither signal by completing the squares. For any $a_{13} \in \mathbb{R}_{>0}$ we have the bound

$$\begin{aligned}
\Delta V_x[k+1] \leq & \sum_{i=1}^{p} -\frac{\beta_4}{p} \left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 - p\left(K_g - K_g^*\right) \left\|\boldsymbol{\theta}_{1,i}[k]\right\|^2 \\
& + pK_g \left\|\boldsymbol{\theta}_{1,i}[k]\right\| \left\|\widetilde{\boldsymbol{\theta}}_{1,i}[k]\right\| + \frac{p}{2a_{13}} \left\|\boldsymbol{\theta}_{1,i}[k]\right\|^2 + \frac{pa_{13}}{2} \left\|\boldsymbol{d}_i[k]\right\|^2 \\
= & \sum_{i=1}^{p} -\frac{\beta_4}{p} \left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 - p\left(K_g - K_g^* - \frac{1}{2a_{13}}\right) \left\|\boldsymbol{\theta}_{1,i}[k]\right\|^2 \\
& + pK_g \left\|\boldsymbol{\theta}_{1,i}[k]\right\| \left\|\widetilde{\boldsymbol{\theta}}_{1,i}[k]\right\| + \frac{pa_{13}}{2} \left\|\boldsymbol{d}_i[k]\right\|^2 .
\end{aligned} \tag{F.38}$$

By choosing $K_g > K_g^* + \frac{1}{2a_{13}}$, we can ensure that the first two terms are negative definite. Assuming that the parameter estimates converge, this Lyapunov function tells us that the proportional controller can stabilize the nonlinear system to a neighbourhood the steady-state manifold $\boldsymbol{\pi}(\widehat{\boldsymbol{u}})$ provided that the proportional gain is large enough. The size of this neighbourhood depends on the dither amplitude.

Since this Lyapunov function does not say anything about the convergence of parameter estimates, we add it to the parameter estimate Lyapunov function to get

$$V_{\text{stab}}[k] = V_{\text{PE}}[k] + V_x[k]. \tag{F.39}$$

Its first difference can easily be computed by adding the first differences already computed for V_{PE} and V_x:

$$\Delta V_{\text{stab}}[k+1] = \Delta V_{\text{PE}}[k+1] + \Delta V_x[k+1]$$

$$\begin{aligned}
&\leq \sum_{i=1}^{p} -c_{15} \left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 - c_{15} \left\|\widetilde{\boldsymbol{\theta}}_i[k]\right\|^2 - c_{20}\widetilde{\eta}_i^2[k] + c_{34} \left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \\
&\quad + c_{35} \left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 + c_{36}K_g + c_{37}\frac{1}{\tau_I} + c_{38}D \\
&\quad + \sum_{i=1}^{p} -\frac{\beta_4}{p} \left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 - p\left(K_g - K_g^* - \frac{1}{2a_{13}}\right) \left\|\boldsymbol{\theta}_{1,i}[k]\right\|^2 \\
&\quad + pK_g \left\|\boldsymbol{\theta}_{1,i}[k]\right\| \left\|\widetilde{\boldsymbol{\theta}}_{1,i}[k]\right\| + \frac{pa_{13}}{2} \left\|\boldsymbol{d}_i[k]\right\|^2 .
\end{aligned} \tag{F.40}$$

Next we collect terms to simplify. Since $\|\widetilde{\boldsymbol{\theta}}_i[k]\|^2 = \|\widetilde{\boldsymbol{\theta}}_{0,i}[k]\|^2 + \|\widetilde{\boldsymbol{\theta}}_{1,i}[k]\|^2$, we can split up the first terms into two terms. One of the terms can then be expressed along with two other terms in a matrix form:

$$\begin{aligned}
\Delta V_{\text{stab}}[k+1] = &\sum_{i=1}^{p} -c_{15} \left\|\widetilde{\boldsymbol{\theta}}_{0,i}[k]\right\|^2 - c_{20} \left\|\widetilde{\eta}_i[k]\right\|^2 - \left(\frac{\beta_4}{p} - c_{34}\right) \left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \\
&+ c_{35} \left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 + c_{36}K_g + c_{37}\frac{1}{\tau_I} + c_{38}D \\
&- \begin{bmatrix} \|\widetilde{\boldsymbol{\theta}}_{1,i}[k]\| \\ \|\boldsymbol{\theta}_{1,i}[k]\| \end{bmatrix}^{\top} \begin{bmatrix} c_{15} & -\frac{1}{2}pK_g \\ -\frac{1}{2}pK_g & p(K_g - K_g^* - {}^{1}/_{2a_{13}}) \end{bmatrix} \begin{bmatrix} \|\widetilde{\boldsymbol{\theta}}_{1,i}[k]\| \\ \|\boldsymbol{\theta}_{1,i}[k]\| \end{bmatrix} \\
&+ \frac{pa_{13}}{2} \left\|\boldsymbol{d}_i[k]\right\|^2 .
\end{aligned} \tag{F.41}$$

By defining $\boldsymbol{v}_{\theta,i}$ and $\boldsymbol{\Lambda}_\theta$ by

$$\boldsymbol{v}_{\theta,i} = \begin{bmatrix} \|\widetilde{\boldsymbol{\theta}}_{1,i}\| \\ \|\boldsymbol{\theta}_{1,i}\| \end{bmatrix} \qquad \boldsymbol{\Lambda}_\theta = \begin{bmatrix} c_{15} & -\frac{1}{2}pK_g \\ -\frac{1}{2}pK_g & p(K_g - K_g^* - {}^{1}/_{2a_{13}}) \end{bmatrix},$$

we can write the difference of the stabilization Lyapunov function as

$$\begin{aligned}\Delta V_{\text{stab}}[k+1] = \sum_{i=1}^{p} &-c_{15}\left\|\widetilde{\boldsymbol{\theta}}_{0,i}[k]\right\|^2 - c_{20}\left\|\widetilde{\eta}_i[k]\right\|^2 - c_{39}\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \\ &+ c_{35}\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 + c_{36}K_g + c_{37}\frac{1}{\tau_I} + c_{38}D \\ &- \boldsymbol{v}_{\theta,i}^{\top}[k]\boldsymbol{\Lambda}_\theta \boldsymbol{v}_{\theta,i}[k] + \frac{pa_{13}}{2}\left\|\boldsymbol{d}_i[k]\right\|^2\end{aligned} \tag{F.42}$$

where $c_{39} = \frac{\beta_4}{p} - c_{34}$. For stability, we need $c_{39} > 0$. Unfortunately, we cannot force c_{39} to be positive by choosing sufficiently large tuning parameters as β_4 is a property of the system and not of the controller. However, from the definitions of the constants, we see that

$$\begin{aligned}c_{34} &= \mathcal{O}\left(c_6\right) = \mathcal{O}\left(c_1 + L_G\right) = \mathcal{O}\left(\mathcal{O}\left(L_f + L_G\right) + L_G\right) \\ &= \mathcal{O}\left(L_f\right) + \mathcal{O}\left(L_G\right) = \mathcal{O}\left(\Delta t\right).\end{aligned}$$

Therefore, for a sufficiently small time step, $c_{39} > 0$. Also note that a smaller time step is needed when there are more agents as $\frac{\beta_4}{p}$ decreases with the number of agents.

Since $\boldsymbol{v}_{\theta,i}$ depends on parameter estimates, we need $-\boldsymbol{v}_{\theta,i}^{\top}\boldsymbol{\Lambda}_\theta \boldsymbol{v}_{\theta,i}$ to be negative definite for parameter estimates to still converge. This term is negative definite if and only if $\boldsymbol{\Lambda}_\theta$ is positive definite. We can check that a matrix is positive definite by checking that the determinants of all of its principal submatrices are positive. Since $\boldsymbol{\Lambda}_\theta$ is a 2×2 matrix, we simply need to check that $\det(\boldsymbol{\Lambda}_\theta) > 0$ and $c_{15} > 0$. Since we had previously shown that α can always be chosen such that $c_{15} > 0$, it remains to show that $\det(\boldsymbol{\Lambda}_\theta) > 0$. The determinant is

$$\det(\boldsymbol{\Lambda}_\theta) = c_{15} p \left(K_g - K_g^* - \frac{1}{2a_{13}} \right) - \frac{1}{4} p^2 K_g^2. \tag{F.43}$$

This term is positive if

$$c_{15} > \frac{pK_g^2}{4\left(K_g - K_g^* - \frac{1}{2a_{13}}\right)}.$$

Note that since a_{13} is arbitrary, we can choose it to be large so that it does not affect the right hand expression much. Suppose the system is stable. Then $K_g^* = 0$ so for a fixed c_{15}, we can always choose $K_g > 0$ to be small enough so that $c_{15} > \frac{1}{4} pK_g$. Now suppose that the system is not stable. Then we must choose $K_g > K_g^*$ which limits our ability to ensure stability by choosing a small enough K_g. Alternatively, we can ensure stability by choosing c_{15} to be sufficiently large. Recall that

$$c_{15} = \left(1 - \alpha - \frac{\alpha}{a_1}\right) \min\{\sigma_0, \gamma_w^-\} \alpha^{T+1}.$$

By using a large amplitude dither signal, γ_w^- can be made arbitrarily large. Then by choosing $\sigma_0 \geq \gamma_w^-$, we can make c_{15} arbitrarily large and ensure that $\boldsymbol{\Lambda}_\theta$ is positive definite. Let γ_Λ^- be the minimum eigenvalue of $\boldsymbol{\Lambda}_\theta$. Then we can write the stabilization Lyapunov difference as

$$\begin{aligned} \Delta V_{\text{stab}}[k+1] = \sum_{i=1}^{p} & -c_{15} \left\| \widetilde{\boldsymbol{\theta}}_{0,i}[k] \right\|^2 - c_{20} \left\| \widetilde{\eta}_i[k] \right\|^2 - c_{39} \left\| \widetilde{\boldsymbol{x}}[k] \right\|^2 \\ & - \gamma_\Lambda^- \left\| \widetilde{\boldsymbol{\theta}}_{1,i}[k] \right\|^2 - \gamma_\Lambda^- \left\| \boldsymbol{\theta}_{1,i}[k] \right\|^2 + c_{35} \left\| \widetilde{\boldsymbol{u}}[k] \right\|^2 \\ & + c_{36} K_g + c_{37} \frac{1}{\tau_I} + c_{38} D + \frac{p a_{13}}{2} \left\| \boldsymbol{d}_i[k] \right\|^2. \end{aligned} \tag{F.44}$$

Then noting that $\|\boldsymbol{d}_i[k]\|^2 \leq D$, we have

$$\begin{aligned}
\Delta V_{\text{stab}}[k+1] \leq \; & \sum_{i=1}^{p} -c_{15} \left\|\widetilde{\boldsymbol{\theta}}_{0,i}[k]\right\|^2 - c_{20} \left\|\widetilde{\eta}_i[k]\right\|^2 - c_{39} \left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \\
& - \gamma_\Lambda^- \left\|\widetilde{\boldsymbol{\theta}}_{1,i}[k]\right\|^2 - \gamma_\Lambda^- \left\|\boldsymbol{\theta}_{1,i}[k]\right\|^2 + c_{35} \left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 \\
& + c_{36} K_g + c_{37} \frac{1}{\tau_I} + c_{38} D + \frac{p a_{13}}{2} D D \\
\leq \; & \sum_{i=1}^{p} -c_{15} \left\|\widetilde{\boldsymbol{\theta}}_{0,i}[k]\right\|^2 - c_{20} \left\|\widetilde{\eta}_i[k]\right\|^2 - c_{39} \left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \\
& - \gamma_\Lambda^- \left\|\widetilde{\boldsymbol{\theta}}_{1,i}[k]\right\|^2 - \gamma_\Lambda^- \left\|\boldsymbol{\theta}_{1,i}[k]\right\|^2 + c_{35} \left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 \\
& + c_{36} K_g + c_{37} \frac{1}{\tau_I} + \left(c_{38} + \frac{p a_{13}}{2} D\right) D \\
\leq \; & \sum_{i=1}^{p} -c_{15} \left\|\widetilde{\boldsymbol{\theta}}_{0,i}[k]\right\|^2 - c_{20} \left\|\widetilde{\eta}_i[k]\right\|^2 - c_{39} \left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 \\
& - \gamma_\Lambda^- \left\|\widetilde{\boldsymbol{\theta}}_{1,i}[k]\right\|^2 - \gamma_\Lambda^- \left\|\boldsymbol{\theta}_{1,i}[k]\right\|^2 + c_{35} \left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 \\
& + c_{36} K_g + c_{37} \frac{1}{\tau_I} + c_{40} D
\end{aligned} \tag{F.45}$$

where $c_{40} = c_{38} + \frac{p a_{13}}{2} D$ is a positive constant. This Lyapunov function says that if $\widetilde{\boldsymbol{u}}$ converges, then the parameter estimates converge and the system is stabilized to a neighbourhood of the steady-state manifold. Again, the sizes of the neighbourhoods are $\mathcal{O}(K_g + \frac{1}{\tau_I} + D)$.

The final part of the proof is to show that the extremum-seeking controller's input bias, $\widehat{\boldsymbol{u}}$, converges to the optimal input, $\boldsymbol{u}^*$. Consider the Lyapunov function

$$V_u[k] = \sum_{i=1}^{p} \widetilde{\boldsymbol{u}}_i^\top[k] \widetilde{\boldsymbol{u}}_i[k]. \tag{F.46}$$

Since $\widetilde{\boldsymbol{u}}_i[k] = \boldsymbol{u}_i^* - \boldsymbol{u}_i[k]$, by using the dynamics of $\boldsymbol{u}_i$ from (4.24), the dynamics for

$\widetilde{\boldsymbol{u}}_i$ are $\widetilde{\boldsymbol{u}}_i[k+1] = \widetilde{\boldsymbol{u}}_i[k] + \frac{1}{\tau_I}\widehat{\boldsymbol{\theta}}_{1,i}[k]$. Using these dynamics, the Lyapunov difference is

$$\begin{aligned}
\Delta V_u[k+1] &= V_u[k+1] - V_u[k] \\
&= \sum_{i=1}^{p} \widetilde{\boldsymbol{u}}_i^\top[k+1]\widetilde{\boldsymbol{u}}_i[k+1] - \sum_{i=1}^{p} \widetilde{\boldsymbol{u}}_i^\top[k]\widetilde{\boldsymbol{u}}_i[k] \\
&= \sum_{i=1}^{p} \left(\widetilde{\boldsymbol{u}}_i^\top[k] + \frac{1}{\tau_I}\widehat{\boldsymbol{\theta}}_{1,i}^\top[k]\right)\left(\widetilde{\boldsymbol{u}}_i[k] + \frac{1}{\tau_I}\widehat{\boldsymbol{\theta}}_{1,i}[k]\right) - \widetilde{\boldsymbol{u}}_i^\top[k]\widetilde{\boldsymbol{u}}_i[k] \\
&= \sum_{i=1}^{p} \widetilde{\boldsymbol{u}}_i^\top[k]\widetilde{\boldsymbol{u}}_i[k] + \frac{2}{\tau_I}\widehat{\boldsymbol{\theta}}_{1,i}^\top[k]\widetilde{\boldsymbol{u}}_i[k] + \frac{1}{\tau_I^2}\widehat{\boldsymbol{\theta}}_{1,i}^\top[k]\widehat{\boldsymbol{\theta}}_{1,i}[k] - \widetilde{\boldsymbol{u}}_i^\top[k]\widetilde{\boldsymbol{u}}_i[k] \\
&= \sum_{i=1}^{p} \frac{2}{\tau_I}\widehat{\boldsymbol{\theta}}_{1,i}^\top[k]\widetilde{\boldsymbol{u}}_i[k] + \frac{1}{\tau_I^2}\widehat{\boldsymbol{\theta}}_{1,i}^\top[k]\widehat{\boldsymbol{\theta}}_{1,i}[k].
\end{aligned} \tag{F.47}$$

Then using the definition of $\widetilde{\boldsymbol{\theta}}_{1,i}$, we have that $\widehat{\boldsymbol{\theta}}_{1,i}[k] = \boldsymbol{\theta}_{1,i}[k] - \widetilde{\boldsymbol{\theta}}_{1,i}[k]$. Substituting this expression gives

$$\begin{aligned}
\Delta V_u[k+1] = \sum_{i=1}^{p} &\frac{2}{\tau_I}\left(\boldsymbol{\theta}_{1,i}^\top[k] - \widetilde{\boldsymbol{\theta}}_{1,i}^\top[k]\right)\widetilde{\boldsymbol{u}}_i[k] \\
&+ \frac{1}{\tau_I^2}\left(\boldsymbol{\theta}_{1,i}^\top[k] - \widetilde{\boldsymbol{\theta}}_{1,i}^\top[k]\right)\left(\boldsymbol{\theta}_{1,i}[k] - \widetilde{\boldsymbol{\theta}}_{1,i}[k]\right).
\end{aligned} \tag{F.48}$$

Next we substitute (4.6), the definition of $\boldsymbol{\theta}_{1,i}$, into the first term and expand both terms:

$$\begin{aligned}
\Delta V_u[k+1] &= \sum_{i=1}^{p} \frac{2}{\tau_I}\left(\frac{1}{p}\nabla H\left(\overline{\boldsymbol{x}}\right)\boldsymbol{G}_i\left(\boldsymbol{x}[k]\right)\right)\widetilde{\boldsymbol{u}}_i[k] - \frac{2}{\tau_I}\widetilde{\boldsymbol{\theta}}_{1,i}^\top[k]\widetilde{\boldsymbol{u}}_i[k] \\
&\quad + \frac{1}{\tau_I^2}\boldsymbol{\theta}_{1,i}^\top[k]\boldsymbol{\theta}_{1,i}[k] - \frac{2}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^\top[k]\boldsymbol{\theta}_{1,i}[k] + \frac{1}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^\top[k]\widetilde{\boldsymbol{\theta}}_{1,i}[k] \\
&= \frac{2}{p\tau_I}\nabla H\left(\overline{\boldsymbol{x}}\right)\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\widetilde{\boldsymbol{u}}[k] + \sum_{i=1}^{p} -\frac{2}{\tau_I}\widetilde{\boldsymbol{\theta}}_{1,i}^\top[k]\widetilde{\boldsymbol{u}}_i[k] \\
&\quad + \frac{1}{\tau_I^2}\boldsymbol{\theta}_{1,i}^\top[k]\boldsymbol{\theta}_{1,i}[k] - \frac{2}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^\top[k]\boldsymbol{\theta}_{1,i}[k] + \frac{1}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^\top[k]\widetilde{\boldsymbol{\theta}}_{1,i}[k].
\end{aligned} \tag{F.49}$$

We would like to be able to simplify this expression using the convexity of the steady-state cost from Assumption 4.1. Since the steady-state gradient is $\frac{\partial \ell}{\partial \boldsymbol{u}}\big|_{\widehat{\boldsymbol{u}}[k]} = \nabla H(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k]))\boldsymbol{G}(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}([k]))$, we add and subtract $\frac{2}{p\tau_I}\nabla H(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k]))\boldsymbol{G}(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}([k]))\widetilde{\boldsymbol{u}}[k] = \frac{2}{p\tau_I}\frac{\partial \ell}{\partial \boldsymbol{u}}\big|_{\widehat{\boldsymbol{u}}[k]}\widetilde{\boldsymbol{u}}[k]$ from the Lyapunov difference:

$$\begin{aligned}
\Delta V_u[k+1] &= \frac{2}{p\tau_I}\nabla H\left(\overline{\boldsymbol{x}}\right)\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\widetilde{\boldsymbol{u}}[k] - \frac{2}{p\tau_I}\nabla H\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}([k])\right)\widetilde{\boldsymbol{u}}[k] \\
&\quad + \frac{2}{p\tau_I}\nabla H\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}([k])\right)\widetilde{\boldsymbol{u}}[k] + \sum_{i=1}^{p} -\frac{2}{\tau_I}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\widetilde{\boldsymbol{u}}_i[k] \\
&\quad + \frac{1}{\tau_I^2}\boldsymbol{\theta}_{1,i}^{\top}[k]\boldsymbol{\theta}_{1,i}[k] - \frac{2}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\boldsymbol{\theta}_{1,i}[k] + \frac{1}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\widetilde{\boldsymbol{\theta}}_{1,i}[k] \\
&= \frac{2}{p\tau_I}\nabla H\left(\overline{\boldsymbol{x}}\right)\boldsymbol{G}\left(\boldsymbol{x}[k]\right)\widetilde{\boldsymbol{u}}[k] - \frac{2}{p\tau_I}\nabla H\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])\right)\boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}([k])\right)\widetilde{\boldsymbol{u}}[k] \\
&\quad + \frac{2}{p\tau_I}\frac{\partial \ell}{\partial \boldsymbol{u}}\bigg|_{\widehat{\boldsymbol{u}}[k]}\widetilde{\boldsymbol{u}}[k] + \sum_{i=1}^{p} -\frac{2}{\tau_I}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\widetilde{\boldsymbol{u}}_i[k] + \frac{1}{\tau_I^2}\boldsymbol{\theta}_{1,i}^{\top}[k]\boldsymbol{\theta}_{1,i}[k] \\
&\quad - \frac{2}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\boldsymbol{\theta}_{1,i}[k] + \frac{1}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\widetilde{\boldsymbol{\theta}}_{1,i}[k].
\end{aligned} \tag{F.50}$$

By the Lipschitzness of H from Assumption 4.3, both $\nabla H(\overline{\boldsymbol{x}}) \leq L_H$ and $\nabla H(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}[k])) \leq L_H$, so we can combine the first two terms to obtain

$$\begin{aligned}
\Delta V_u[k+1] &\leq \frac{2L_H}{p\tau_I}\left\|\boldsymbol{G}\left(\boldsymbol{x}[k]\right) - \boldsymbol{G}\left(\boldsymbol{\pi}(\widehat{\boldsymbol{u}}([k])\right)\right\|\left\|\widetilde{\boldsymbol{u}}[k]\right\| + \frac{2}{p\tau_I}\frac{\partial \ell}{\partial \boldsymbol{u}}\bigg|_{\widehat{\boldsymbol{u}}[k]}\widetilde{\boldsymbol{u}}[k] \\
&\quad + \sum_{i=1}^{p} -\frac{2}{\tau_I}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\widetilde{\boldsymbol{u}}_i[k] + \frac{1}{\tau_I^2}\boldsymbol{\theta}_{1,i}^{\top}[k]\boldsymbol{\theta}_{1,i}[k] - \frac{2}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\boldsymbol{\theta}_{1,i}[k] \\
&\quad + \frac{1}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\widetilde{\boldsymbol{\theta}}_{1,i}[k].
\end{aligned} \tag{F.51}$$

Then we can use the Lipschitzness of $\boldsymbol{G}$ from Assumption 4.3 to further simplify the first term:

$$
\begin{aligned}
\Delta V_u[k+1] \leq{}& \frac{2L_H L_G}{p\tau_I}\left\|\boldsymbol{x}[k]-\boldsymbol{\pi}(\widehat{\boldsymbol{u}}([k]))\right\|\left\|\widetilde{\boldsymbol{u}}[k]\right\| + \frac{2}{p\tau_I}\frac{\partial \ell}{\partial \boldsymbol{u}}\bigg|_{\widehat{\boldsymbol{u}}[k]}\widetilde{\boldsymbol{u}}[k] \\
&+\sum_{i=1}^{p} -\frac{2}{\tau_I}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\widetilde{\boldsymbol{u}}_i[k] + \frac{1}{\tau_I^2}\boldsymbol{\theta}_{1,i}^{\top}[k]\boldsymbol{\theta}_{1,i}[k] - \frac{2}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\boldsymbol{\theta}_{1,i}[k] \\
&+\frac{1}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\widetilde{\boldsymbol{\theta}}_{1,i}[k] \\
={}& \frac{2L_H L_G}{p\tau_I}\left\|\widetilde{\boldsymbol{x}}[k]\right\|\left\|\widetilde{\boldsymbol{u}}[k]\right\| + \frac{2}{p\tau_I}\frac{\partial \ell}{\partial \boldsymbol{u}}\bigg|_{\widehat{\boldsymbol{u}}[k]}\widetilde{\boldsymbol{u}}[k] + \sum_{i=1}^{p} -\frac{2}{\tau_I}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\widetilde{\boldsymbol{u}}_i[k] \\
&+\frac{1}{\tau_I^2}\boldsymbol{\theta}_{1,i}^{\top}[k]\boldsymbol{\theta}_{1,i}[k] - \frac{2}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\boldsymbol{\theta}_{1,i}[k] + \frac{1}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\widetilde{\boldsymbol{\theta}}_{1,i}[k].
\end{aligned}
\tag{F.52}
$$

We can now use Assumption 4.1, which says that the steady-state total cost is convex, to bound this expression as

$$
\begin{aligned}
\Delta V_u[k+1] \leq{}& \sum_{i=1}^{p}\frac{2L_H L_G}{p^2\tau_I}\left\|\widetilde{\boldsymbol{x}}[k]\right\|\left\|\widetilde{\boldsymbol{u}}[k]\right\| - \frac{2}{p^2\tau_I}\beta_4\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 - \frac{2}{\tau_I}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\widetilde{\boldsymbol{u}}_i[k] \\
&+\frac{1}{\tau_I^2}\boldsymbol{\theta}_{1,i}^{\top}[k]\boldsymbol{\theta}_{1,i}[k] - \frac{2}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\boldsymbol{\theta}_{1,i}[k] + \frac{1}{\tau_I^2}\widetilde{\boldsymbol{\theta}}_{1,i}^{\top}[k]\widetilde{\boldsymbol{\theta}}_{1,i}[k].
\end{aligned}
\tag{F.53}
$$

Next, we remove the indefinite terms by completing the squares. For any arbitrary $a_{14},\ldots,a_{16}\in\mathbb{R}_{>0}$, we have

$$
\begin{aligned}
\Delta V_u[k+1] \leq{}& \sum_{i=1}^{p}\frac{a_{14}L_H L_G}{p^2\tau_I}\left\|\widetilde{\boldsymbol{x}}[k]\right\|^2 + \frac{L_H L_G}{a_{14}p^2\tau_I}\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 - \frac{2\beta_1}{p^2\tau_I}\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 \\
&+\frac{a_{15}}{\tau_I}\left\|\widetilde{\boldsymbol{\theta}}_{1,i}[k]\right\|^2 + \frac{1}{a_{15}\tau_I}\left\|\widetilde{\boldsymbol{u}}[k]\right\|^2 + \frac{1}{\tau_I^2}\left\|\boldsymbol{\theta}_{1,i}[k]\right\|^2 \\
&+\frac{a_{16}}{\tau_I^2}\left\|\widetilde{\boldsymbol{\theta}}_{1,i}[k]\right\|^2 + \frac{1}{a_{16}\tau_I^2}\left\|\boldsymbol{\theta}_{1,i}[k]\right\|^2 + \frac{1}{\tau_I^2}\left\|\widetilde{\boldsymbol{\theta}}_{1,i}[k]\right\|^2.
\end{aligned}
\tag{F.54}
$$

Finally, we collect terms to simplify the Lyapunov difference.

$$\begin{aligned}\Delta V_u[k+1] &= \sum_{i=1}^{p} \frac{a_{14} L_H L_G}{p^2 \tau_I} \left\| \widetilde{\boldsymbol{x}}[k] \right\|^2 - \left(\frac{2\beta_1}{p^2 \tau_I} - \frac{L_H L_G}{a_{14} p^2 \tau_I} - \frac{1}{a_{15} \tau_I} \right) \left\| \widetilde{\boldsymbol{u}}[k] \right\|^2 \\ &\quad + \left(\frac{a_{15}}{\tau_I} + \frac{a_{16}}{\tau_I^2} + \frac{1}{\tau_I^2} \right) \left\| \widetilde{\boldsymbol{\theta}}_{1,i}[k] \right\|^2 + \left(\frac{1}{\tau_I^2} + \frac{1}{a_{16} \tau_I^2} \right) \left\| \boldsymbol{\theta}_{1,i}[k] \right\|^2 \\ &= \sum_{i=1}^{p} -c_{41} \left\| \widetilde{\boldsymbol{u}}[k] \right\|^2 + c_{42} \left\| \widetilde{\boldsymbol{x}}[k] \right\|^2 + c_{43} \left\| \widetilde{\boldsymbol{\theta}}_{1,i}[k] \right\|^2 + c_{44} \left\| \boldsymbol{\theta}_{1,i}[k] \right\|^2 \end{aligned} \quad \text{(F.55)}$$

where $c_{41} = \frac{2\beta_1}{p^2 \tau_I} - \frac{L_H L_G}{a_{14} p^2 \tau_I} - \frac{1}{a_{15} \tau_I}$, $c_{42} = \frac{a_{14} L_H L_G}{p^2 \tau_I}$, $c_{43} = \frac{a_{15}}{\tau_I} + \frac{a_{16}}{\tau_I^2} + \frac{1}{\tau_I^2}$, and $c_{44} = \frac{1}{\tau_I^2} + \frac{1}{a_{16} \tau_I^2}$. Since a_{14} and a_{15} are constants obtained by completing the squares, we can always choose them such that $c_{41} > 0$. Therefore this Lyapunov function can be used to show that $\widehat{\boldsymbol{u}}$ converges to a neighbourhood of $\boldsymbol{u}^*$. However, choosing a_{14} very large or a_{15} very small can result in large values of c_{42} and c_{43} which results in this neighbourhood being large.

Now that we have a candidate Lyapunov function for the extremum-seeking aspect of the controller, we add it V_{stab} to get the final, overall Lyapunov function

$$V[k] = V_{\text{Stab}}[k] + V_u[k]. \quad \text{(F.56)}$$

The first difference of this Lyapunov function is

$$\begin{aligned}\Delta V[k+1] \leq \sum_{i=1}^{p} &-c_{15} \left\| \widetilde{\boldsymbol{\theta}}_{0,i}[k] \right\|^2 - c_{20} \left\| \widetilde{\eta}_i[k] \right\|^2 - (c_{39} - c_{42}) \left\| \widetilde{\boldsymbol{x}}[k] \right\|^2 \\ &- \left(\gamma_\Lambda^- - c_{43} \right) \left\| \widetilde{\boldsymbol{\theta}}_{1,i}[k] \right\|^2 - \left(\gamma_\Lambda^- - c_{44} \right) \left\| \boldsymbol{\theta}_{1,i}[k] \right\|^2 \\ &- (c_{41} - c_{35}) \left\| \widetilde{\boldsymbol{u}}[k] \right\|^2 + c_{36} K_g + c_{37} \frac{1}{\tau_I} + c_{40} D. \end{aligned} \quad \text{(F.57)}$$

We would like to tune the controller in such a way that all six quadratic terms are negative definite. At this stage, our main tuning parameter to manipulate is τ_I. To ensure that $c_{39}-c_{42} > 0$, $\gamma_\Lambda^- - c_{43} > 0$, and $\gamma_\Lambda^- - c_{44} > 0$, we should try to minimize c_{42}, c_{43}, and c_{44} which can be done by making τ_I large. However, to make $c_{41} - c_{35} > 0$, we need to make c_{41} large which can be done by choosing τ_I to be small. We should therefore choose τ_I to be small enough that $c_{41} > c_{35}$ but still large enough that the other terms are positive. Alternatively, by looking at the parameter definitions, we can see that

$$\begin{aligned} c_{35} &= \mathcal{O}\left(c_{23} + c_{24} + c_{28} + c_{29}\right) = \mathcal{O}\left(c_7 + c_{12}\right) \\ &= \mathcal{O}\left(c_2 + L_G\right) = \mathcal{O}\left(L_G\right) = \mathcal{O}\left(\Delta t\right). \end{aligned}$$

Therefore by choosing a small enough time step, it is possible to make c_{35} small enough that τ_I can be chosen to ensure that $c_{41} - c_{35} > 0$ while still being large enough that all the other terms are indeed negative definite.

Suppose that Δt and τ_I are chosen such that the first six terms are negative definite. Then since the last four terms are bounded, we can conclude that $\widetilde{\boldsymbol{u}}$, $\widetilde{\boldsymbol{x}}$, and $\widetilde{\boldsymbol{\theta}}$ all converge to a neighbourhood of the origin. The size of this neighbourhood is $\mathcal{O}(K_g + \frac{1}{\tau_I} + D)$.