Dual iterative linear quadratic Gaussian control for uncertain nonlinear systems

by

Andrew Craig Mathis

Master of Science in Engineering, UNB, 2018
Bachelor of Science in Engineering, UNB, 2016

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

In the Graduate Academic Unit of Mechanical Engineering

Supervisors:	Jon Sensinger, Ph.D., Electrical & Computer Engineering
	Juan Antonio Carretero, Ph.D., Mechanical Engineering
Examining Board:	Andrew Gerber, Ph.D., Mechanical Engineering
	Gobinda Saha, Ph.D., Mechanical Engineering
	Chris Diduch, Ph.D., Electrical & Computer Engineering
External Examiner:	Ali Mesbah, Ph.D., College of Chemistry, University of California, Berkeley

This dissertation is submitted to the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

June, 2024

Abstract

Dual iterative linear quadratic Gaussian (iLQG) is an approximately optimal control method that implicitly balances the often-opposing control goals of cost function minimization and system identification for uncertain dynamic systems. The balancing of these two control goals is known as dual control, and existing dual strategies fall into two classes: explicit and implicit approximations. The explicit strategy relies on a pre-determined trade-off between cost function minimization and system identification, whereas the implicit strategy leaves the determination of this trade-off to the controller. Existing implicit dual strategies are limited by only considering discrete realizations of control inputs and/or parameters or are only applicable for problems with a limited number of states. Dual iLQG on the other hand is based on continuous state and parameter space with the use of derivatives of a linearized system about a control trajectory. This approach avoids the curse of dimensionality, allowing the algorithm being able to solve larger problems and to solve problems more efficiently than existing dual control approaches. Compared to other learning algorithms based on neural networks, dual iLQG has the advantage that it can be implemented on a system during normal operation, without needing an initial training period. Dual iLQG is applicable to many present and future high-impact system and control applications.

Preface

In January 2023, this document was in its second draft when I found a reference to a body of work by Bar-Shalom and Tse from the 1970s [68] and realized that it overlapped with my research. In this preface, the reason I missed this work in my literature review and the differences between my work and Bar-Shalom's will be discussed.

In 1966, Mayne published an optimal control algorithm for deterministic nonlinear systems known as differential dynamic programming [45]. In 2004, Li and Todorov published an approximation of differential dynamic programming that was similar to the linear quadratic regulator but for nonlinear systems, and named it the iterative linear quadratic regulator (iLQR) [36]. In 2006, Li and Todorov extended iLQR to stochastic systems, creating the iterative Linear Quadratic Gaussian (iLQG) [37]. My research has focused on extending iLQG to be dual, which appeared to have not been considered in the optimal feedback control community.

In 1960, Feldbaum introduced dual control theory, but showed that optimal dual control was intractable, resulting in subsequent research into approximate dual control methods [19]. Between 1973 and 1976, Bar-Shalom and Tse derived a dual control algorithm for nonlinear stochastic systems known as wide-sense dual control [4, 5, 68]. This algorithm is functionally the same as differential dynamic programming for systems with additive noise, but used a partial certainty equivalence approximation where the stochastic nature of the uncertain states and parameters are handled by treating their estimates as if they were the true values.

Bar-Shalom and Tse do not appear to have known of Mayne's similar work on differential dynamic programming at the time, and likewise, Todorov does not appear to have known of Bar-Shalom's similar work on wide-sense dual control. These two lines of research seem to have evolved independently; Mayne and Todorov in the optimal feedback control field, and Bar-Shalom and Tse in the dual control field. These papers are further compared in Table 1. Although my literature search was based in both of these fields, since iLQR and iLQG were developed in the 2000s, when I searched for anyone else who had thought to work on dual iLQG, I had not

considered that a dual version of iLQR may have been developed independently decades before Todorov's work.

The type of noise considered is a critical factor for differentiating these works. Bar-Shalom and Tse considered nonlinear systems with additive noise and ended up with terms in their quadratic approximation of the cost-to-go function that were based on the noise and state covariances. In 2010, Theodorou, Tassa, and Todorov [65] showed that in stochastic differential dynamic programming additive noise has no impact on the control policy, and that these noise-based terms only persist for systems with state and/or control multiplicative noise. Since the noise terms in the quadratic approximation of the cost-to-go function equate to zero for Bar-Shalom and Tse's case of additive noise, the additional terms in Bar-Shalom and Tse's wide-sense dual control represent an explicit consideration of the features of dual control, and is therefore an explicit dual technique.

Comparing my work on dual iLQG with Bar-Shalom and Tse's wide-sense dual controller, my work is a more general algorithm that can handle more than just additive noise and does so in such a way that it is implicitly dual instead of explicitly dual. Having partially observable states and multiplicative noise means that a filter that can handle multiplicative noise must be incorporated into the control algorithm, creating terms in the control policy that are not required in the additive noise case presented by Bar-Shalom and Tse. Many applications involve multiplicative noise and dual iLQG would allow for dual control to be applied to these fields. These applications include the control of prostheses using myoelectric signals [57], robot motion planning when using sensors with distance-dependent errors [15], stochastic fluid dynamics [13], and power grids where renewable energy represents a significant portion of total power generation [20]. Since no terms have been added into dual iLQG to create the dual features of probing, caution, and selectiveness, dual iLQG is an implicit dual method, which are generally able to better manage the dual trade-off between objective function minimization and system identification.

Bar-Shalom and Tse's work was extended by Klenske in [29], using a parametric, Gaussian process and neural network regression to estimate the system dynamics. Klenske also uses a different linearization method, augmenting the state vector with the state-dependent nonlinear terms. Finally, like Bar-Shalom's work on which Klenske's is based, he only considers additive noise, and this approach cannot handle multiplicative noise.

The innovation of this present work is the extension of iLQG to create an implicit dual method that overcomes Bellman's "curse of dimensionality" that currently limits conventional

implicit dual control algorithms. iLQG was made dual by treating the uncertain parameters as states and creating an augmented state vector, state covariance matrix, and system model. Additionally, an adaptive form of iLQG was created, and the robustness of both dual and adaptive iLQG to time-varying parameter dynamics and unmodeled dynamics were investigated.

The significance of this present work is that a class of control problem can now be solved in a quick and robust way. Dual iLQG represents an approximately optimal control solution for nonlinear systems with uncertain parameters and it can handle large systems and those with multiplicative noise. Having the ability to cautiously and selectively probe a system to learn its parameters online puts dual iLQG in a unique position to handle present and future control challenges. In this thesis, dual iLQG is applied to the control of COVID-19 and the production of renewable energy from an anaerobic digester, but it could be applied to a wide spectrum of problems including those in the healthcare, financial, or mechatronic fields.

In September 2023, the University of New Brunswick's Office of Research Services (ORS) agreed that this work was significant enough to pursue patenting the dual iLQG approach. ORS chose to employ a patent office with expertise in fields related to this thesis, which completed a comprehensive independent audit on the novelty of this idea in the literature. The patent agent only identified a single set of similar works - the same Tse and Bar-Shalom articles mentioned above - and after further analysis agreed that this present work was innovative with respect to these articles and all of the other work in this field.

Year	Author(s)	Controller	States Filtered?	State Noise	Dual	Dynamics approx. in cost fcn.
1966	Mayne	DDP	No	None	No	2nd order
1976	Tse, Bar-Shalom	Wide-sense dual	Yes	Additive	Explicit	2nd order
2004	Li, Todorov	iLQR	No	None	No	1st order
2006	Li, Todorov	iLQG	Yes	Multiplicative	No	1st order
2010	Theodorou, Tassa, Todorov	SDDP	No	Multiplicative	No	2nd order
Present	Mathis, Carretero, Sensinger	Dual iLQG	Yes	Multiplicative	Implicit	1st order

Table 1: Comparison of DDP-like controllers

Acknowledgements

I would like to acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and the New Brunswick Innovation Foundation (NBIF).

I would also like to thank Sakthi Thangavel for providing his dual MS-SP-NMPC code to assist in this work.

Table of Contents

Abstract ii

Preface iii

Acknowledgments vii

Table of Contents viii

List of Tables xi

List of Figures xii

Abbreviations xvi

List of Variables xvii

1 Introduction 1

1.1 Motivation and gap . . . 1

1.2 Organization of the thesis . . . 4

2 Background 5

2.1 Optimal control . . . 5

2.1.1 The linear quadratic regulator (LQR) . . . 6

2.1.2 Iterative LQR (iLQR) . . . 7

2.1.3 Iterative Linear Quadratic Gaussian (iLQG) . . . 9

2.1.3.1 The forward pass . . . 10

2.1.3.2 The estimator . . . 13

2.1.3.3 The backward pass . . . 15

2.1.3.4 The line search . . . 19

2.1.3.5 Stopping criteria . . . 20

2.1.3.6 Moving horizon implementation . . . 20
2.2 Adaptive control . . . 21
2.3 Dual control . . . 22
2.3.1 Explicit dual control . . . 23
2.3.2 Implicit dual control . . . 25
2.4 Summary . . . 28

3 Dual iLQG **29**
3.1 Motivation . . . 29
3.2 Adaptive iLQG . . . 29
3.3 Dual iLQG . . . 30
3.3.1 Seeding approach . . . 33
3.3.2 Moving horizon approaches . . . 33
3.4 Comparison with dual MS-SP-NMPC . . . 34
3.5 Robustness to time-varying parameters . . . 37
3.6 Importance of compensating for multiplicative noise . . . 41
3.7 Robustness to unmodelled dynamics . . . 47
3.8 Summary . . . 58

4 Application to anaerobic digestion **59**
4.1 Motivation . . . 59
4.2 System model . . . 60
4.2.1 The AM2 model . . . 60
4.2.2 AM2 with uncertain feedstocks . . . 61
4.3 Results . . . 62
4.3.1 Two-parameter comparison with dual MS-SP-NMPC . . . 62
4.3.2 Seventeen-parameter comparison with adaptive iLQG . . . 66
4.4 Summary . . . 71

5 Application to the control of the COVID-19 outbreak **72**
5.1 Motivation . . . 72
5.2 System model . . . 73
5.2.1 SIDARTHE model limitations . . . 77
5.2.2 Changes to the SIDARTHE model . . . 77

5.3 Results . . . 78
5.3.1 Two-parameter comparison with dual MS-SP-NMPC . . . 78
5.3.2 Sixteen-parameter comparison with adaptive iLQG . . . 84
5.4 Summary . . . 89

6 Conclusions **90**
6.1 Key contributions . . . 91
6.2 Potential applications . . . 92
6.3 Limitations . . . 93
6.4 Future directions . . . 93

Appendix A Detailed comparison of DDP-like controllers **95**

Bibliography **108**

Vita

List of Tables

1 Comparison of DDP-like controllers vi

4.1 Parameters for the AM2 model 62

4.2 Parameter values for the AM2 comparison with seventeen uncertainties 67

5.1 Parameters for the SIDARTHE model 75

5.2 Results from 100 seeded runs of the 2 parameter SIDARTHE COVID-19 model with several rolling horizon lengths 83

5.3 Parameter values **d** for the SIDARTHE comparison with sixteen uncertainties 86

A.1 Detailed comparison of DDP-like controllers: titles 96

A.2 Detailed comparison of DDP-like controllers: algorithms 97

A.3 Detailed comparison of DDP-like controllers: noise 98

A.4 Detailed comparison of DDP-like controllers: approximations 99

A.5 Detailed comparison of DDP-like controllers: regularization 100

A.6 Detailed comparison of DDP-like controllers: constraints 101

List of Figures

2.1 Closed-loop iLQG system diagram. Terms defined in the list of variables. 21

2.2 The tree of scenarios considered in multi-stage NMPC [62]. 27

3.1 Flowcharts for the three variations of iLQG discussed in this work. Note: Figure 3.1a is the same as Figure 2.1 and is shown here for ease of comparison. Terms defined in the list of variables. 32

3.2 Control comparison between dual and adaptive iLQG and MS-SP-NMPC on linear example. 35

3.3 State comparison between dual and adaptive iLQG and MS-SP-NMPC on linear example. 36

3.4 Parameter comparison between dual and adaptive iLQG and MS-SP-NMPC on linear example. 36

3.5 Cost comparison between dual and adaptive iLQG and MS-SP-NMPC on linear example. 37

3.6 Control comparison between dual and adaptive iLQG on time-varying parameters example. 39

3.7 State comparison between dual and adaptive iLQG on time-varying parameters example. 39

3.8 Parameter comparison between dual and adaptive iLQG on time-varying parameters example. 40

3.9 Cost comparison between dual and adaptive iLQG on time-varying parameters example. 40

3.10 Frequency (%) of final cost of 100 seeded runs of the iLQG algorithms for the time-varying parameter example. 42

3.11 Distribution of the simulation time required for 100 seeded runs of the iLQG algorithms for the time-varying parameter example. 42

3.12 Control comparison between dual iLQG with and without the system's multiplicative noise being compensated for in the controller. 44
3.13 State comparison between dual iLQG with and without the system's multiplicative noise being compensated for in the controller. 45
3.14 Parameter comparison between dual iLQG with and without the system's multiplicative noise being compensated for in the controller. 45
3.15 Parameter ratio comparison between dual iLQG with and without the system's multiplicative noise being compensated for in the controller. 46
3.16 Cost comparison between dual iLQG with and without the system's multiplicative noise being compensated for in the controller. 46
3.17 Model-reference adaptive control flowchart for Rohr's example [58]. 48
3.18 Control comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example. 52
3.19 Control comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example for first seven seconds. 53
3.20 State comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example. 53
3.21 State comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example for first seven seconds. 54
3.22 Parameter comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example. 54
3.23 Parameter comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example for first seven seconds. 55
3.24 Parameter ratio comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example. 55
3.25 Parameter ratio comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example for first seven seconds. 56
3.26 Output comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example. 56
3.27 Output comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example for first seven seconds. 57
3.28 Cost comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example. 57

3.29 Cost comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example for first seven seconds. 58

4.1 Control comparison between dual and adaptive iLQG and dual MS-SP-NMPC on AM2 anaerobic digestion model with two uncertain parameters. 64
4.2 State comparison between dual and adaptive iLQG and dual MS-SP-NMPC on AM2 anaerobic digestion model with two uncertain parameters. 64
4.3 Parameter comparison between dual and adaptive iLQG and dual MS-SP-NMPC on AM2 anaerobic digestion model with two uncertain parameters. 65
4.4 Biogas production comparison between dual and adaptive iLQG and dual MS-SP-NMPC on AM2 anaerobic digestion model with two uncertain parameters. 65
4.5 Cost comparison between dual and adaptive iLQG and dual MS-SP-NMPC on AM2 anaerobic digestion model with two uncertain parameters. 66
4.6 Control comparison between dual and adaptive iLQG on AM2 anaerobic digestion model with seventeen uncertain parameters. 68
4.7 State comparison between dual and adaptive iLQG on AM2 anaerobic digestion model with seventeen uncertain parameters. 69
4.8 Parameter comparison between dual and adaptive iLQG on AM2 anaerobic digestion model with seventeen uncertain parameters. 69
4.9 Biogas production comparison between dual and adaptive iLQG on AM2 anaerobic digestion model with seventeen uncertain parameters. 70
4.10 Cost comparison between dual and adaptive iLQG on AM2 anaerobic digestion model with seventeen uncertain parameters. 70

5.1 SIR model compartmental diagram [43]. 73
5.2 SIDARTHE model compartmental diagram [30]. 74
5.3 Partial compartmental diagram for considering the impact of an overwhelmed ICU [30]. 76
5.4 Control comparison between dual and adaptive iLQG and dual MS-SP-NMPC on SIDARTHE COVID-19 model with two uncertain parameters. 81
5.5 State comparison between dual and adaptive iLQG and dual MS-SP-NMPC on SIDARTHE COVID-19 model with two uncertain parameters. 81
5.6 Parameter comparison between dual and adaptive iLQG and dual MS-SP-NMPC on SIDARTHE COVID-19 model with two uncertain parameters. 82

5.7 Cost comparison between dual and adaptive iLQG and dual MS-SP-NMPC on SIDARTHE COVID-19 model with two uncertain parameters. 82
5.8 Frequency (%) of final cost of 100 seeded runs of the iLQG algorithms for the 2 parameter SIDARTHE COVID-19 model with a rolling horizon length of 40. 85
5.9 Distribution of the simulation time required for 100 seeded runs of the iLQG algorithms for the 2 parameter SIDARTHE COVID-19 model with a rolling horizon length of 40. 85
5.10 Control comparison between dual and adaptive iLQG on modified SIDARTHE COVID-19 model with sixteen uncertain parameters. 87
5.11 State comparison between dual and adaptive iLQG on modified SIDARTHE COVID-19 model with sixteen uncertain parameters. 88
5.12 Parameter comparison between dual and adaptive iLQG on modified SIDARTHE COVID-19 model with sixteen uncertain parameters. 88
5.13 Cost comparison between dual and adaptive iLQG on modified SIDARTHE COVID-19 model with sixteen uncertain parameters. 89

List of Abbreviations

LQR	Linear quadratic regulator
iLQR	Iterative linear quadratic regulator
iLQG	Iterative linear quadratic Gaussian
MPC	Model predictive control
NMPC	Nonlinear model predictive control
MS-SP-NMPC	Multi-Stage Sigma Point Nonlinear Model Predictive Control
SIDARTHE	Susceptible, Infected, Diagnosed, Ailing, Recognized, Threatened, Healed, Extinct
ICU	Intensive Care Unit

List of Variables

A	linear process state coefficient matrix, ailing state for COVID-19 model in Chapter 5
B	linear process control coefficient matrix
C	total inorganic carbon concentration
$\mathcal{C}$	vector of linearized process noise terms
C^a	total inorganic carbon concentration
C^{in}	inlet concentrations of total inorganic carbon concentration
$C^{\mathbf{u}}$	linearized process noise control coefficient matrix
$C^{\mathbf{x}}$	linearized process noise state coefficient matrix
D	eigenvalues from eigenvalue decomposition, dilution rate in Chapter 4, detected population state for COVID-19 model in Chapter 5
$\mathcal{D}$	vector of linearized measurement noise terms
$D^{\mathbf{u}}$	linearized measurement noise control coefficient matrix
$D^{\mathbf{x}}$	linearized measurement noise state coefficient matrix
E	linearized measurement state coefficient matrix, extinct state for COVID-19 model in Chapter 5
F	linearized measurement control coefficient matrix
$\mathcal{F}$	process noise scaling matrix
ℓ	stage cost
$\mathcal{G}$	regularized cost-to-go quadratic state dependent shorthand term
$\mathcal{G}$	measurement noise scaling matrix

$G^{\mathbf{x}}$	cost-to-go quadratic state dependent shorthand term
$G^{\hat{\mathbf{x}}}$	cost-to-go quadratic state estimate dependent shorthand term
H	cost-to-go quadratic control dependent shorthand term, transfer function in section 3.5, healed state for COVID-19 model in Chapter 5
$\mathcal{H}$	regularized cost-to-go quadratic control dependent shorthand term
I	infected population state for COVID-19 model
J	cost function
K	gain
K_H	Henry's constant
K_{I_2}	inhibition constant associated with μ_2
K_{S_1}	half-saturation constant associated with μ_1
K_{S_2}	half-saturation constant associated with μ_2
L	state dependant control gain
$\mathcal{M}$	estimator process noise shorthand term
N	control horizon, total population for COVID-19 model in Chapter 5
P	state-control quadratic cost-weighting matrix
P_{CO_2}	partial pressure of carbon dioxide
$\mathcal{P}$	estimator measurement noise shorthand term
P_t	total pressure
Q	state quadratic cost-weighting matrix
Q_f	terminal state quadratic cost-weighting matrix
R	control quadratic cost-weighting matrix, recovered population for COVID-19 model in Chapter 5
S	susceptible population state for COVID-19 model
S_1	organic substrate concentration
S_1^a	organic substrate concentration of feedstock a
S_1^{in}	inlet concentrations of organic substrate concentration
S_2	volatile fatty acid concentration

S_2^a	volatile fatty acid concentration
S_2^{in}	inlet concentrations of volatile fatty acid concentration
$S^{\mathbf{x}}$	state deviation dependent quadratic cost-to-go term
$S^{\hat{\mathbf{x}}}$	estimate of state deviation dependent quadratic cost-to-go term
$S^{\mathbf{x}\hat{\mathbf{x}}}$	state deviation and estimate of state deviation dependent quadratic cost-to-go term
T	discrete sampling period, threatened state for COVID-19 model in Chapter 5
T_1	threatened state for COVID-19 model that do not require ICU treatment
T_2	threatened state for COVID-19 model that do require ICU treatment
T_{ICU}	capacity of the threatened state in the ICU
$T_{ICU_{\max}}$	maximum capacity of the threatened state in the ICU
$T_{ICU_{\min}}$	minimum capacity of the threatened state in the ICU
V	eigenvectors from eigenvalue decomposition
V_k	optimal cost-to-go from time k
X_1	concentration of acidogenic bacteria
X_2	concentration of methanogenic bacteria
Z	total alkalinity concentration
Z^a	total alkalinity concentration
Z^{in}	inlet concentrations of total alkalinity concentration
$\hat{\mathbf{a}}$	adaptation parameter vector
a_m	negative pole of reference model transfer function
a_p	negative pole of plant transfer function
$\hat{a}_r$	input reference adaptation parameter
$\hat{a}_y$	output adaptation parameter
$\mathbf{b}_{\max}$	squashing function maximum vector limit
$\mathbf{b}_{\min}$	squashing function minimum vector limit
$\mathbf{c}$	constant vector
$\mathbf{c}^a$	augmented constant vector

$\mathbf{c}$	scalar linearized process noise terms
$\mathbf{d}$	uncertain parameter vector
$\mathbf{d}$	scalar linearized measurement noise terms
$\hat{\mathbf{d}}$	estimate of uncertain parameter vector
$\mathbf{d}^p$	uncertain parameter trajectory vector
e	output error
f	system dynamics
g	measurement equation
$\mathbf{g}$	cost-to-go linear control dependent shorthand term
h	terminal cost
k	discrete time step counter
k_1	yield for substrate degradation
k_2	yield for volatile fatty acid production
k_3	yield for volatile fatty acid consumption
k_4	yield for carbon dioxide production
k_5	yield for carbon dioxide production
k_6	yield for methane production
$k_L a$	liquid/gas transfer constant
k_m	gain for reference model
k_p	gain for plant
l	scalar control gain
ℓ_Q	quadratized stage cost
$m^{\mathbf{e}}$	unconditional mean of the error of the estimate of state deviation vector
$m^{\hat{\mathbf{x}}}$	unconditional mean of the estimate of the state deviation vector
p	Laplace variable
q	state scalar cost-weighting matrix
q_a	flow rate of feedstock A
q_{biogas}	molar flow rate of biogas
$\mathbf{q}$	state linear cost-weighting matrix
q_{CH_4}	molar flow rate of methane

q_{CO_2}	molar flow rate of carbon dioxide
r	input reference
$\mathbf{r}$	control linear cost-weighting matrix
$\mathbf{s}(\mathbf{u})$	squashing function
s	scalar cost-to-go term
$\mathbf{s}^{\mathbf{x}}$	state deviation dependent linear cost-to-go term
$\mathbf{s}^{\hat{\mathbf{x}}}$	estimate of state deviation dependent linear cost-to-go term
u	control vector
$\mathbf{u}$	control deviation vector
$\mathbf{u}^p$	control trajectory vector
$\bar{\mathbf{u}}^p$	nominal control vector
$\mathbf{u}^{p^*}$	optimal control trajectory
u_{ss}	steady state control input
v	cost-to-go function
$\mathbf{v}$	measurement noise,
	regressor vector in section 3.5
$\mathbf{w}$	process noise
x	state vector
$\mathbf{x}^a$	augmented state vector
$\mathbf{x}$	state deviation vector
$\dot{x}$	time derivative of x
$\hat{\mathbf{x}}$	estimate of state deviation vector
$\boldsymbol{x}^p$	state trajectory vector
$\boldsymbol{x}^{p^a}$	augmented state trajectory vector
$\bar{\mathbf{x}}^p$	nominal state vector
$\mathbf{x}^{p^*}$	desired terminal state
$\mathbf{y}$	measurement deviation vector
y_m	output of reference model
$\mathbf{y}^p$	measurement trajectory vector
$\bar{\mathbf{y}}^p$	nominal measurement vector
y_{ss}	steady state output
Δ	regularization term modification factor

Δ_0	minimum regularization term modification factor
Δt	discrete time step
Σ	covariance
Σ^x	state covariance
Σ^d	parameter covariance
Σ^{x^a}	augmented state covariance
ϕ_{CO_2}	concentration of dissolved carbon dioxide
α	backtracking search parameter, fraction of biomass in the liquid phase in section 4, rate of infection of state S by state I for COVID-19 model in Chapter 5
$\alpha_{\max}$	maximum rate of infection of state S by state I
$\alpha_{\min}$	minimum rate of infection of state S by state I
β	transmission rate between the infected and susceptible populations
$\beta_{\max}$	maximum transmission rate between the infected and susceptible populations
$\beta_{\min}$	minimum transmission rate between the infected and susceptible populations
γ	adaptation law constant, inverse infectious period for COVID-19 model in Chapter 5
$\gamma_{\max}$	maximum rate of infection of state S
$\gamma_{\min}$	minimum rate of infection of state S
ϵ	rate of diagnosis of state I
$\epsilon_{\max}$	maximum rate of diagnosis of state I
$\epsilon_{\min}$	minimum rate of diagnosis of state I
ζ	Gaussian noise for linearized process, rate of developing symptoms of states I and D for COVID-19 model in Chapter 5
η	Gaussian noise for linearized measurement
η_{α_1}	effectiveness of control action 1 to influence α
η_{α_2}	effectiveness of control action 2 to influence α

η_{β_1}	effectiveness of control action 1 to influence β
η_{β_2}	effectiveness of control action 2 to influence β
η_{β_5}	effectiveness of control action 5 to influence β
η_{γ_1}	effectiveness of control action 1 to influence γ
η_{γ_2}	effectiveness of control action 2 to influence γ
η_{ϵ_1}	effectiveness of control action 1 to influence ϵ
η_{ϵ_3}	effectiveness of control action 3 to influence ϵ
η_{θ_1}	effectiveness of control action 1 to influence θ
η_{θ_4}	effectiveness of control action 4 to influence θ
θ	rate of diagnosis of state A
$\theta_{\max}$	maximum rate of diagnosis of state A
$\theta_{\min}$	minimum rate of diagnosis of state A
κ	rate of recovery of states A and R
λ	regularization term, rate of recovery of states I and D for COVID-19 model in Chapter 5
$\lambda_{\max}$	maximum regularization term setting
$\lambda_{\min}$	minimum regularization term setting
μ	rate of developing acute symptoms of states A and R
μ_1	specific acidogenic bacteria growth rate, rate of developing acute symptoms of states A and R for those that do not require ICU treatment for COVID-19 model in Chapter 5
$\bar{\mu}_1$	maximum acidogenic bacteria growth rate
μ_2	specific methanogenic bacteria growth rate, rate of developing acute symptoms of states A and R for those that do require ICU treatment for COVID-19 model in Chapter 5
$\bar{\mu}_2$	maximum methanogenic bacteria growth rate
π	control policy
σ	rate of recovery of state T

σ_1	rate of recovery of state T for those that do not require ICU treatment
$\sigma_{1_{\max}}$	maximum rate of recovery of state T for those that do not require ICU treatment
$\sigma_{1_{\min}}$	minimum rate of recovery of state T for those that do not require ICU treatment
σ_2	rate of recovery of state T for those that do require ICU treatment
$\sigma_{2_{\max}}$	maximum rate of recovery of state T for those that do require ICU treatment
$\sigma_{2_{\min}}$	minimum rate of recovery of state T for those that do require ICU treatment
τ	rate of mortality of state T
τ_1	rate of mortality of state T for those that do not require ICU treatment
$\tau_{1_{\max}}$	maximum rate of mortality of state T for those that do not require ICU treatment
$\tau_{1_{\min}}$	minimum rate of mortality of state T for those that do not require ICU treatment
τ_2	rate of mortality of state T for those that do require ICU treatment
$\tau_{2_{\max}}$	maximum rate of mortality of state T for those that do require ICU treatment
$\tau_{2_{\min}}$	minimum rate of mortality of state T for those that do require ICU treatment

Chapter 1

Introduction

In real-world control problems, there are often uncertainties due to the system dynamics being difficult to model and/or measurements being inaccurate or unavailable. These uncertainties make it difficult to determine what control actions should be taken to have a system achieve a desired objective. System identification techniques can identify these uncertainties through testing before the system is put into normal operation, but this option is not always available and cannot identify time-varying uncertainties. Adaptive control techniques can identify uncertain parameters while directing a system toward a desired objective, but cannot determine what actions would result in measurements with more information about the uncertainties, and are therefore passively adaptive. Actively adaptive control techniques, on the other hand, estimate the reductions in uncertainty that will result from their control actions and probe the system to identify the uncertain parameters to a sufficient level such that desired goal is optimized. This actively adaptive control is known as dual control, as the controls are chosen to learn about the uncertainties only when it is likely to result in a lower long-term cost.

1.1 Motivation and gap

When designing a control system for systems with uncertain parameters, there are several approaches that could be used. Robust approaches consider the set of possible values that the parameters could take and can account for the uncertainties to limit their impact on achieving the control goals (*e.g.*, [40]). Stochastic approaches consider the probability density function of the uncertain parameters to account for parameter realizations with the higher chances of being true (*e.g.*, [7]). Adaptive approaches continuously update estimates of the uncertain parameters using

measurements from the system and controls the system as if these estimates are the true values of the parameters (*e.g.*, [16]). Finally, dual approaches consider what actions could be taken to improve the information in future measurements in such a way that the resulting reduction in future costs is greater than the cost of these probing actions (*e.g.*, [48]).

Adaptive control methods are passively adaptive, in that they only consider changes to parameter uncertainties due to past measurements. Dual control on the other hand is an actively adaptive control method, in that it actively modifies its control actions to seek out future measurements that will reduce parameter uncertainties.

Dual control has three features: it is probing, cautious, and selective [29]. Dual control is probing in that it modifies its control signals to obtain more information-rich measurements, it is cautious in that it will tend to make smaller control actions when uncertainties are high, and it is selective in that it will only attempt to identify parameters that impact the system's performance.

An analogy for dual control that has been used is driving a new car from location A to location B [33]. Although it may be more efficient to just start driving under assumptions of how cars generally operate and changing your driving based on differences that you notice (taking an adaptive control approach), taking the time to quickly test (probe) the car's responsiveness to the brake and the gas pedals at a safe time (cautious) will be informative, and although this approach will slow down the start of your trip a bit, could prevent an accident. Also, although understanding how the air conditioning works may also be helpful, it is not as important to know as the car's responsiveness (selectiveness).

Dual control was first identified by Feldbaum in 1960 [19], and they recognized that the optimal dual control problem could be solved using stochastic dynamic programming. Unfortunately, stochastic dynamic programming involves solving the Bellman equations, which are generally computationally inefficient due to what Bellman termed the *curse of dimensionality*, where the size of the problem grows exponentially with the number of states [67]. Although dual control is the only class of stochastic control policies that exhibits active learning, computationally tractable approximations of dual control have been developed [8].

There are two types of approximations of dual control; explicit and implicit. Explicit dual control methods directly incorporate one or more of the three features of dual control into the objective function. Although this approach is simple, the relative importance of the regulation and identification functions must be defined [62]. Implicit dual controllers on the other hand, approximate the Bellman equations in such a way that the reduction of uncertainty due to future measurements is estimated, which generally comes at the cost of higher computational effort

compared to explicit methods [23]. The relative importance of the dual features and the other objectives does not have to be quantified with the implicit approach as they are linked, but probing for increased parameter information is only done if it will reduce future costs. By not prescribing the relative importance of the control objective and these dual features, implicit methods can give better results than explicit methods [8].

There are several existing dual implicit approaches, but each only considers limited control or parameter realizations to make the control problem tractable or is only applicable for systems with a limited number of states.

Bayard and Schumitzky [8] use an iteration in policy space algorithm that combines particle filtering with Monte Carlo simulations to estimate the cost-to-go and iteratively improve on a given control policy but is limited to control inputs that only take on two discrete values. Sehr and Bitmead [56] approximate the system dynamics as a Partially Observable Markov Decision Processes, allowing the dual stochastic MPC problem to be solved explicitly for small and medium-sized problems.

Thangavel *et al.* [63] and Hanssen and Foss [21] take a multi-stage approach to implicit dual control and consider a branching network of scenarios, where each branch represents the system's predicted response for a single realization of the uncertain parameters. The optimization problem that is solved is to determine the control actions (over the control horizon) which minimize the sum of the costs of the scenarios over the prediction horizon. The reduction in the uncertainties due to future measurements is estimated for each time step in a "robust" horizon, and this reduction is reflected in the selection of the discrete uncertainty values upon which the subsequent scenarios are based. As this process causes the number of scenarios to grow exponentially, the robust horizon is usually chosen to be less than the control horizon, and the uncertainties after that point are assumed to be equal to the nominal value, no longer incorporating the dual features. The existing implicit dual approaches are limited by Bellman's curse of dimensionality as they do not take a derivative-based approach in continuous state space.

In this work, an existing derivative-based control method is extended to handle systems with uncertain parameters in either an adaptive or dual manner. Iterative linear quadratic Gaussian (iLQG) control can calculate locally-optimal control policies for nonlinear stochastic systems. iLQG is based on continuous state space with the use of derivatives of a linearized system about a nominal control trajectory. Although iLQG is not a dual or adaptive control algorithm, it can handle nonlinear and stochastic systems with many states through its derivative-based approach in continuous state space and avoids Bellman's curse of dimensionality. By modifying iLQG to

treat the uncertain parameters as uncertain states, the resulting novel dual iLQG control algorithm can predict how changes to the inputs and states can result in future reductions in the parameter uncertainty and therefore increase overall performance (lower costs). By calculating the derivatives of the (implicit) cost function at each time step, dual iLQG can identify changes to the inputs that can decrease parameter uncertainty, and although these actions have an associated cost, they decrease the overall cost over the control trajectory. This work's major contribution is extending iLQG to be an implicitly dual control method.

1.2 Organization of the thesis

The remainder of this thesis starts by covering the relevant material on optimal, adaptive, and dual control, before describing this new dual iLQG approach in detail and investigating its robustness as well as its superior performance over an existing implicit dual control method for particular examples. Next, dual iLQG is compared against adaptive iLQG and dual and adaptive MS-SP-NMPC on two applications; an anaerobic digester and COVID-19 to demonstrate its performance on complex nonlinear systems with large numbers of states. Finally, the key contributions of this thesis are highlighted, the many potential applications of dual iLQG are described, and the limitations and future direction of this work are discussed.

Chapter 2

Background

2.1 Optimal control

In optimal control, the control actions to a dynamic system over a period of time are determined by solving an optimization problem where a given objective function is minimized [67]. The objective function is chosen to cause the system to demonstrate a particular behaviour, such as following a particular state trajectory or minimizing energy usage. Objective, or cost, functions can have terms that impose costs at each point in time, known as stage costs, or terms that only impose costs at the final time, known as terminal costs. Constraints can also be imposed on the states and/or controls in the form of inequalities or equalities [67].

A common example of an optimal control problem is finding the cheapest connections to fly to a destination [67]. Here, the state x represents the cities, the control u represents the choices of which flights to take, $\text{cost}(x, u)$ represents the cost of the plane ticket, and $\text{next}(x, u)$ represents the city where the flight u from city x lands. The goal is therefore to find the series of flights $(u_0, u_1, \ldots, u_{n-1})$ to get from the initial departure city x_0 to the final destination x_n that minimizes the total cost

$$J(x, u) = \sum_{k=0}^{N-1} \text{cost}(x_k, u_k), \tag{2.1}$$

where the dynamics are $x_{k+1} = \text{next}(x_k, u_k)$, $x \in \mathcal{X}$, and $u \in \mathcal{U}(x)$, where $\mathcal{X}$ and $\mathcal{U}(x)$ are finite sets.

Problems such as this one can be solved using Bellman's optimality principle, which states that "an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy" [9]. Therefore, optimal control

problems can be simplified from a single optimization of a control sequence to a sequence of optimizations of individual control decisions starting from the final control decision and proceeding backward in time. Once the optimal control action at a given time step is determined, the cost of that time step is used to calculate the optimal cost-to-go function $v(x)$, which is the minimum total cost to reach the final state from each state, and can be represented as

$$v(x) = \min_{u \in \mathcal{U}(x)} \left[\text{cost}(x, u) + v(\text{next}(x, u)) \right]. \tag{2.2}$$

Once the cost-to-go function has been calculated at every time step by starting at the final time step and doing this backward pass through time, the optimal cost-to-go can be determined by selecting the lowest cost-to-go at the initial time step, which provides the optimal control trajectory.

In the context of the flight example given above, this approach means that the problem can be solved by considering flights to a single city at a time and keeping running totals of the flight costs. The cost of flights to the final destination can be recorded first, giving the cost-to-go (to the final destination) from each of those cities. The cost of flights to those second-to-last destinations from other cities can then be added to their respective cost-to-go functions, giving the cost-to-go from those cities. This process then continues until all of these possible routes start with the pre-determined initial departure city. The optimal control problem is then solved by simply selecting the lowest cost-to-go from the initial departure city.

This approach is known as dynamic programming and it is well-suited to optimal control problems with defined final costs and a small number of states. As dynamic programming is an exhaustive method that considers all of the possible control trajectories, the size of the optimization problem grows exponentially with the number of states. This is Bellman's *curse of dimensionality*, and it likely has no general solution [67], but several approximately optimal control strategies can circumvent this issue.

2.1.1 The linear quadratic regulator (LQR)

A special case of optimal control is the linear quadratic regulator (LQR), which is applicable to deterministic linear systems that have quadratic costs [14]. For the discrete-time linear system

$$x_{k+1} = Ax_k + Bu_k \tag{2.3}$$

with the quadratic cost function over a finite horizon N of

$$J = x_N^\top Q x_N + \sum_{k=0}^{N-1} \left(x_k^\top Q x_k + u_k^\top R u_k + 2 x_k^\top P u_k \right), \tag{2.4}$$

the optimal control at time step k is

$$u_k = -K_k x_k, \tag{2.5}$$

where

$$K_k = \left(R + B^\top S_{k+1} B \right)^{-1} \left(B^\top S_{k+1} A + P^\top \right) \tag{2.6}$$

and S_k is found by using the Riccati equation

$$S_{k-1} = A^\top S_k A - \left(A^\top S_k B + P \right) \left(R + B^\top S_k B \right)^{-1} \left(B^\top S_k A + P^\top \right) + Q \tag{2.7}$$

iteratively backwards in time from the terminal condition of $S_N = Q$.

2.1.2 Iterative LQR (iLQR)

Iterative LQR (iLQR) extends LQR to nonlinear deterministic systems by linearizing the system about a nominal state-control trajectory. Consider a continuous-time nonlinear deterministic dynamic system with state $\mathbf{x}^p \in \mathbb{R}^{n_x}$ and control $\mathbf{u}^p \in \mathbb{R}^{n_u}$

$$\dot{\mathbf{x}}^p = f\left(\mathbf{x}^p, \mathbf{u}^p \right). \tag{2.8}$$

Using this system, iLQR uses a quadratic cost function over a finite horizon N of

$$J = \frac{1}{2} \left(\mathbf{x}_N^p - \mathbf{x}^{p^*} \right)^\top Q_f \left(\mathbf{x}_N^p - \mathbf{x}^{p^*} \right) + \frac{1}{2} \sum_{k=0}^{N-1} \left(\mathbf{x}_k^{p^\top} Q \mathbf{x}_k^p + \mathbf{u}_k^{p^\top} R \mathbf{u}_k^p \right), \tag{2.9}$$

where $\mathbf{x}^{p^*}$ is the desired terminal state, Q_f and Q are the state cost-weighting matrices that are symmetric positive semi-definite, and R is the control cost-weighting matrix which is positive definite.

This iterative algorithm starts with a given nominal control trajectory, $\bar{\mathbf{u}}^p$, which is used to determine the nominal state trajectory $\bar{\mathbf{x}}^p$ through integration of the discreteized version of (2.8),

$$\bar{\mathbf{x}}_{k+1}^p = \bar{\mathbf{x}}_k^p + f(\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p) \Delta t. \tag{2.10}$$

Variables representing the deviations between the actual and nominal state and control trajectories can then be expressed as $\mathbf{x}_k = \mathbf{x}_k^p - \bar{\mathbf{x}}_k^p$ and $\mathbf{u}_k = \mathbf{u}_k^p - \bar{\mathbf{u}}_k^p$, and the dynamic system and output equation in equations (2.8) can be expressed as

$$\bar{\mathbf{x}}_{k+1}^p + \mathbf{x}_{k+1} = f\left(\bar{\mathbf{x}}_k^p + \mathbf{x}_k, \bar{\mathbf{u}}_k^p + \mathbf{u}_k\right). \tag{2.11}$$

Linearizing equation (2.11) about $(\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p)$ and subtracting equation (2.10), the linear approximation of equation (2.8) are given as

$$\begin{aligned} \mathbf{x}_{k+1} &= A_k \mathbf{x}_k + B_k \mathbf{u}_k \\ A_k &= \frac{\partial f}{\partial \mathbf{x}_k^p} \\ B_k &= \frac{\partial f}{\partial \mathbf{u}_k^p}. \end{aligned} \tag{2.12}$$

As shown in [36], the locally-optimal controller can be derived as

$$\mathbf{u}_k = -K\mathbf{x}_k - K_v v_{k+1} - K_u \bar{\mathbf{u}}_k^p, \tag{2.13}$$

$$K = \left(B_k^\top S_{k+1} B_k + R\right)^{-1} B_k^\top S_{k+1} A_k, \tag{2.14}$$

$$K_v = \left(B_k^\top S_{k+1} B_k + R\right)^{-1} B_k^\top, \tag{2.15}$$

$$K_u = \left(B_k^\top S_{k+1} B_k + R\right)^{-1} R, \tag{2.16}$$

$$S_k = A_k^\top S_{k+1} \left(A_k - B_k K\right) + Q, \tag{2.17}$$

$$v_k = \left(A_k - B_k K\right)^\top v_{k+1} - K^\top R \bar{\mathbf{u}}_k^p + Q \bar{\mathbf{x}}_k^p, \tag{2.18}$$

which is solved iteratively backwards in time from the terminal conditions of $S_N = Q_f$ and $v_N = Q_f\left(\bar{\mathbf{x}}_N^p - x^{p^*}\right)$. Once $\mathbf{u}_k$ is determined, the improved nominal control trajectory is given as

$$\mathbf{u}_k^{p^*} = \bar{\mathbf{u}}_k^p + \mathbf{u}_k. \tag{2.19}$$

This control approach is only locally-optimal due to it being a trajectory-based approach [36]. As iLQR considers deviations around an initial nominal control trajectory, $\bar{\mathbf{u}}_k^p$, the resulting solution can only be optimal within that region of control space, and no claims of global optimality can be made.

iLQR can also be applied in a moving horizon setting where feedback is used to improve performance by accounting for inaccuracies due to the linearization process. To do this, the

algorithm is initialized with a nominal control trajectory as described above that could be a vector of zeros, randomly generated, or any other form of seed. The iLQR algorithm is then run for a given finite time horizon, resulting in an improved control policy. The control policy is then applied to the true system for a single time step, after which a state measurement from the system is obtained. The length of the time horizon and the improved control policy can then be adjusted and used to run the iLQR algorithm. This process can be repeated for a problem with a finite time horizon until the final time step is reached, or continuously for problems with infinite time horizons.

2.1.3 Iterative Linear Quadratic Gaussian (iLQG)

iLQG extends iLQR to systems that are nonlinear systems that are stochastic and do not have quadratic costs. To handle non-quadratic cost functions, the cost function is "quadratized" about the nominal state-control trajectory in a similar way that the system dynamics are linearized. [1] Since the states are uncertain, the measurement dynamics are also included in the iLQG algorithm, and a filter is required to estimate the states' mean values and covariances.

Importantly, iLQG can handle multiplicative noise in both the state and measurement dynamics. Multiplicative noise occurs when there is an undesirable random signal that is multiplied by one or more states or controls of the system [66]. Many stochastic control approaches are based on only additive noise, where the noise term is independent of the state or control vectors, and iLQG's more general approach is a beneficial feature.

The forward integration of the system dynamics to obtain the nominal state trajectory from the nominal control trajectory and the calculations of the derivatives required for the linearization of the system dynamics as well as the "quadratization" of the cost function are grouped together in what's known as a forward pass. The next step in the algorithm is the estimator, which uses the noisy measurements to infer the value of the states and their covariances. Next, a backward pass is required to calculate a quadratic approximation to the cost-to-go function, and then the optimal control deviations can be found.

Since the linear approximation of the nonlinear system dynamics loses accuracy for larger control deviations, a line search is implemented to iteratively reduce the control deviations if the solution's estimated cost is not less than the cost of the initial nominal trajectory, improving the algorithm's convergence.

[1] Please note that non-quadratic cost functions can be handled in the same way for iLQR, but was not discussed in that section for the sake of clarity.

Finally, as with iLQR, iLQG gives locally-optimal solutions due to its trajectory-based approach which also allows it to be very efficient. To account for this, the algorithm can be initialized multiple times with different nominal control trajectories, known as seeds, to search a larger control space. These seeds may converge to different local minima, and a selection criteria can be determined to select a single solution from this set of solutions.

Additional details on iLQR and iLQG publications and their differences can be found in Appendix A.

2.1.3.1 The forward pass

Linearizing the system dynamics

Consider a nonlinear stochastic dynamic system with state trajectory $\mathbf{x}^p \in \mathbb{R}^{n_x}$, control trajectory $\mathbf{u}^p \in \mathbb{R}^{n_u}$, and output trajectory $\mathbf{y}^p \in \mathbb{R}^{n_y}$ with standard and interdependent Brownian noise $\mathbf{w} \in \mathbb{R}^{n_w}$ and $\mathbf{v} \in \mathbb{R}^{n_v}$

$$\mathrm{d}\mathbf{x}^p(t) = f(\mathbf{x}^p, \mathbf{u}^p)\mathrm{d}t + \mathcal{F}(\mathbf{x}^p, \mathbf{u}^p)\mathrm{d}\mathbf{w}(t) \tag{2.20}$$

$$\mathrm{d}\mathbf{y}^p(t) = g(\mathbf{x}^p, \mathbf{u}^p)\mathrm{d}t + \mathcal{G}(\mathbf{x}^p, \mathbf{u}^p)\mathrm{d}\mathbf{v}(t). \tag{2.21}$$

Starting with a nominal control trajectory $\bar{\mathbf{u}}^p$, a nominal state trajectory $\bar{\mathbf{x}}^p$ can be determined by applying $\bar{\mathbf{u}}^p$ to a deterministic and discreteized version of equation (2.20),

$$\bar{\mathbf{x}}^p_{k+1} = \bar{\mathbf{x}}^p_k + f(\bar{\mathbf{x}}^p_k, \bar{\mathbf{u}}^p_k)\Delta t, \tag{2.22}$$

through Euler integration with $\bar{\mathbf{x}}^p(0) = \mathbf{x}^p(0)$. The nominal output trajectory can then be obtained with both $\bar{\mathbf{x}}^p$ and $\bar{\mathbf{u}}^p$ with a deterministic and discreteized version of equation (2.21),

$$\bar{\mathbf{y}}^p_k = g(\bar{\mathbf{x}}^p_k, \bar{\mathbf{u}}^p_k)\Delta t. \tag{2.23}$$

Variables representing the deviations of the state, control, and output trajectories from the nominal trajectories can then be expressed as $\mathbf{x}_k = \mathbf{x}^p_k - \bar{\mathbf{x}}^p_k$, $\mathbf{u}_k = \mathbf{u}^p_k - \bar{\mathbf{u}}^p_k$, and $\mathbf{y}_k = \mathbf{y}^p_k - \bar{\mathbf{y}}^p_k$ respectively, and the dynamic system and output equation in equations (2.20) and (2.21) can be expressed as

$$\bar{\mathbf{x}}_{k+1}^p + \mathbf{x}_{k+1} = \bar{\mathbf{x}}_k^p + \mathbf{x}_k + f(\bar{\mathbf{x}}_k^p + \mathbf{x}_k, \bar{\mathbf{u}}_k^p + \mathbf{u}_k)\Delta t + \mathcal{F}(\bar{\mathbf{x}}_k^p + \mathbf{x}_k, \bar{\mathbf{u}}_k^p + \mathbf{u}_k)\Delta \mathbf{w}_k \tag{2.24}$$

$$\bar{\mathbf{y}}_k^p + \mathbf{y}_k = g(\bar{\mathbf{x}}_k^p + \mathbf{x}_k, \bar{\mathbf{u}}_k^p + \mathbf{u}_k)\Delta t + \mathcal{G}(\bar{\mathbf{x}}_k^p + \mathbf{x}_k, \bar{\mathbf{u}}_k^p + \mathbf{u}_k)\Delta \mathbf{v}_k. \tag{2.25}$$

Linearizing equations (2.24) and (2.25) about the nominal state-control trajectory $(\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p)$ and subtracting equations (2.22) and (2.23), locally linear discrete forms of equations (2.20) and (2.21) are obtained as

$$\begin{aligned}
\mathbf{x}_{k+1} &= A_k\mathbf{x}_k + B_k\mathbf{u}_k + \mathcal{C}_k(\mathbf{x}_k, \mathbf{u}_k)\zeta_k && (2.26)\\
\mathbf{y}_k &= F_k\mathbf{x}_k + E_k\mathbf{u}_k + \mathcal{D}_k(\mathbf{x}_k, \mathbf{u}_k)\eta_k && (2.27)\\
A_k &= \frac{\partial f}{\partial \mathbf{x}_k^p}\\
B_k &= \frac{\partial f}{\partial \mathbf{u}_k^p}\\
\mathcal{C}_k &\triangleq \left[\mathbf{c}_{1,k} + C_{1,k}^{\mathbf{x}}\mathbf{x}_k + C_{1,k}^{\mathbf{u}}\mathbf{u}_k, \ldots, \mathbf{c}_{n_w,k} + C_{n_w,k}^{\mathbf{x}}\mathbf{x}_k + C_{n_w,k}^{\mathbf{u}}\mathbf{u}_k\right]\\
\mathbf{c}_{i,k} &= \sqrt{\Delta t}\mathcal{F}^{[i]}\\
C_{i,k}^{\mathbf{x}} &= \sqrt{\Delta t}\frac{\partial \mathcal{F}^{[i]}}{\partial \mathbf{x}_k^p}\\
C_{i,k}^{\mathbf{u}} &= \sqrt{\Delta t}\frac{\partial \mathcal{F}^{[i]}}{\partial \mathbf{u}_k^p}\\
F_k &= \frac{\partial g}{\partial \mathbf{x}_k^p}\\
E_k &= \frac{\partial g}{\partial \mathbf{u}_k^p}\\
\mathcal{D}_k &\triangleq \left[\mathbf{d}_{1,k} + D_{1,k}^{\mathbf{x}}\mathbf{x}_k + D_{1,k}^{\mathbf{u}}\mathbf{u}_k, \ldots, \mathbf{d}_{n_v,k} + D_{n_v,k}^{\mathbf{x}}\mathbf{x}_k + D_{n_v,k}^{\mathbf{u}}\mathbf{u}_k\right]\\
\mathbf{d}_{i,k} &= \frac{1}{\sqrt{\Delta t}}\mathcal{G}^{[i]}\\
D_{i,k}^{\mathbf{x}} &= \frac{1}{\sqrt{\Delta t}}\frac{\partial \mathcal{G}^{[i]}}{\partial \mathbf{x}_k^p}\\
D_{i,k}^{\mathbf{u}} &= \frac{1}{\sqrt{\Delta t}}\frac{\partial \mathcal{G}^{[i]}}{\partial \mathbf{u}_k^p},
\end{aligned}$$

where the Gaussian noise terms $\zeta_k \in \mathbb{R}^{n_w}$ and $\eta_k \in \mathbb{R}^{n_v}$ are independent and zero-mean with covariances $\Sigma^\zeta = I$ and $\Sigma^\eta = I$ respectively and the superscript $[i]$ indicates the ith columns of $\mathcal{F} \in \mathbb{R}^{n_x \times n_w}$ and $\mathcal{G} \in \mathbb{R}^{n_y \times n_v}$ respectively.

Quadratizing the cost function

The total cost J from an initial state $\mathbf{x}_0^p$ to a terminal time N can be defined in terms of a stage

cost $\ell(t, \mathbf{x}_k^p, \mathbf{u}_k^p) \geq 0$ and a terminal cost $h(\mathbf{x}_N^p) \geq 0$ as

$$J(\mathbf{x}_0^p, \mathbf{u}^p) = E\left[\sum_{k=0}^{N-1} \ell(k, \mathbf{x}_k^p, \mathbf{u}_k^p) + h(\mathbf{x}_N^p)\right]. \tag{2.28}$$

Since $\mathbf{x}^p$ can be obtained through the integration of equation (2.20) from $\mathbf{x}_0^p$, the optimal control problem is to find the optimal control trajectory $\mathbf{u}^{p^*}$ which minimizes J,

$$\mathbf{u}^{p^*} \triangleq \underset{\mathbf{u}^p}{\operatorname{argmin}}\ J(\mathbf{x}_0^p, \mathbf{u}^p). \tag{2.29}$$

The cost-to-go J_k is the sum of the costs occurring from time step k until the terminal time N, during which the partial control trajectory $\mathbf{u}_{k:N}^p \triangleq \{\mathbf{u}_k^p, \mathbf{u}_{k+1}^p, \ldots \mathbf{u}_N^p\}$ is applied, starting from $\mathbf{x}_k^p$,

$$J_k(\mathbf{x}_k^p, \mathbf{u}_{k:N}^p) = E\left[\sum_{i=k}^{N-1} \ell(i, \mathbf{x}_i^p, \mathbf{u}_i^p) + h(\mathbf{x}_N^p)\right]. \tag{2.30}$$

The optimal cost-to-go at time k starting from $\mathbf{x}_k^p$ is given as

$$V_k(\mathbf{x}_k^p) \triangleq \min_{\mathbf{u}_{k:N}^p} J_k(\mathbf{x}_k^p, \mathbf{u}_{k:N}^p) \tag{2.31}$$

and its value at the terminal time N is

$$V_N(\mathbf{x}_N^p) \triangleq E\left[h(\mathbf{x}_N^p)\right]. \tag{2.32}$$

As described in [61], the Dynamic Programming Principle then reduces the optimal control problem to a sequence of minimizations over a single control starting at the terminal time, progressing backwards in time to the initial state,

$$V_k(\mathbf{x}_k^p) = \min_{\mathbf{u}_k^p} E\left[\ell(\mathbf{x}_k^p, \mathbf{u}_k^p) + V_{k+1}(\mathbf{x}_{k+1}^p)\right]. \tag{2.33}$$

Quadratizing the stage cost term in equation (2.33) and writing it in terms of the deviation variables, as was done with the system dynamics in equation (2.26), results in ℓ_Q as

$$
\begin{aligned}
\ell_Q &= q_k + \delta\mathbf{x}_k^\top \mathbf{q}_k + \frac{1}{2}\delta\mathbf{x}_k^\top Q_k \delta\mathbf{x}_k^\top + \delta\mathbf{u}_k^\top \mathbf{r}_k + \frac{1}{2}\delta\mathbf{u}_k^\top R_k \delta\mathbf{u}_k + \delta\mathbf{u}_k^\top P_k \delta\mathbf{x}_k^\top \qquad (2.34)\\
q_k &= \Delta t\ \ell|_{\{\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p\}}\\
\mathbf{q}_k &= \Delta t\ \frac{\partial \ell}{\partial \mathbf{x}_k^p}\bigg|_{\{\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p\}}\\
Q_k &= \Delta t\ \frac{\partial^2 \ell}{\partial \mathbf{x}_k^p \partial \mathbf{x}_k^p}\bigg|_{\{\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p\}}\\
\mathbf{r}_k &= \Delta t\ \frac{\partial \ell}{\partial \mathbf{u}_k^p}\bigg|_{\{\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p\}}\\
R_k &= \Delta t\ \frac{\partial^2 \ell}{\partial \mathbf{u}_k^p \partial \mathbf{u}_k^p}\bigg|_{\{\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p\}}\\
P_k &= \Delta t\ \frac{\partial^2 \ell}{\partial \mathbf{x}_k^p \partial \mathbf{u}_k^p}\bigg|_{\{\bar{\mathbf{x}}_k^p, \bar{\mathbf{u}}_k^p\}}.
\end{aligned}
$$

2.1.3.2 The estimator

Since the states and measurements for the system in question are subject to noise, an estimator is required to infer the values of the states, as they are required to determine the control policy deviations. Li and Todorov suggest that a filter that is based on a sequence of fixed gains be used to determine the control policy in the inner loop of this iterative algorithm, but that in the outer loop moving horizon implementation an adaptive filter could be used [38].

For this fixed-gain inner loop filter, assuming that the system has an initial state estimate with a mean of $\hat{\mathbf{x}}_0$ and a covariance of Σ_0 and the unconditional means and covariances are defined as $m_k^{\hat{\mathbf{x}}} \triangleq E[\hat{\mathbf{x}}_k]$, $m_k^e \triangleq E[e_k]$, $\Sigma_k^{\hat{\mathbf{x}}} \triangleq E\left[\hat{\mathbf{x}}_k \hat{\mathbf{x}}_k^\top\right]$, $\Sigma_k^e \triangleq E\left[e_k e_k^\top\right]$, and $\Sigma_k^{\hat{\mathbf{x}}e} \triangleq E\left[\hat{\mathbf{x}}_k e^\top\right]$, the

optimal filter gain can be computed as

$$\begin{aligned}
\hat{\mathbf{x}}_{k+1} &= A_k\hat{\mathbf{x}}_k + B_k\pi_k + K_k\left(y_k - F_k\hat{\mathbf{x}}_k - E_k\pi_k\right) && (2.35)\\
K_k &= A_k\Sigma_k^e F_k^\top\left(F_k\Sigma_k^e F_k^\top + \mathcal{P}_k\right)^{-1} && (2.36)\\
m_{k+1}^{\hat{\mathbf{x}}} &= \left(A_k + B_kL_k\right)m_k^{\hat{\mathbf{x}}} + K_kF_km_k^e + B_kl_k && (2.37)\\
m_0^{\hat{\mathbf{x}}} &= \hat{\mathbf{x}}_0 && (2.38)\\
m_{k+1}^e &= \left(A_k - K_kF_k\right)m_k^e && (2.39)\\
m_0^e &= 0 && (2.40)\\
\Sigma_{k+1}^{\hat{\mathbf{x}}} &= \left(A_k + B_kL_k\right)\Sigma_k^{\hat{\mathbf{x}}}\left(A_k + B_kL_k\right)^\top + K_kF_k\Sigma_k^eA_k^\top\\
&\quad + \left(A_k + B_kL_k\right)\Sigma_k^{\hat{\mathbf{x}}e}F_k^\top K_k^\top + K_kF_k\Sigma_k^{e\hat{\mathbf{x}}}\left(A_k + B_kL_k\right)^\top\\
&\quad \left(\left(A_k + B_kL_k\right)m_k^{\hat{\mathbf{x}}} + K_kF_km_k^e\right)l_k^\top B_k^\top\\
&\quad B_kl_k\left(\left(A_k + B_kL_k\right)m_k^{\hat{\mathbf{x}}} + K_kF_km_k^e\right)^\top + B_kl_kl_k^\top B_k^\top && (2.41)\\
\Sigma_0^{\hat{\mathbf{x}}} &= \hat{\mathbf{x}}_0\hat{\mathbf{x}}_0^\top && (2.42)\\
\Sigma_{k+1}^e &= \left(A_k - K_kF_k\right)\Sigma_k^eA_k^\top + \mathcal{M}_k && (2.43)\\
\Sigma_0^e &= \Sigma_0 && (2.44)\\
\Sigma_{k+1}^{\hat{\mathbf{x}}e} &= \left(A_k + B_kL_k\right)\Sigma_k^{\hat{\mathbf{x}}e}\left(A_k - K_kF_k\right)^\top\\
&\quad + B_kl_k\left(m_k^e\right)^\top\left(A_k - K_kF_k\right)^\top && (2.45)\\
\Sigma_0^{\hat{\mathbf{x}}e} &= 0 && (2.46)
\end{aligned}$$

$$
\begin{aligned}
\mathcal{P}_k &= \sum_{i=1}^{n_v} \Big[\mathbf{d}_{i,k}\mathbf{d}_{i,k}^\top + \mathbf{d}_{i,k}\left(m_k^{\hat{\mathbf{x}}} + m_k^e\right)^\top \left(D_{i,k}^{\mathbf{x}}\right)^\top \\
&\quad + D_{i,k}^{\mathbf{x}}\left(m_k^{\hat{\mathbf{x}}} + m_k^e\right)\mathbf{d}_{i,k}^\top + \mathbf{d}_{i,k}\left(l_k + L_k m_k^{\hat{\mathbf{x}}}\right)^\top \left(D_{i,k}^{\mathbf{u}}\right)^\top \\
&\quad + D_{i,k}^{\mathbf{u}}\left(l_k + L_k m_k^{\hat{\mathbf{x}}}\right)\mathbf{d}_{i,k}^\top \\
&\quad + D_{i,k}^{\mathbf{x}}\left(\left(m_k^{\hat{\mathbf{x}}} + m_k^e\right) l_k^\top + \left(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{e\hat{\mathbf{x}}}\right) L_k^\top\right)\left(D_{i,k}^{\mathbf{u}}\right)^\top \\
&\quad + D_{i,k}^{\mathbf{u}}\left(l_k\left(m_k^{\hat{\mathbf{x}}} + m_k^e\right)^\top + L_k\left(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{\hat{\mathbf{x}}e}\right)\right)\left(D_{i,k}^{\mathbf{x}}\right)^\top \\
&\quad + D_{i,k}^{\mathbf{x}}\left(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{\hat{\mathbf{x}}e} + \Sigma_k^{e\hat{\mathbf{x}}} + \Sigma_k^e\right)\left(D_{i,k}^{\mathbf{x}}\right)^\top \\
&\quad + D_{i,k}^{\mathbf{u}}\left(l_k l_k^\top + l_k\left(m_k^{\hat{\mathbf{x}}}\right)^\top L_k^\top + L_k m_k^{\hat{\mathbf{x}}} l_k^\top + L_k \Sigma_k^{\hat{\mathbf{x}}} L_k^\top\right) \\
&\quad \times \left(D_{i,k}^{\mathbf{u}}\right)^\top \Big] \qquad (2.47) \\
\mathcal{M}_k &= \sum_{i=1}^{n_w} \Big[\mathbf{c}_{i,k}\mathbf{c}_{i,k}^\top + \mathbf{c}_{i,k}\left(m_k^{\hat{\mathbf{x}}} + m_k^e\right)^\top \left(C_{i,k}^{\mathbf{x}}\right)^\top \\
&\quad + C_{i,k}^{\mathbf{x}}\left(m_k^{\hat{\mathbf{x}}} + m_k^e\right)\mathbf{c}_{i,k}^\top + \mathbf{c}_{i,k}\left(l_k + L_k m_k^{\hat{\mathbf{x}}}\right)^\top \left(C_{i,k}^{\mathbf{u}}\right)^\top \\
&\quad + C_{i,k}^{\mathbf{u}}\left(l_k + L_k m_k^{\hat{\mathbf{x}}}\right)\mathbf{c}_{i,k}^\top \\
&\quad + C_{i,k}^{\mathbf{x}}\left(\left(m_k^{\hat{\mathbf{x}}} + m_k^e\right) l_k^\top + \left(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{e\hat{\mathbf{x}}}\right) L_k^\top\right)\left(C_{i,k}^{\mathbf{u}}\right)^\top \\
&\quad + C_{i,k}^{\mathbf{u}}\left(l_k\left(m_k^{\hat{\mathbf{x}}} + m_k^e\right)^\top + L_k\left(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{\hat{\mathbf{x}}e}\right)\right)\left(C_{i,k}^{\mathbf{x}}\right)^\top \\
&\quad + C_{i,k}^{\mathbf{x}}\left(\Sigma_k^{\hat{\mathbf{x}}} + \Sigma_k^{\hat{\mathbf{x}}e} + \Sigma_k^{e\hat{\mathbf{x}}} + \Sigma_k^e\right)\left(C_{i,k}^{\mathbf{x}}\right)^\top \\
&\quad + C_{i,k}^{\mathbf{u}}\left(l_k l_k^\top + l_k\left(m_k^{\hat{\mathbf{x}}}\right)^\top L_k^\top + L_k m_k^{\hat{\mathbf{x}}} l_k^\top + L_k \Sigma_k^{\hat{\mathbf{x}}} L_k^\top\right) \\
&\quad \times \left(C_{i,k}^{\mathbf{u}}\right)^\top \Big] \qquad (2.48)
\end{aligned}
$$

2.1.3.3 The backward pass

Approximating the cost-to-go function

Assuming a quadratic form for the cost-to-go function $v_k(\mathbf{x}_k,\ \hat{\mathbf{x}}_k)$ as

$$
v_k(\mathbf{x}_k,\ \hat{\mathbf{x}}_k) = \frac{1}{2}\mathbf{x}_k^\top S_k^{\mathbf{x}}\mathbf{x}_k + \frac{1}{2}\hat{\mathbf{x}}_k^\top S_k^{\hat{\mathbf{x}}}\hat{\mathbf{x}}_k + \mathbf{x}_k^\top S_k^{\mathbf{x}\hat{\mathbf{x}}}\hat{\mathbf{x}}_k + \mathbf{x}_k^\top \mathbf{s}_k^{\mathbf{x}} + \hat{\mathbf{x}}_k^\top \mathbf{s}_k^{\hat{\mathbf{x}}} + s_k, \qquad (2.49)
$$

the parameters $S_k^{\mathbf{x}}$, $S_k^{\hat{\mathbf{x}}}$, $S_k^{\mathbf{x}\hat{\mathbf{x}}}$, $\mathbf{s}_k^{\mathbf{x}}$, $\mathbf{s}_k^{\hat{\mathbf{x}}}$, and s_k can be computed recursively backwards in time for

$k < N$, as

$$
\begin{aligned}
S_k^{\mathbf{x}} &= Q_k + A_k^\top S_{k+1}^{\mathbf{x}} A_k + F_k^\top K_k^\top S_{k+1}^{\hat{\mathbf{x}}} K_k F_k + 2A_k^\top S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}} K_k F_k \\
&\quad + \sum_{i=1}^{n_w} \left(C_{i,k}^{\mathbf{x}}\right)^\top S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} + \sum_{i=1}^{n_v} \left(D_{i,k}^{\mathbf{x}}\right)^\top K_k^\top S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}} \qquad (2.50)\\
S_N^{\mathbf{x}} &= Q_N \\
S_k^{\hat{\mathbf{x}}} &= \left(A_k - K_k F_k\right)^\top S_{k+1}^{\hat{\mathbf{x}}} \left(A_k - K_k F_k\right) + L_k^\top H L_k + L_k^\top G^{\hat{\mathbf{x}}} \\
&\quad + \left(G^{\hat{\mathbf{x}}}\right)^\top L_k \\
S_k^{\hat{\mathbf{x}}} &= 0 \\
S_k^{\mathbf{x}\hat{\mathbf{x}}} &= F_k^\top K_k^\top S_{k+1}^{\hat{\mathbf{x}}} \left(A_k - K_k F_k\right) + A_k^\top S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}} \left(A_k - K_k F_k\right) + \left(G^{\mathbf{x}}\right)^\top L_k \\
S_N^{\mathbf{x}\hat{\mathbf{x}}} &= 0 \\
\mathbf{s}_k^{\mathbf{x}} &= \mathbf{q}_k + A_k^\top \mathbf{s}_{k+1}^{\mathbf{x}} + F_k^\top K_k^\top s_{k+1}^{\hat{\mathbf{x}}} + \left(G^{\mathbf{x}}\right)^\top l_k + \sum_{i=1}^{n_w} \left(C_{i,k}^{\mathbf{x}}\right)^\top S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} \\
&\quad + \sum_{i=1}^{n_v} \left(D_{i,k}^{\mathbf{x}}\right)^\top K_k^\top S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k} \\
\mathbf{s}_N^{\mathbf{x}} &= \mathbf{q}_N \\
\mathbf{s}_k^{\hat{\mathbf{x}}} &= \left(A_k - K_k F_k\right)^\top \mathbf{s}_{k+1}^{\hat{\mathbf{x}}} + L_k^\top H l_k + L_k^\top \mathbf{g} + \left(G^{\hat{\mathbf{x}}}\right)^\top l_k \\
\mathbf{s}_N^{\hat{\mathbf{x}}} &= 0 \\
s_k &= q_k + s_{k+1} + l_k^\top \mathbf{g} + \frac{1}{2} l_k^\top H l_k + \frac{1}{2} \left(\sum_{i=1}^{n_w} \mathbf{c}_{i,k}^\top S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} \right. \\
&\quad \left. + \sum_{i=1}^{n_v} \mathbf{d}_{i,k}^\top K_k^\top S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k} \right) \\
s_N &= q_N
\end{aligned}
$$

$$
\begin{aligned}
H &\triangleq R_k + B_k^\top \left(S_{k+1}^{\mathbf{x}} + S_{k+1}^{\hat{\mathbf{x}}} + 2S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}\right) B_k \\
&\quad + \sum_{i=1}^{n_w} \left(C_{i,k}^{\mathbf{u}}\right)^\top S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{u}} + \sum_{i=1}^{n_v} \left(D_{i,k}^{\mathbf{u}}\right)^\top K_k^\top S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{u}} \qquad (2.51)\\
\mathbf{g} &\triangleq \mathbf{r}_k + B_k^\top \left(s_{k+1}^{\mathbf{x}} + s_{k+1}^{\hat{\mathbf{x}}}\right) \\
&\quad + \sum_{i=1}^{n_w} \left(C_{i,k}^{\mathbf{u}}\right)^\top S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k} + \sum_{i=1}^{n_v} \left(D_{i,k}^{\mathbf{u}}\right)^\top K_k^\top S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k} \\
G^{\mathbf{x}} &\triangleq P_k + B_k^\top \left(S_{k+1}^{\mathbf{x}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}\right) A_k + B_k^\top \left(S_{k+1}^{\hat{\mathbf{x}}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}\right) K_k F_k \\
&\quad + \sum_{i=1}^{n_w} \left(C_{i,k}^{\mathbf{u}}\right)^\top S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} + \sum_{i=1}^{n_v} \left(D_{i,k}^{\mathbf{u}}\right)^\top K_k^\top S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}} \\
G^{\hat{\mathbf{x}}} &\triangleq B_k^\top \left(S_{k+1}^{\hat{\mathbf{x}}} + S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}\right) \left(A_k - K_k F_k\right).
\end{aligned}
$$

The proof can be found in [38].

Determining the control policy

If H is positive semi-definite, then the unconstrained optimal control deviation can be computed as

$$\mathbf{u}_k = -H^{-1}\left(\mathbf{g} + G\hat{\mathbf{x}}_k\right) = l_k + L_k\hat{\mathbf{x}}_k, \tag{2.52}$$

where G is given as

$$\begin{aligned} G \triangleq\ & P_k + B_k^\top \left(S_{k+1}^{\mathbf{x}} + S_{k+1}^{\hat{\mathbf{x}}} + 2S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}}\right) A_k \\ & + \sum_{i=1}^{n_w} \left(C_{i,k}^{\mathbf{u}}\right)^\top S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} + \sum_{i=1}^{n_v} \left(D_{i,k}^{\mathbf{u}}\right)^\top K_k^\top S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}}, \end{aligned} \tag{2.53}$$

and l_k and L_k are the control policy gains.

If H has negative eigenvalues, then the approximation of the cost-to-go function may become negative, and cost-to-go functions are always non-negative. To ensure that H always has non-negative eigenvalues, a regularized versions of H, or H and G, $\mathcal{H}$ and $\mathcal{G}$, are used to capture the second-order information in these matrices to compute the unconstrained optimal control deviation as

$$\mathbf{u}_k = -\mathcal{H}^{-1}\left(\mathbf{g} + \mathcal{G}\hat{\mathbf{x}}_k\right) = l_k + L_k\hat{\mathbf{x}}_k. \tag{2.54}$$

There are several options of how perform this regularization, and all use a positive regularization term λ. The first option [60] is to set

$$\mathcal{H} = H + \lambda I. \tag{2.55}$$

The second option is to regularize H and G through the quadratic cost-to-go terms ([60]) as

$$\begin{aligned} \mathcal{H} \triangleq\ & R_k + B_k^\top \left(S_{k+1}^{\mathbf{x}} + S_{k+1}^{\hat{\mathbf{x}}} + 2S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}} + \lambda I\right) B_k \\ & + \sum_{i=1}^{n_w} \left(C_{i,k}^{\mathbf{u}}\right)^\top S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{u}} + \sum_{i=1}^{n_v} \left(D_{i,k}^{\mathbf{u}}\right)^\top K_k^\top S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{u}}, \end{aligned} \tag{2.56}$$

$$\begin{aligned} \mathcal{G} \triangleq\ & P_k + B_k^\top \left(S_{k+1}^{\mathbf{x}} + S_{k+1}^{\hat{\mathbf{x}}} + 2S_{k+1}^{\mathbf{x}\hat{\mathbf{x}}} + \lambda I\right) A_k \\ & + \sum_{i=1}^{n_w} \left(C_{i,k}^{\mathbf{u}}\right)^\top S_{k+1}^{\mathbf{x}} C_{i,k}^{\mathbf{x}} + \sum_{i=1}^{n_v} \left(D_{i,k}^{\mathbf{u}}\right)^\top K_k^\top S_{k+1}^{\hat{\mathbf{x}}} K_k D_{i,k}^{\mathbf{x}}. \end{aligned} \tag{2.57}$$

The third option is to set

$$\mathcal{H} = H + \left(\lambda - \min(eig(H))\right) I, \tag{2.58}$$

where $\min(eig(H))$ is the minimum eigenvalue of H. Finally, a fourth option [38] is to set

$$\mathcal{H} = VDV^\top, \tag{2.59}$$

where $[V, D] = eig(H)$ is the eigenvalue decomposition of H, but the elements in the diagonal matrix D that are less than λ are replaced with λ before computing $\mathcal{H}$.

In order to have λ be as small as possible so that the algorithm converges quickly but also have λ increase quickly if H is not being positive definite, Tassa *et al.* present a quadratic modification schedule for the regularization term in [60]. Given a minimum vale for λ of $\lambda_{\min}$ (typically 10^{-6}) and a minimum modification factor Δ_0 (typically 2), λ is either increased with

$$\Delta \leftarrow \max(\Delta_0,\ \Delta\Delta_0) \tag{2.60}$$

$$\lambda \leftarrow \max(\lambda_{\min},\ \lambda\Delta) \tag{2.61}$$

or decreased with

$$\Delta \leftarrow \max\left(\frac{1}{\Delta_0},\ \frac{\Delta}{\Delta_0}\right) \tag{2.62}$$

$$\lambda \leftarrow \begin{cases} \lambda\Delta & \text{if } \lambda\Delta > \lambda_{\min}, \\ 0 & \text{if } \lambda\Delta < \lambda_{\min}. \end{cases} \tag{2.63}$$

The regularization parameter λ is increased if the backward pass fails due to H not being positive definite or if the line search fails to find a solution that improves upon the initial control trajectory. The regularization parameter λ is decreased when the line search finds an improved control trajectory or when the the algorithm terminates due to a small control deviation gradient. A parameter defining the maximum regularization term, $\lambda_{\max}$, is used for another stopping criteria.

Control constraints

As discussed in [61], there are two main options for introducing control constraints in iLQG; by using squashing functions or by solving a quadratic program that is subject to box constraints.

A squashing function, $\mathbf{s}(\mathbf{u})$ is an element-wise sigmoid with vector limits of

$$\lim_{\mathbf{u}\to-\infty} \mathbf{s}(\mathbf{u}) = \mathbf{b}_{\min} \tag{2.64}$$

$$\lim_{\mathbf{u}\to\infty} \mathbf{s}(\mathbf{u}) = \mathbf{b}_{\max} \tag{2.65}$$

that is introduced in the system dynamics as

$$\mathbf{x}_{k+1} = f\left(\mathbf{x}_k, \mathbf{s}(\mathbf{u_k})\right), \tag{2.66}$$

where $\mathbf{b}_{\min}$ and $\mathbf{b}_{\max}$ are the vector limits to be imposed on the control inputs. An example of a squashing function is

$$\mathbf{s}(\mathbf{u}) = \frac{\mathbf{b}_{\max} - \mathbf{b}_{\min}}{2} \tanh(\mathbf{u}) + \frac{\mathbf{b}_{\max} - \mathbf{b}_{\min}}{2}. \tag{2.67}$$

To keep $\mathbf{u}$ from taking on extreme values, cost terms should be imposed on both $\mathbf{u}$ and $\mathbf{s}(\mathbf{u})$.

The other option to introduce control constraints is by solving the quadratic program of minimizing the cost-to-go approximation from equation (2.34), subject to the box control constraints $\mathbf{b}_{\min}$ and $\mathbf{b}_{\max}$. The quadratic optimization problem to be solved is

$$\min_{\mathbf{u}_k} \quad \ell_Q\left(\mathbf{x}_k, \mathbf{u}_k\right) \tag{2.68}$$

$$\text{subject to} \quad \mathbf{b}_{\min} \leq \bar{\mathbf{u}}_k^p + \mathbf{u}_k \leq \mathbf{b}_{\max}. \tag{2.69}$$

This approach directly solves for the sequence of control actions that minimize the approximation of the total cost, by solving a sequence of these problems in a backward pass. For more details, please refer to [61].

2.1.3.4 The line search

Regardless of the method used to impose control constraints, the gain matrices l_k and L_k are the final result, such that the improved policy for the control deviations can be determined as shown in equation (2.52). As discussed previously, this optimization process can result in control deviations that are arbitrarily large and can therefore send the state trajectory outside of the region where the linear approximation to the system dynamics is reasonably accurate. To account for this, a line search is performed to sequentially reduce the improved control deviations until a solution is found that is estimated to cause a reduction in the total cost. This locally-linear policy is determined by performing a forward pass through the estimated system dynamics:

$$\hat{\mathbf{x}}_0^p = \bar{\mathbf{x}}_0^p \tag{2.70}$$

$$\hat{\mathbf{u}}_i^p = \bar{\mathbf{u}}_i^p + \alpha l_i + L_i\left(\hat{\mathbf{x}}_i^p - \bar{\mathbf{x}}_i^p\right) \tag{2.71}$$

$$\hat{\mathbf{x}}_{i+1}^p = f\left(\hat{\mathbf{x}}_i^p, \hat{\mathbf{u}}_i^p\right) \tag{2.72}$$

where α is a backtracking search parameter that is set to 1 and then sequentially reduced. If the line search fails to find a reduced cost solution, the regularization parameter is increased as shown in equations (2.62) and (2.63), and the control policy backward pass and line search forward pass are repeated until the algorithm converges to a locally optimal control policy.

2.1.3.5 Stopping criteria

Four stopping criteria are used to define convergence and control how long this iterative algorithm runs. First, if the gradient of the control deviations is less than a predefined threshold, the algorithm terminates. Second, if the positive reduction in the cost function is less than a predefined threshold, the algorithm also terminates. Third, if λ is greater than the predefined $\lambda_{\max}$ value, that algorithm is not able to regularize H and the algorithm terminates. Finally, if the number of iterations that the algorithm has completed exceeds a predefined maximum number of iterations, the algorithm also terminates.

2.1.3.6 Moving horizon implementation

iLQG can be implemented with a moving horizon approach as described for iLQR, but since there is noise in the system dynamics and the measurements, a filter must be used to obtain an estimate of the states. A system diagram for this approach is shown in Figure 2.1, where the iLQG block implements the iterative iLQG algorithm, which is referred to as the inner loop, while the outer loop contains the control signal being sent to the true system for a single time step, the state dynamics being sampled with a sampling period of T as well as the measurement and filtering steps. For each outer loop iteration, the measurements and the existing knowledge about the system can be used to update the estimate of the states with the use of a filter such as a sigma-point Kalman filter [47]. Here, where the iLQG algorithm is neither adaptive nor dual, the uncertain parameters $\hat{\mathbf{d}}$ are treated as constants, and an augmented constants vector can be formed by their concatenation.

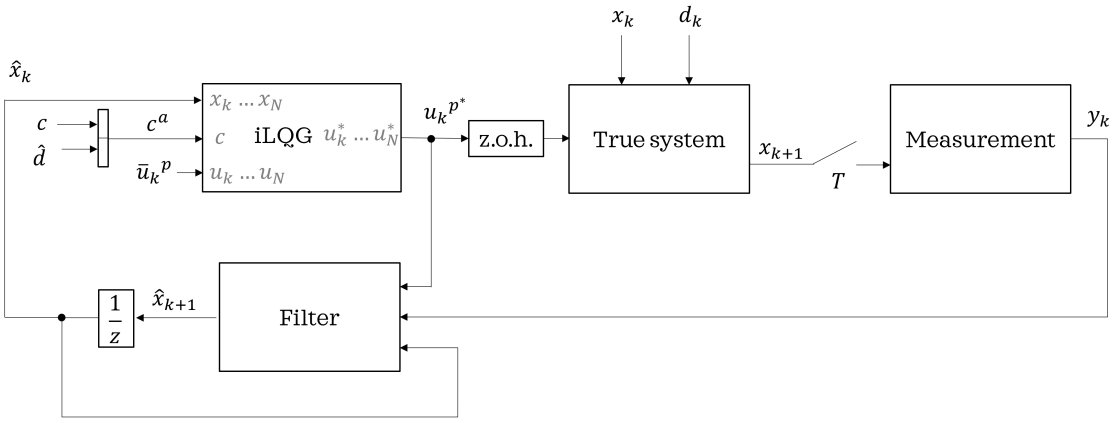


Figure 2.1: Closed-loop iLQG system diagram. Terms defined in the list of variables.

2.2 Adaptive control

Adaptive control is the control of systems with uncertain parameters. This uncertainty can be due to the cost or complexity of accurately measuring the parameters being high, or when the control scheme is to be applied to multiple similar systems that have different values for the parameters [16]. In [29], adaptive control methods are divided into four categories: gain scheduling, model-reference adaptive control, self-tuning adaptive control, and dual control.

The idea behind gain scheduling is to change the controller or its parameters based on pre-defined conditions. A series of local controllers are required to implement gain scheduling, which are each tuned for their operating range, and a method of switching between them. The selection of the local controller can be seen as an adaptive process.

A model reference adaptive controller adjusts the estimates of a model's parameters such that the tracking error of the plant converges to zero. Here, the control and adaptation laws are coupled, and these can be derived using Lyapunov theory such that stability can be shown. Although perfect tracking can be achieved, this condition does not imply that the parameter estimates have converged on their true values; a control task must have sufficient richness for parameter convergence [58].

In the self-tuning adaptive control method, the plant parameters are recursively estimated from the input-output data and then used by a controller as if they were the true plant parameters. Using the estimated value as if it were the true value is often called the certainty equivalence principle, allowing the design of the estimation process and controller to be

independent, unlike in model-reference adaptive control. With self-tuning adaptive control there are also no guarantees of parameter convergence without sufficient richness, and the design of the estimator and the controller being independent makes the stability of the system harder to prove [58].

Dual control is theoretically the ideal adaptive control method [29], and one of the major differences is that dual control takes into account the fact that uncertainties will be reduced in the future. This consideration of future information allows for a controller that can probe the plant and make control actions to reduce the parameter uncertainty in the future in a cautious way and focus on the most relevant parameters.

2.3 Dual control

Bayard *et al.* divide stochastic control policies into three classes: open-loop, feedback, and closed-loop [8]. Open-loop control policies do not use any process measurements and therefore no learning occurs. Feedback control policies use all measurements up to the current time step, and therefore learning can occur, but the learning is passive since the data generated is only due to performing the control task. Closed-loop control policies also use all measurements up to the current time step, but additionally, they anticipate that future measurements are going to be made in the future. Taking future measurements into account to determine the current control action allows planned, or active, learning to take place. As the control policy has two objectives; system regulation and identification, these stochastic control methods are known as dual controllers.

Due to the curse of dimensionality, Bellman's equations cannot be efficiently solved for problems of arbitrary size, and therefore the explicit and implicit approximations of dual control are necessary for its practical implementation. The explicit approximation involves modifying the cost function to elicit one or more of the dual behaviours of probing, caution, and selectiveness. The implicit approximation on the other hand elicits these dual behaviours through a method that allows the control algorithm to consider the impact of probing actions on future costs. Adding terms to the cost function with the explicit approximation requires an explicit trade-off between the control objectives and the system identification, while with the implicit approximation, the controller can balance these dual objectives without additional information.

While it is simple to add extra terms to the cost function to elicit the dual features of caution, probing, and/or selectiveness, this approach fixes the value of system information relative to the

minimization of the rest of the cost function. Even with methods that vary the value of system information based on a specific measure, explicit approaches are likely to overvalue or undervalue system information at different points in time compared to implicit approaches. In implicit approaches, the value of system information is determined through the consideration of its impact on the unmodified cost function over the control horizon. For this reason, implicit dual approximations can give better results than explicit approaches [8].

2.3.1 Explicit dual control

As the explicit dual approach is manifested through the cost function, many control approaches have been used as a basis for explicit dual control. In [11], a self-tuning adaptive controller for linear systems was modified to include caution and probing terms. In [44], a linear MPC approach was modified to include the dual behaviour of probing by adding a sufficient richness condition such that each control input considers the control inputs that have recently been made, allowing for persistent excitation. A similar approach was used in [24] for linear single input single output systems and was extended in [22] to include disturbances, and in [31] to multiple input multiple output systems with unmeasured stochastic disturbances. A robust tube-based MPC was combined with a partially closed-loop stochastic MPC to give the cautious and probing features of dual control in an explicit way in [17], and was later extended to use an unscented Kalman filter for state estimation in [18]. In [71], a robust invariant set-based MPC method was applied to linear systems with additive and parametric uncertainties using MPC that was formulated to select the controls that maximize probing and regulation objectives. In [32, 33], explicit dual NMPC is developed using sequential optimal experimental design. Due to the relative ease of modifying a cost function to encourage dual behaviours, there have been more publications on explicit dual control than the implicit approximation.

Lastly, and very relevant to the present work, in 1976, Bar-Shalom and Tse presented an explicit dual control algorithm known as wide-sense dual control in [5]. Wide-sense dual control is a dual version of differential dynamic programming, which was developed by Mayne in 1966 [45], and Bar-Shalom and Tse appear to not have known about Mayne's prior work. Bar-Shalom and Tse use a partial certainty equivalence assumption to treat the estimates of the states and parameters as their true values, which is only suitable for the additive noise that they use. This assumption makes wide-sense dual control essentially an explicit dual fully observable iLQG with a second-order approximation of the dynamics instead of a first-order approximation. Li and

Todorov developed iLQR in 2004 [36] and iLQG in 2006 [37], and they appear to not have known about Bar-Shalom and Tse's prior work, basing their work on Mayne's differential dynamic programming. In 2010, Theodorou, Tassa, and Todorov [65] derived stochastic differential dynamic programming for multiplicative noise, which, other than not being dual, is a more general version of wide-sense dual control. For more details on these comparisons, see Appendix A.

Comparing wide-sense dual control with fully observable iLQG, other than the use of a second-order approximation of the dynamics and the use of an augmented state vector, there is a difference in the scalar term of the quadratic approximation of the cost-to-go function. Using different notation, Bar-Shalom and Tse list the scalar component of the assumed quadratic form of the cost-to-go function as

$$\begin{aligned} g_0(j) &= g_0(j+1) - \frac{1}{2} H_{0,u}^{\top}(j) \mathcal{H}_{0,uu}^{-1} H_{0,u}(j) \\ &+ \frac{1}{2} Tr\left[\mathbf{K}_o(j+1)\mathbf{Q}(j) + \mathcal{A}_{0,xx}(j)\mathbf{\Sigma}_0(j|j)\right], \end{aligned} \tag{2.73}$$

which can be written in the notation used by Li and Todorov in [38] as

$$s_k = s_{k+1} - \frac{1}{2} g^{\top} H^{-1} g + \frac{1}{2} Tr\left[\mathbf{S}_{k+1}\Sigma^f + GH^{-1}G\Sigma_k^{\boldsymbol{x}}\right], \tag{2.74}$$

where Σ^f is the covariance of the additive Gaussian process noise and $\Sigma_k^{\boldsymbol{x}}$ is the state covariance along the nominal trajectory.

These terms provide an explicit addition to the cost function used for differential dynamic programming [61] to balance the notion of caution and probing. They go beyond the terms needed for non-dual control, as evidenced by the fact that the additive noise should have no impact on the control policy for differential dynamic programming [65]. For comparison, the non-dual scalar term of the quadratic approximation of the cost-to-go function is

$$s_k = s_{k+1} - \frac{1}{2} g^{\top} H^{-1} g. \tag{2.75}$$

For context, if Bar-Shalom and Tse had added multiplicative noise, the equation would have gained the term $\sum_{i=1}^{n_w} \mathbf{c}_{i,k}^{\top} S_{k+1}^{\mathbf{x}} \mathbf{c}_{i,k}$, and if the states were not fully observable, it also gains the term $\sum_{i=1}^{n_v} \mathbf{d}_{i,k}^{\top} K_k^{\top} S_{k+1}^{\hat{\mathbf{x}}} K_k \mathbf{d}_{i,k}$ [38], as shown in our implementation in equation (2.50). Bar-Shalom and Tse point to their terms $\frac{1}{2} Tr\left[\mathbf{S}_{k+1}\Sigma^f + GH^{-1}G\Sigma_k^{\boldsymbol{x}}\right]$, as implementing the dual features of caution and probing respectively [5], and this makes sense because changing the

coefficient in front of either of those terms will explicitly affect the balance of caution versus probing. The inclusion of that term accordingly makes their wide-sense dual control approach an explicit dual control algorithm that can accommodate additive noise.

Bar-Shalom and Tse's work is extended in [29] by estimating the system dynamics using a parametric, Gaussian process and neural network regression. Like Bar-Shalom and Tse's work on which it is based, this control method is explicitly dual and only applicable to additive noise. One application shows how dual NMPC can be applied to control the climate of a building, and this example is also detailed in [28]. Two future research directions listed were time-scale separation and reformulating this implicit dual NMPC method without the dynamic programming portion to make it solely NMPC-based. Although this control strategy was formulated considering unknown time-varying system parameters, the stability of the controller and providing guarantees on the parameter estimation were not considered.

2.3.2 Implicit dual control

As with the work on explicit dual control, many of the implicit dual control publications have been based on MPC, but several have taken other approaches. These works also make different assumptions on the uncertain parameters and use different estimation approaches.

Multi-stage NMPC is used as an implicit approximation to dual control in [63], where the unknown system parameters were assumed to be bounded, parametric, and time-invariant. Here, the Fisher information matrix is used to estimate the future reduction in uncertainties, but the least squares estimate of the uncertainties is assumed to be constant. The parameter sensitivities are included in the scenario tree in the same way as the states, allowing the controller to make decisions based on probing for information on specific parameters. More involved estimation schemes for the uncertainties and their bounds are mentioned, including guaranteed parameter estimation. This dual control method is then applied to the control of a simulated chemical batch reactor. This work was extended in [62], and the assumption that the least squares estimate has converged to the true value of the uncertainty is removed. This assumption is replaced by an over-approximation factor to estimate the future changes in the point estimation. Here, guaranteed parameter estimation is mentioned as a possible future extension of this work, but was not explored. This approach is used as a basis of comparison throughout this work and is therefore explained in more detail in the following section.

In [21], a multi-stage implicit dual MPC method is presented where the unknown system

parameters are assumed to be time-invariant. Here, the future measurements at each stage are used to update the states and parameters for all of the other scenarios in that stage using an ensemble Kalman filter. A similar approach was used in [59], except that an Unscented Kalman filter is used instead of an ensemble Kalman filter, and the model parameters are not updated (and therefore this method is not dual).

Using a policy iteration approach and particle filtering for the nonlinear estimation problem an implicit dual control method is developed in [8]. This strategy is applied to a generic system with process and measurement noise, and the computations are performed in a novel H-block format. This control scheme is then applied to the simulation of a linear pendulum with uncertain length, mass, and control gain.

In [56], the finite-horizon implicit dual control problem is made tractable for problems of reasonable dimensions by approximating the system dynamics by a partially observable Markov decision process. This control method is then applied to a health care decision-making process.

One method that has been developed to implement implicit dual control is multi-stage NMPC. Multi-stage NMPC considers a branching network of scenarios, where each branch represents the system's predicted response for a single realization of the uncertainties. This scenario tree is illustrated in Figure 2.2, where $\mathbf{x}$ represents the system states, $\mathbf{u}$ represents the control actions, and $\mathbf{d}$ represents the discrete realizations of the uncertainties with superscripts indicating the scenario number and subscripts indicating the time index into the future. It is important to note that the control actions that share a node are equal, and this requirement is known as the non-anticipatory constraint. The reduction in the uncertainties due to future measurements is estimated for each time step in a robust horizon, and this reduction is reflected in the selection of the uncertainty values upon which the subsequent scenarios are based. As this process causes the number of scenarios to grow exponentially, the robust horizon is usually chosen to be less than the control horizon, and the uncertainties after that point are assumed to be equal to the nominal value.

The optimization problem that is solved is to determine the control actions (over the control horizon) that minimize the sum of the costs of the scenarios over the prediction horizon. To ensure that none of the scenarios are excited more than necessary, the selected control action is cautious when uncertainties are high. Probing is also introduced as small control actions that may increase a scenario's cost initially but can lead to larger reductions in future costs associated with the uncertainty. In the same manner, selectiveness is introduced, as probing actions that have a higher "return on investment" will be prioritized, and therefore the most important uncertainties

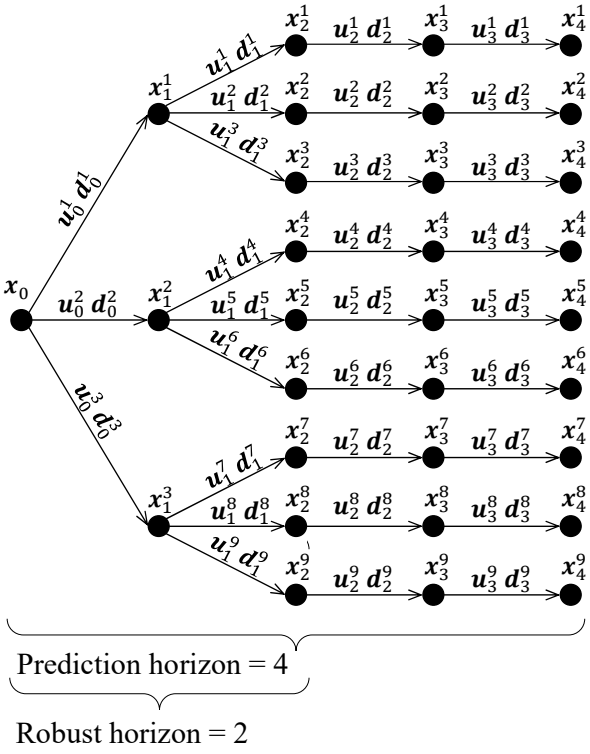


Figure 2.2: The tree of scenarios considered in multi-stage NMPC [62].

will be reduced. The multi-stage NMPC method relies on predicting the future reductions in uncertainties for given control actions, and even when representing uncertainties by a discrete set of realizations, the number of scenarios grows exponentially and must be limited.

The selection of the discrete realizations of the uncertainties impacts the controller's robustness and computational burden. For linear systems, using the minimum, maximum, and nominal values of the uncertainties (assuming they are bounded) keeps the number of scenarios per branch low and can be shown to be "usually" robust [63]. For nonlinear systems, this approach is not robust but has shown good results in practice [63]. In [12], uncertainties in a stochastic NMPC method were propagated using the unscented Kalman filter, but this control approach was neither dual nor multi-stage. Dual multi-stage NMPC with the unscented Kalman filter method was shown to be robust in [39].

Present implementations of dual and non-dual multi-stage NMPC [1, 41, 42, 46, 64] use IPOPT, an interior-point optimization routine for large-scale problems [70], and CasADi, an open-source numerical optimization framework designed to solve optimal control problems [3], from within Python, MATLAB, or Octave. To symbolically calculate the confidence regions that are needed to set up and solve the dual multi-stage NMPC problem, the eigenvectors and

eigenvalues of the parameter covariance matrix are required. Currently, CasADi's symbolic eigenvector and eigenvalue functions only support matrices up to 3×3 in size, limiting the number of uncertain parameters to three. The code from [64] that was used in this work was only able to handle a maximum of two parameters. Therefore in this work, comparing our dual iLQG approach with the dual MS-SP-NMPC approach was only done on systems with two parameters.

2.4 Summary

In this chapter, the mathematical background of dual iLQG was introduced, including LQR, iLQR, and iLQG, as well as adaptive and dual control. The explicit and implicit approximations of dual control were explained, and past work using these approaches were discussed. In the next chapter, adaptive and dual extensions of iLQG will be described. After these methods are presented, they will be compared to MS-SP-NMPC and their robustness to time-varying parameters and unmodelled dynamics will be investigated. Finally, the importance of compensating for multiplicative noise will be demonstrated.

Chapter 3

Dual iLQG

3.1 Motivation

iLQG is a powerful control technique due to its ability to handle nonlinear and stochastic systems with multiplicative noise but has not been extended to handle systems with uncertain parameters in either an adaptive or dual manner. Dual iLQG represents a fast (due to the linearization of the system) and feasible (due to working with derivatives about a nominal state-control trajectory) solution to the implicit dual control of small and large systems while avoiding Bellman's curse of dimensionality.

3.2 Adaptive iLQG

To extend iLQG to uncertain systems in an adaptive manner, two changes are made to the closed-loop iLQG approach shown in Figure 2.1. First, the initial estimates of the uncertain parameters are provided to the iLQG inner loop as constants. Second, in the outer loop, the parameters are estimated along with the states in the filter. This is done by concatenating the uncertain states and parameters together into a single augmented state vector,

$$\mathbf{x}^{p^a} = \begin{bmatrix} \mathbf{x}^p \\ \mathbf{d}^p \end{bmatrix} \tag{3.1}$$

where $\mathbf{x}^a$ is the augmented state vector and $\mathbf{d}^p$ is the uncertain parameter trajectory, and their respective covariance matrices are also combined in a block diagonal manner,

$$\Sigma^{\mathbf{x}^a} = \begin{bmatrix} \Sigma^{\mathbf{x}} & \mathbf{0} \\ \mathbf{0} & \Sigma^{\mathbf{d}} \end{bmatrix} \tag{3.2}$$

where $\Sigma^{\mathbf{x}^a}$ is the augmented covariance matrix and $\Sigma^{\mathbf{d}}$ is the parameter covariance matrix. The augmentation of the state vector with the parameters for the closed-loop filter is an approach that is known as joint simultaneous state and parameter estimation [51]. To pass the parameters to the inner loop iLQG algorithm, an augmented constants vector $\mathbf{c}^a$ is created similar to the augmented state vector, through the concatenation of the constants and the parameters,

$$\mathbf{c}^a = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}. \tag{3.3}$$

This approach allows adaptive iLQG to update its parameter estimates in the outer loop after getting new measurements and pass them to the inner loop iLQG algorithm as constants, which is known as the certainty equivalence principle. Critically, because the parameters are treated as constants in the inner loop iLQG algorithm, the uncertainty associated with the parameters does not impact the determination of the control policy. These changes are shown in the system diagram for closed-loop adaptive iLQG in Figure 3.1b.

Although adaptive iLQG controllers have been developed, they are generally focused on identifying the system dynamics as a whole as opposed to a set of parameters in an existing model of the system. In [35], an adaptive iLQG algorithm was developed where the entire Gaussian Process system dynamics were adapted using Gaussian Process regression. The system dynamics are also adapted in [50] using Locally Weighted Projection Regression.

3.3 Dual iLQG

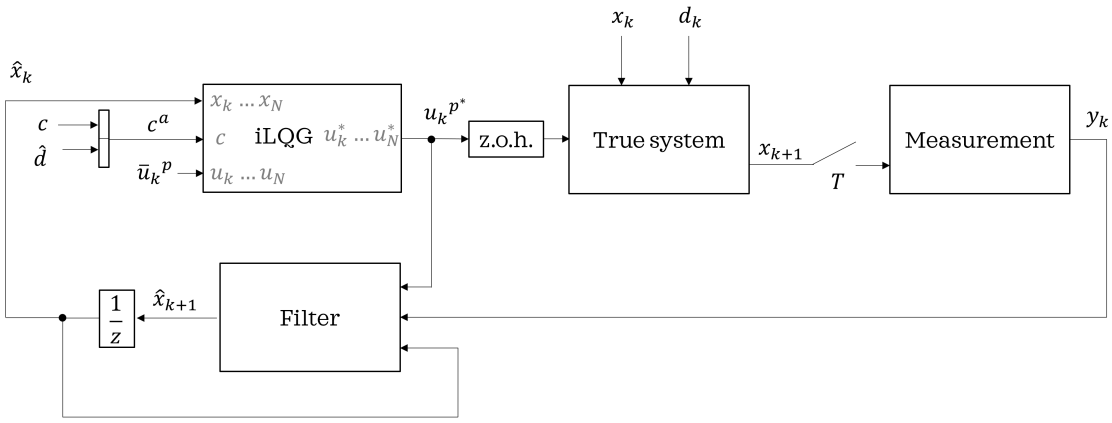
To extend this adaptive iLQG approach to be dual, the uncertainty associated with the parameters must influence the control policy. Therefore, instead of treating the parameters as constants in the inner loop iLQG algorithm, the parameters are treated as states and an augmented state vector is formed as shown in equation (3.1). An augmented state covariance is

also created as shown in (3.2), and the augmented state dynamics must be formed as

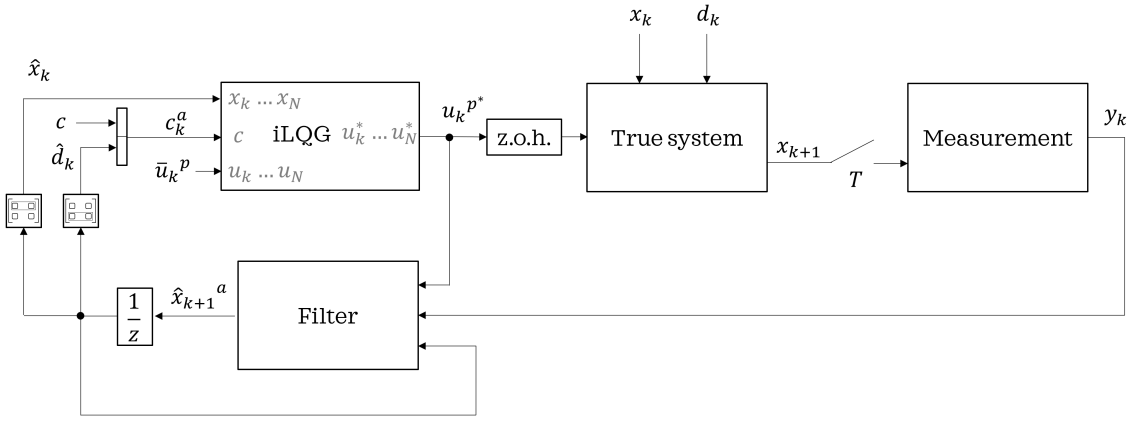
$$\mathrm{d}\mathbf{x}^{p^a} = \begin{bmatrix} \mathbf{f}(\mathbf{x}^{p^a}, \mathbf{u}^p) \\ \mathbf{f}_d(\mathbf{x}^{p^a}, \mathbf{u}^p) \end{bmatrix} \mathrm{d}t = \mathbf{f}^a(\mathbf{x}^{p^a}, \mathbf{u}^p)\mathrm{d}t \tag{3.4}$$

where $\mathbf{f}_d(\mathbf{x}^{a^p}, \mathbf{u}^p)$ is the parameter dynamics and $\mathbf{f}^a(\mathbf{x}^{a^p}, \mathbf{u}^p)$ is the augmented system dynamics. The inclusion of the parameter dynamics makes dual iLQG able to handle time-varying parameters. These changes are represented in the system diagram in Figure 3.1c. In this way, the iLQG algorithm treats the parameters as unmeasured states and allows the control algorithm to predict how changes to the inputs and states can result in future reductions in the parameter uncertainty, through the backward pass of the cost-to-go function, to lower the total cost of the control trajectory. The parameter uncertainty influences the control actions through the inner loop Kalman filter gain, which impacts the estimates of the augmented state vector and the cost function at each time step. By calculating the derivatives of the cost function at each time step, dual iLQG can identify changes to the inputs that can decrease parameter uncertainties while also decreasing the total cost of the control trajectory.

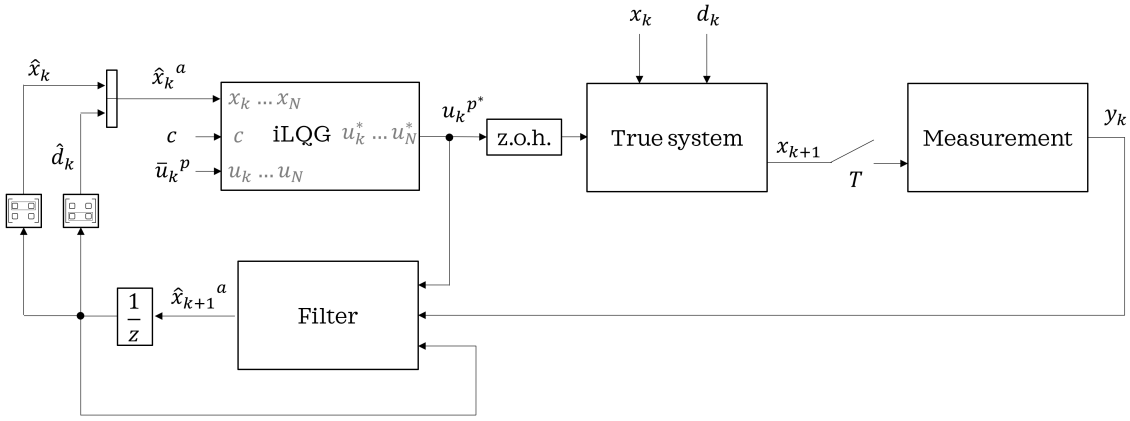
Unlike Wide-Sense dual control where the probing and caution terms can be explicitly pointed to in the mathematical formulation, in dual iLQG these features are an emergent property of the derivative-based calculations. Additionally, if one wanted to change the balance of the features of probing and cautiousness in Wide-Sense dual control, additional coefficients could be introduced in front of each of the terms to weight their relative importance. The same cannot be said for dual iLQG. If cautious changes to a nominal control trajectory ultimately reduces the expected cost, dual iLQG will make these changes, even without explicit terms in the mathematical formulation to encourage this behaviour. Similarly, if changes to a nominal control trajectory that probe for additional information on uncertain parameters ultimately reduces the expected cost, dual iLQG will make these changes, even without explicit terms in the mathematical formulation to encourage this behaviour. And lastly, if selectively focusing probing actions on one set of uncertain parameters ultimately reduces the expected cost, dual iLQG will make these changes, even without explicit terms in the mathematical formulation to encourage this behaviour. That is of course to say that these features of dual control are implicit in dual iLQG's goal of optimizing the control trajectory to minimize the cost function.



(a) Closed-loop iLQG system diagram.



(b) Closed-loop adaptive iLQG system diagram.



(c) Closed-loop dual iLQG system diagram.

Figure 3.1: Flowcharts for the three variations of iLQG discussed in this work.
Note: Figure 3.1a is the same as Figure 2.1 and is shown here for ease of comparison. Terms defined in the list of variables.

3.3.1 Seeding approach

Due to the dual iLQG approach being locally optimal, a seeding method was incorporated to initialize the optimization problem with different sets of initial conditions. In the iLQG algorithm, the nominal control trajectory is the variable that is used to initialize the optimization and is iteratively improved upon through successive runs of the inner loop iLQG algorithm [36]. By running multiple instances of a single inner loop run with different initial control trajectory seeds, a larger control space is explored, allowing a better solution to be found.

Although this seeding process increases the performance of the dual iLQG algorithm, it linearly increases the computation time for each time step that it is used. For applications where the computational time required for the dual iLQG algorithm is much shorter than the desired control update time for the true dynamic system, seeding could take place at every time step. Otherwise, significant performance increases can still be gained by having the seeding process only take place during the first time step.

3.3.2 Moving horizon approaches

Two different moving horizon options were coded into the dual iLQG algorithm; shrinking and rolling horizon approaches. In the shrinking horizon approach, the controller simulates the system and solves for the control actions for the entire time horizon, from the present time step to the final time step. After the inner loop determines the optimal control policy for this time horizon and implements the first control action, the time horizon is reduced by a single time step and the inner loop of the algorithm is reinitialized. This process repeats until the final time step is reached, meaning that the calculation effort for each time step decreases as the algorithm progresses.

The rolling horizon approach on the other hand has a fixed time horizon, such that after each inner loop of the algorithm is complete, the next outer loop iteration is initialized by shifting this fixed time horizon one step forward, dropping the present time step and adding a new time step on the end. Once the final time step is reached, although the controls and system responses have been determined for future time steps, they are not considered in the results. In this way, the calculation effort for each time step stays the same as the algorithm progresses.

3.4 Comparison with dual MS-SP-NMPC

To compare the dual iLQG approach with the dual multi-stage sigma-point nonlinear MPC (MS-SP-NMPC) approach described in [64], the linear system from that paper was used. The dynamic model of the system is

$$\begin{aligned} \dot{x}_1 &= d_1 u_1 - d_2 \\ \dot{x}_2 &= d_2 u_2 \end{aligned} \tag{3.5}$$

where $\mathbf{x} = [x_1, x_2]^\top$ is the state vector, $\mathbf{u} = [u_1, u_2]^\top$ is the control input vector, and $\mathbf{d} = [d_1, d_2]^\top$ is the vector of the uncertain parameters. The true and estimated initial values of the state vector are both $[0, 0]^\top$ and the states are unconstrained, and the control inputs are also unconstrained. The true value of the constant parameter vector is $[125, 50]$, while its initial estimate is $[100, 30]$ with a covariance matrix Σ^d of $[8100, 4500; 4500, 5625]$. This system is simulated over 6 time steps that are 0.05 seconds long. Both states are measured with a noise scaling factor $\mathcal{G}$ of 10^{-2}, and as the MS-SP-NMPC does not consider the states to be uncertain, the noise scaling factors of $\mathcal{F}$ for dual iLQG were set to 10^{-15} for both the state and parameter dynamics. The cost function is given as

$$J(\mathbf{x}, \mathbf{u}) = (x_1 - 5)^2 + 10^{-4} u_2^2. \tag{3.6}$$

This problem is contrived to demonstrate the advantage of dual control over adaptive control. In an adaptive optimal control method, the cost function can be minimized by maintaining u_2 at zero and driving x_1 to 5 using u_1, as u_2 does not influence x_1. A dual control method on the other hand will temporarily use u_2 to reduce the uncertainty associated with d_2, even though it will incur a cost, to reduce the overall cost.

Adaptive MS-SP-NMPC, dual MS-SP-NMPC, and dual iLQG were compared using this system, and the resulting controls are shown in Figure 3.2. Starting with u_2, it can be seen that the adaptive MS-SP-NMPC maintains u_2 at zero as expected. The dual MS-SP-NMPC has a non-zero value of u_2 for the first time step only, while the magnitude of u_2 for the dual iLQG controller gradually steps down to zero over four time steps.

Figure 3.3 shows the resulting states for each of these controllers, the dual and adaptive MS-SP-NMPCs are shown to undershoot the tracking reference of $x_1 = 5$ by the end of the first time step, instead reaching 3.1 and 2.9 respectively, while the dual iLQG controller overshot to 5.6. After this point, both dual controllers tracked the state reference with only small errors,

while the adaptive MS-SP-NMPC took until the last time step to reach a similar performance.

These controls result in both of the dual controllers accurately estimating the true parameter values of 125 and 50 after the first time step, as shown in Figure 3.4, while the adaptive MS-SP-NMPC did not reach a similar level of accuracy until the last time step. Additionally, the dual iLQG controller's estimate of the uncertain parameters was maintained near the true values, while the dual MS-SP-NMPC's estimates varied.

The cumulative cost at each time step for these controllers is shown in Figure 3.5, with totals of 0.4, 3.61, and 4.65 for the dual iLQG, dual MS-SP-NMPC, and adaptive MS-SP-NMPC respectively. The dual iLQG algorithm took an average of 3.5 seconds to run, while the adaptive iLQG algorithm took 0.4 seconds, the adaptive MS-SP-NMPC algorithm took 3.0 seconds and the dual MS-SP-NMPC algorithm took 75 seconds.

In this linear example, the dual iLQG controller significantly outperformed dual MS-SP-NMPC by being less cautious with its initial value for u_1 and by continuing to use u_2 after the first time step to improve its estimate of the uncertain parameters.

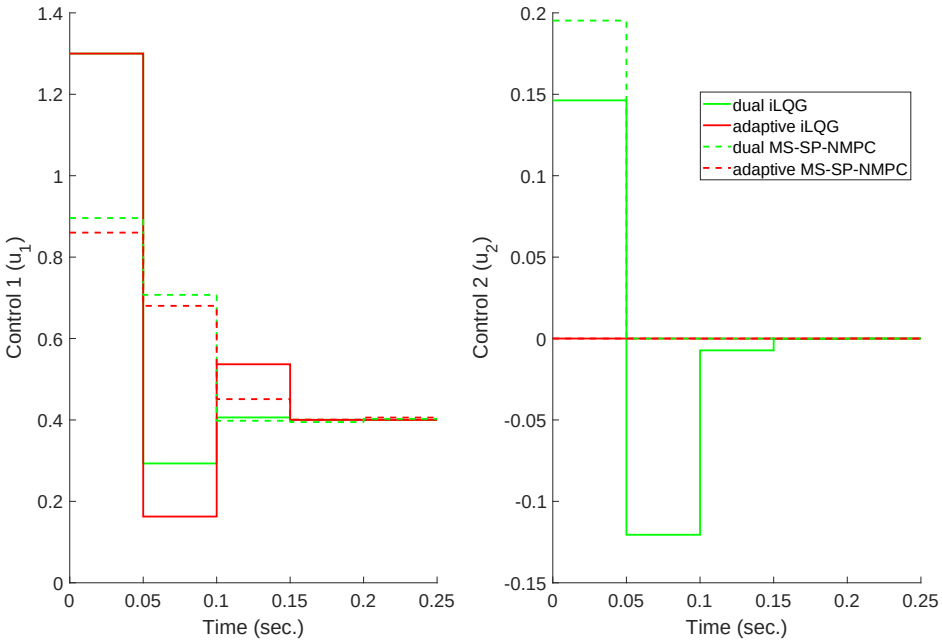


Figure 3.2: Control comparison between dual and adaptive iLQG and MS-SP-NMPC on linear example.

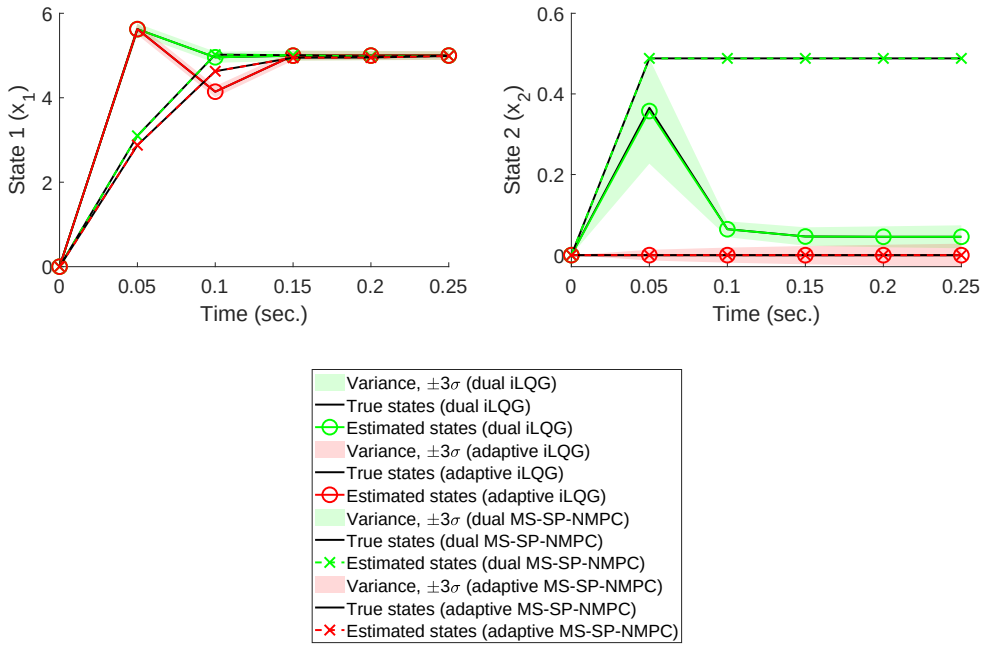


Figure 3.3: State comparison between dual and adaptive iLQG and MS-SP-NMPC on linear example.

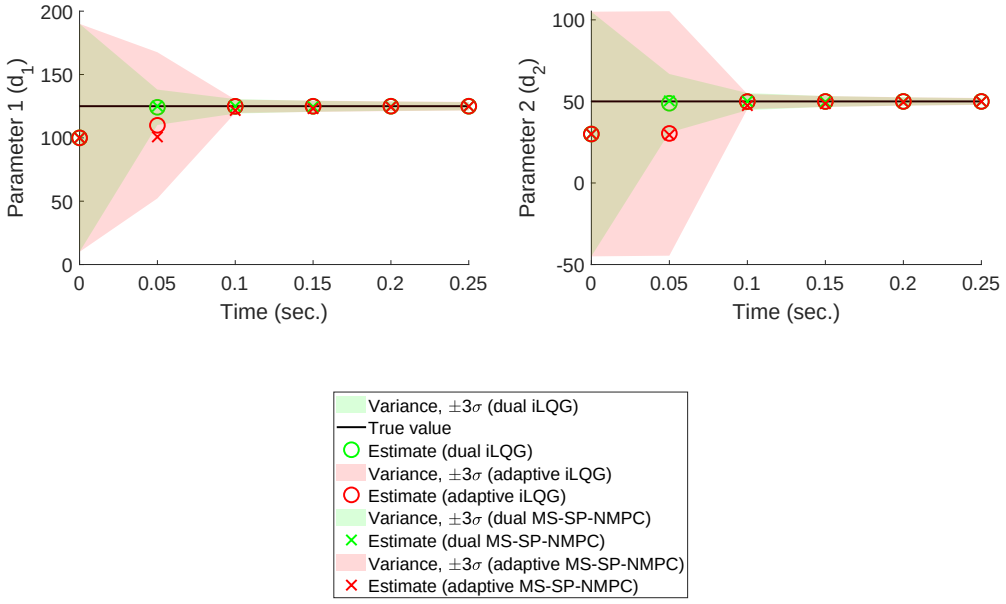


Figure 3.4: Parameter comparison between dual and adaptive iLQG and MS-SP-NMPC on linear example.

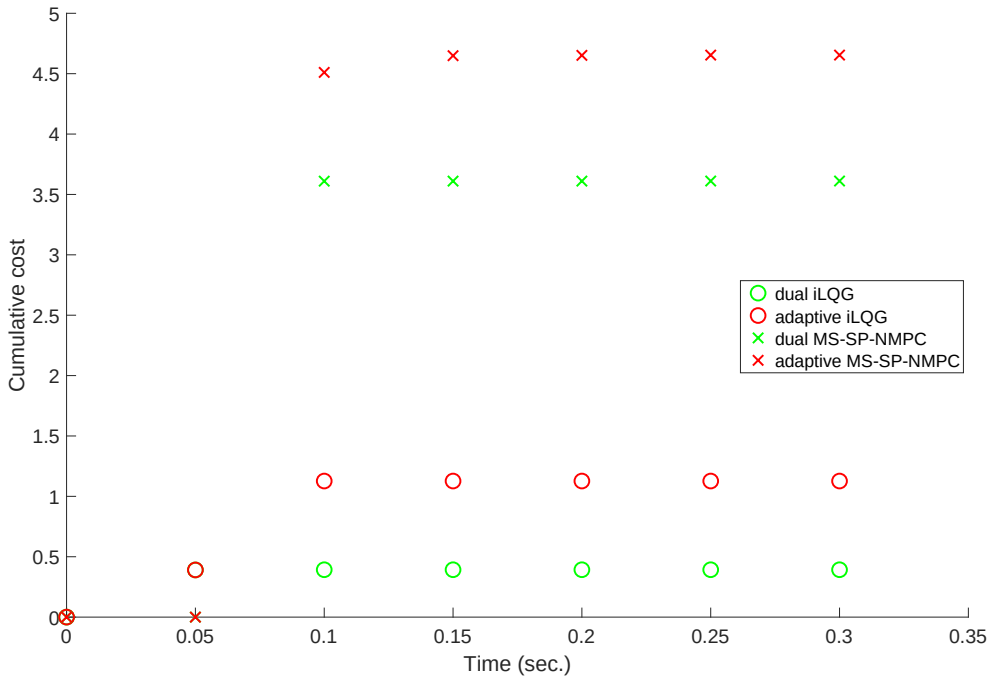


Figure 3.5: Cost comparison between dual and adaptive iLQG and MS-SP-NMPC on linear example.

3.5 Robustness to time-varying parameters

In many applications, parameters that are modelled as constant may vary over time. The ability of a controller to achieve the desired objective in the face of uncertain parameter dynamics can be described as the controller's robustness to time-varying parameters. In this section, the previous linear example is modified to have time-varying parameters to explore dual iLQG's robustness to time-varying parameters.

For this example, the system dynamics are the same as shown in (3.5), and the initial state and parameter estimates also are the same and both the states and the controls remain unconstrained. The initial true value of the parameters is the same as before, but the true parameters vary with dynamics of

$$\dot{d}_1 = -2.5 + 25\sin(5t) \tag{3.7}$$

$$\dot{d}_2 = 5 - 25\sin(5t), \tag{3.8}$$

while the controller believes that the parameters remain constant over time. Although the time step of 0.05 seconds was kept the same as the previous example, instead of simulating the system

over six steps, twenty time steps were used to show the impact of the time-varying parameters. A rolling horizon approach was used, with a horizon length of 6 time steps. The measurement noise scaling factors $\mathcal{G}$ were kept at 10^{-2}, but the noise scaling factor $\mathcal{F}$ associated with the state and parameter dynamics was increased to 5^{-1} from 10^{-15}. The cost function also remained the same, as shown in equation (3.6).

As would be expected from the results of the previous example, the dual controller used the second control action to gain information about the second parameter while the adaptive controller's second control action remained near zero, as shown in Figure 3.6. In terms of the first control action, both controllers had the same value for the first time step, after which they had similar but different values.

In Figure 3.7, both controllers can be seen to maintain the first state near 5 according to the cost function. Other than the second time step, adaptive iLQG appeared to track the desired reference at least as well as dual iLQG on average.

Figure 3.8 shows each iLQG algorithm's estimates of the time-varying parameters. In the first several time steps, the dual controller quickly converged to the true parameter values, while the adaptive iLQG controller took a couple of extra time steps to converge to the true value of the second parameter, and had a small offset error in its estimate of the first parameter. The dual controller's estimates for the parameters were better than the adaptive controller's, and both controllers were better able to estimate the second parameter.

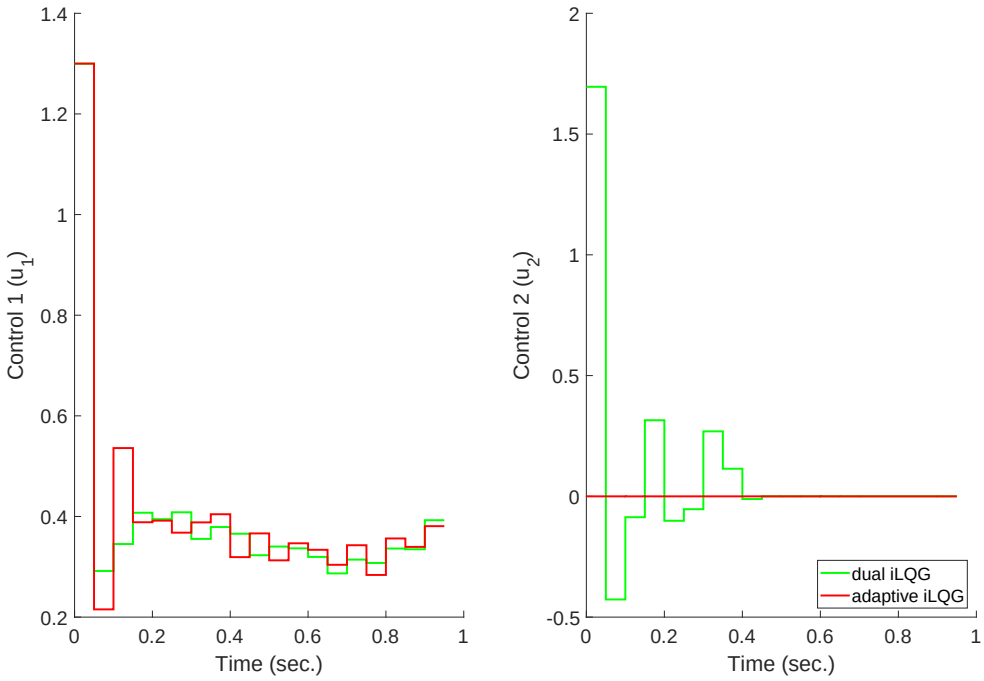


Figure 3.6: Control comparison between dual and adaptive iLQG on time-varying parameters example.

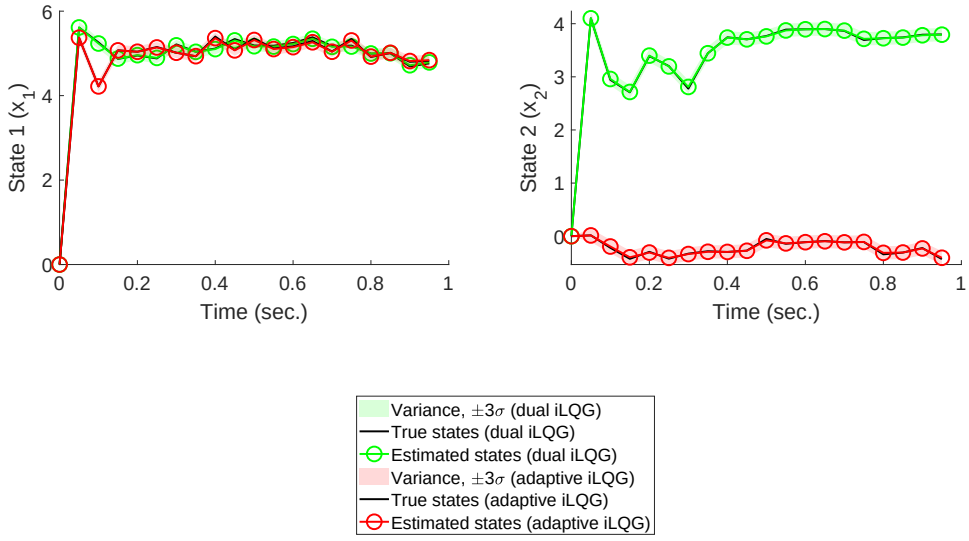


Figure 3.7: State comparison between dual and adaptive iLQG on time-varying parameters example.

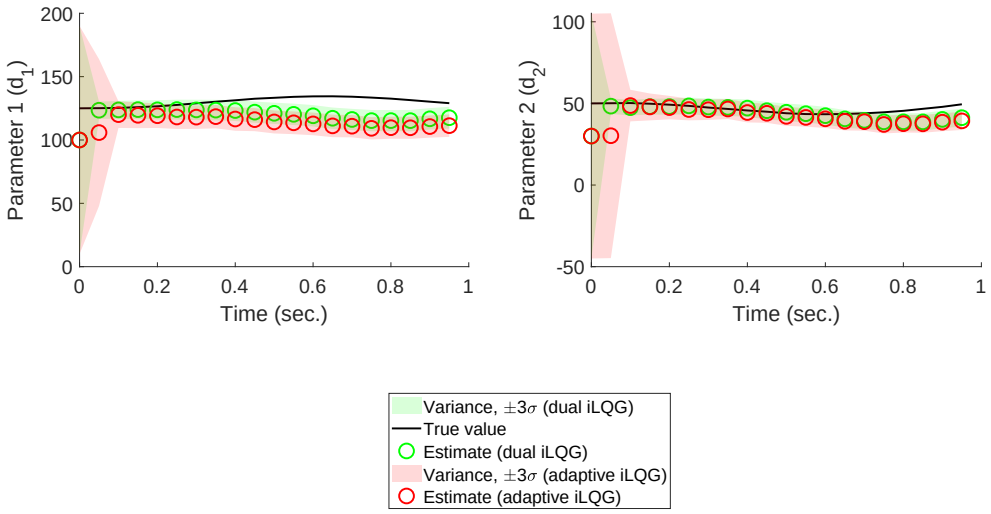


Figure 3.8: Parameter comparison between dual and adaptive iLQG on time-varying parameters example.

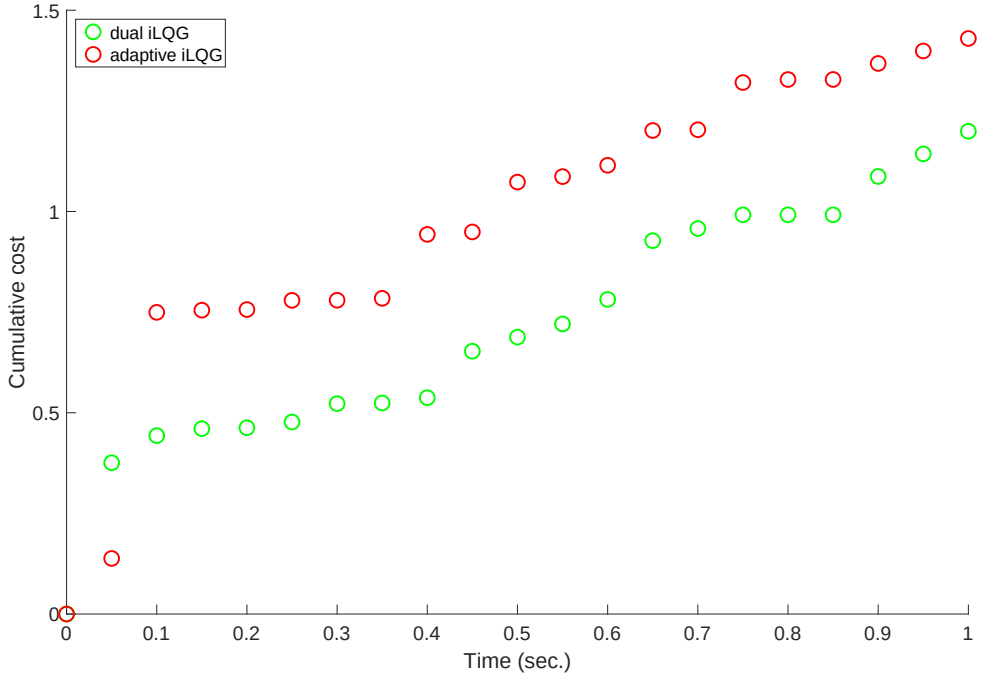


Figure 3.9: Cost comparison between dual and adaptive iLQG on time-varying parameters example.

To explore the stability and robustness of these results, 100 initial control sequence seeds were run, with each element of each sequence drawn from a normal distribution with a mean of zero and a variance of 0.01. This variance was sufficient to generate a diversity of results for these

simulations. A histogram of the final costs and a box plot of the simulation times for each of the algorithms is shown in Figures 3.10 and 3.11.

In Figure 3.10, the expected successive improvement between the three algorithms can be seen. Interestingly in this case, the best iLQG solution has roughly the same value as the median adaptive iLQG solution, and likewise the best adaptive iLQG solution has roughly the same value as the median dual iLQG solution. Additionally, the variance of the solutions visably decreases with the change in the algorithm, with dual iLQG giving more consistent results than the other two algorithms. These results show that although these algorithms give different results with different initial control trajectories, adaptive iLQG is likely to outperform iLQG, and dual iLQG is likely to outperform adaptive iLQG.

Looking at the simulation time required for these seeded runs in Figure 3.11, the results are not as may have been expected. Instead of seeing a pattern of increasing time with increasing algorithm complexity from iLQG to adaptive iLQG and dual iLQG, the classical iLQG solution appears to take more time in this example than would be expected. Perhaps the adaptation of the parameters in both the adaptive and dual cases that may have initially slowed the algorithms down led to solutions that subsequently converged faster at each time step than the non-adaptive iLQG case. Either way, the relationship between the adaptive and dual iLQG times are as expected, with the dual algorithm taking longer and having more of a spread. Interestingly, the results for both the adaptive and dual algorithm times are skewed downward, and only occasionally have longer run times. Overall, all three of these algorithms are within the same order of magnitude for the time they take to run.

Overall, the dual iLQG controller achieved a 16% reduction in the cost function compared to the adaptive controller. This difference in performance was primarily due to dual iLQG's increased ability to determine the uncertain parameters, although both dual and adaptive iLQG showed robustness to time-varying parameters in this example. In the seeded results, the consistent improvement of dual iLQG over adaptive iLQG and adaptive iLQG over iLQG was shown, as well as that they have comparable run times.

3.6 Importance of compensating for multiplicative noise

Dual iLQG can handle multiplicative noise where Bar-Shalom and Tse's wide-sense dual control is limited to additive noise and MS-SP-NMPC does not consider process noise, only measurement noise. This is important, as multiplicative noise appears in many applications such as the control

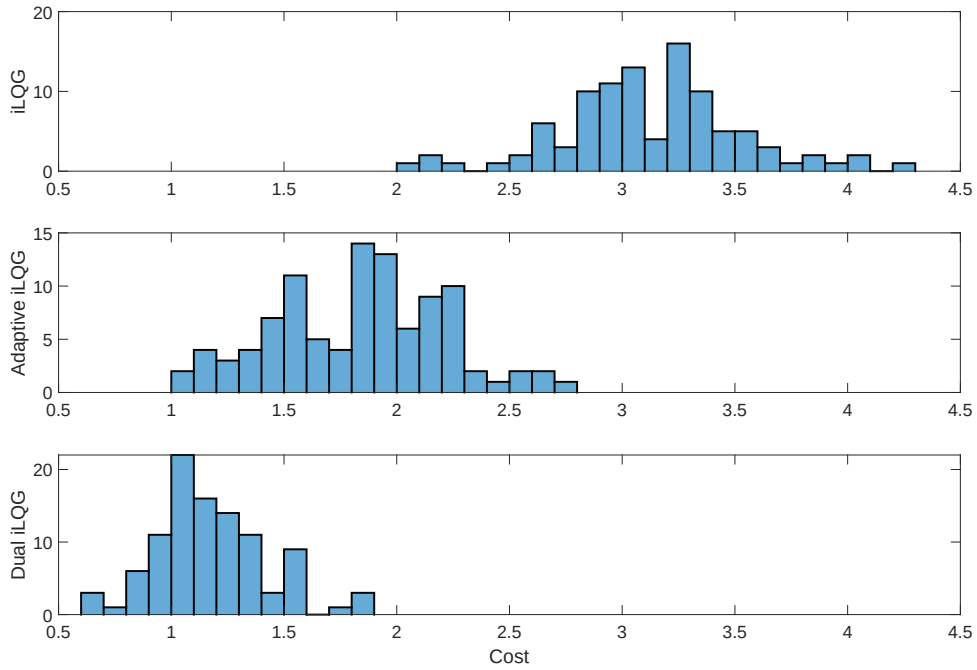


Figure 3.10: Frequency (%) of final cost of 100 seeded runs of the iLQG algorithms for the time-varying parameter example.

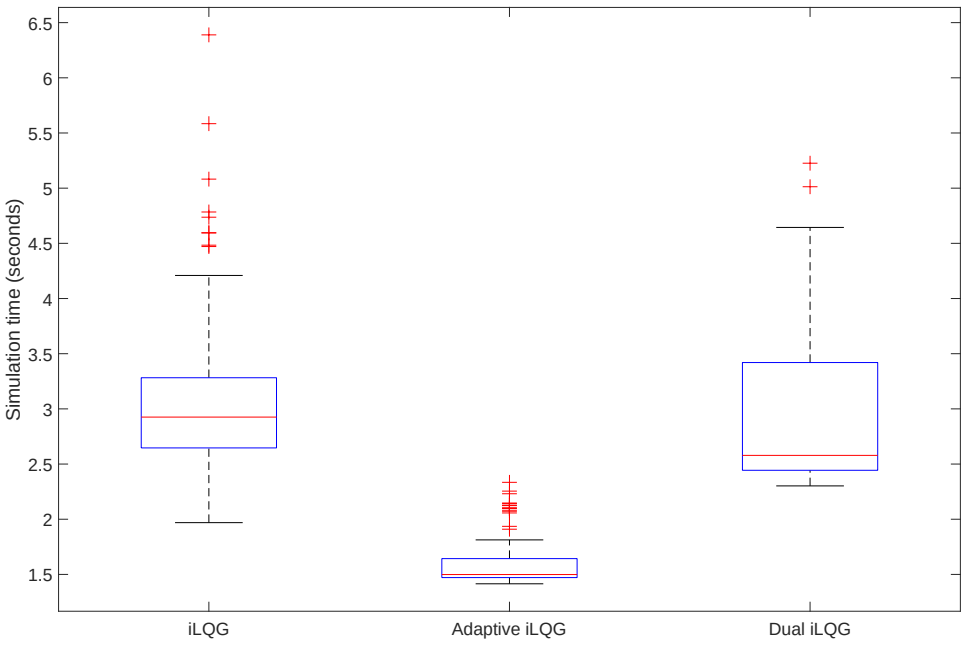


Figure 3.11: Distribution of the simulation time required for 100 seeded runs of the iLQG algorithms for the time-varying parameter example.

of prostheses using myoelectric signals [57], robot motion planning when using sensors with distance-dependent errors [15], stochastic fluid dynamics [13], and power grids where renewable energy represents a significant portion of total power generation [20].

In this example, the importance of compensating for multiplicative noise in the controller was demonstrated using the simple system dynamics shown in (3.5) with multiplicative noise. For this comparison, dual iLQG was used twice, but in one case the multiplicative noise was accounted for in the controller, and in the other case it was not. This approach was used to keep all other variables except the one in question the same and is equivalent to using a controller that can only deal with additive noise when interacting with systems that have multiplicative noise.

For this example, the system dynamics in (3.5) were modified to include multiplicative Brownian noise based on the control signals:

$$\mathrm{d}x_1 = (d_1 u_1 - d_2)\mathrm{d}t + u_1 \mathrm{d}\omega(t) \tag{3.9}$$

$$\mathrm{d}x_2 = (d_2 u_2)\mathrm{d}t + u_2 \mathrm{d}\omega(t) \tag{3.10}$$

where ω is standard Brownian noise [38]. The initial state and parameter estimates, true values, and covariances are the same as in Section 3.4, and both the states and the controls remain unconstrained. The cost function, time step length, and number of time steps are also all the same as in Section 3.4.

The results of this comparison in terms of the control signals are shown in Figure 3.12. As mentioned previously, in the example, dual controllers will use the second control action to gain information about the second parameter, allowing for increased ability to use the first control to get the first state to the desired value of 5. Here, the controller that is compensating for the multiplicative noise uses a smaller first u_2 control action than the other controller, before making a larger control action in the second step. This action allows the compensating controller to gain some information on the system before having additional noise from a larger control step. The controller that is not compensating for the multiplicative noise takes the opposite strategy in the first two time steps. The remaining u_2 and the u_1 results are relatively similar between the controllers.

Figure 3.13 shows the effect of the controls on the states, where the influence of the objective function being the square of the distance between the first state and a value of five can be seen. The non-compensating controller overshoots to 5.45 at the end of the first time step, while the compensating controller overshoots to 5.31. By the second time step, the non-compensating

controller has a x_1 value of 5.31, while the compensating controller has a value of 5.03. The non-compensating controller then takes three more time steps to get lower than 5.03, while the compensating controller stays within $\pm$ 0.04 of five.

A plot of the estimates of the two uncertain parameters is shown in Figure 3.14. Both controllers start to converge to estimates lower than the true value of the parameters, the non-compensating controller from above and the compensating controller from below. As with the unmodelled dynamics example, in this case, it is not the absolute parameter values that are important for the controller, but the ratio of the parameter values. Figure 3.15 shows the ratio of the parameters for the two controllers over the simulation. Here, it is much more obvious that the compensating controller's estimate of the parameter ratio is more accurate than that of the non-compensating controller for all but a single time step.

Figure 3.16 shows the cumulative cost of these controllers over the simulation period. The major changes in cost occur in the first two time steps, with the compensating controller's cost leveling out after the first time step and the non-compensating controller's cost leveling out after the third time step. The controller that is compensating for the multiplicative noise has a total cost that is 62.1% lower than the controller that does not compensate for the multiplicative noise.

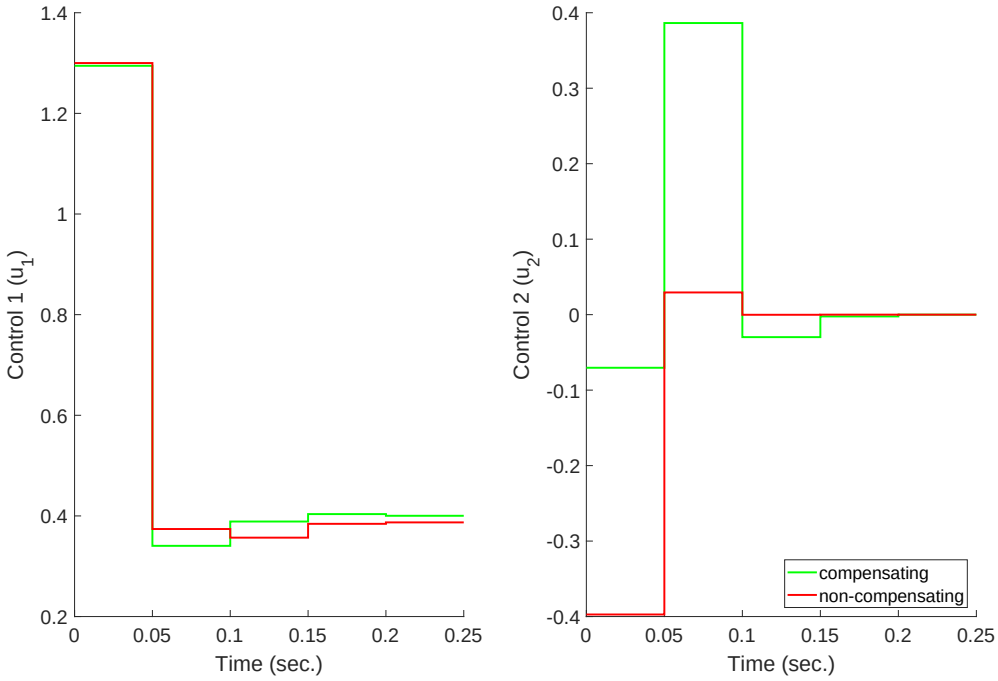


Figure 3.12: Control comparison between dual iLQG with and without the system's multiplicative noise being compensated for in the controller.

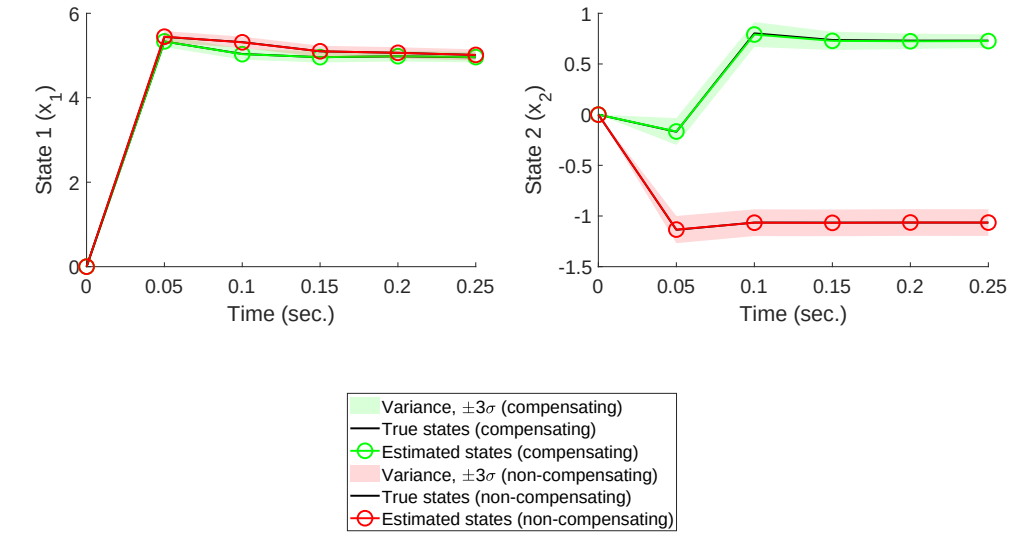


Figure 3.13: State comparison between dual iLQG with and without the system's multiplicative noise being compensated for in the controller.

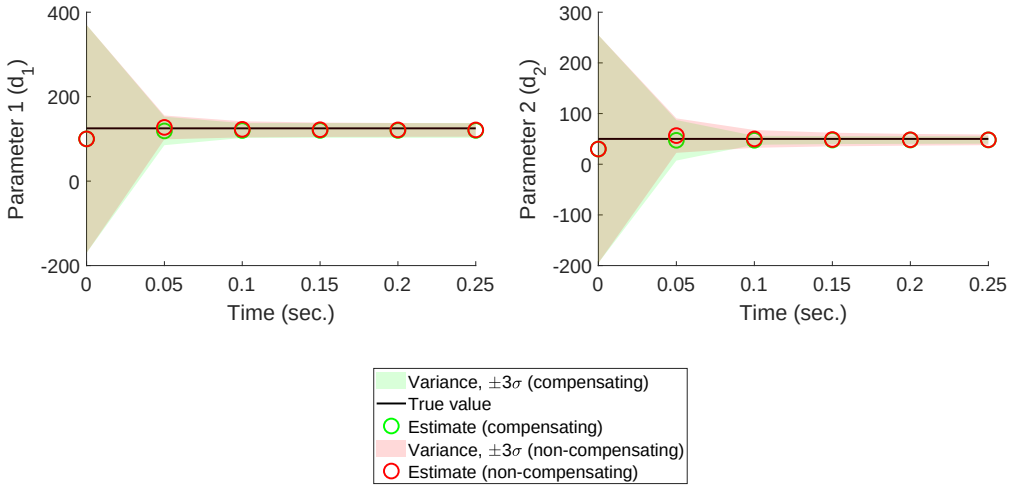


Figure 3.14: Parameter comparison between dual iLQG with and without the system's multiplicative noise being compensated for in the controller.

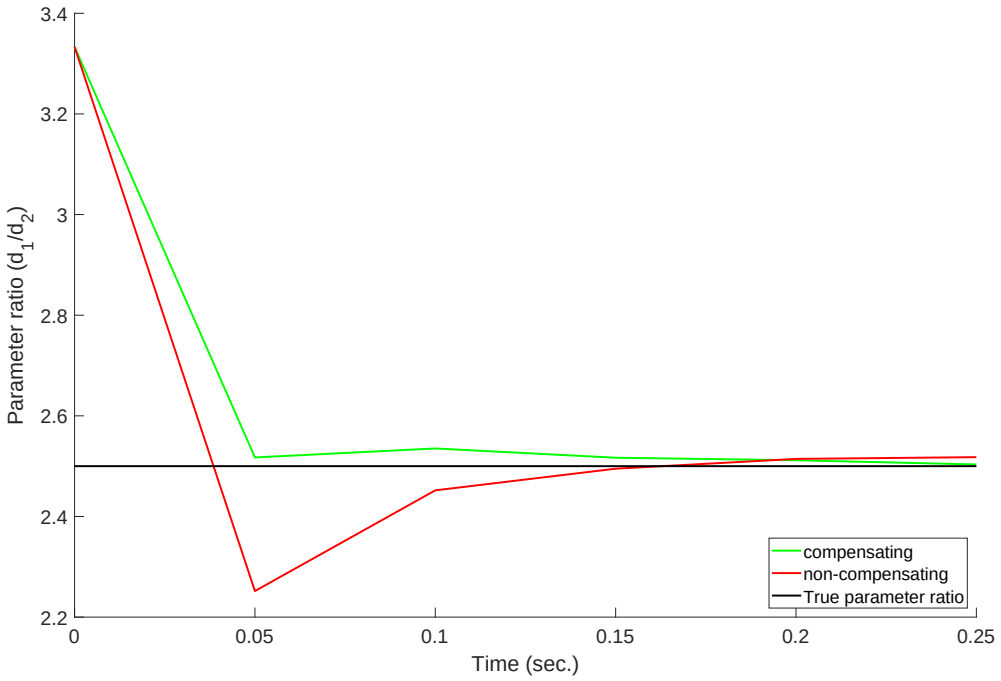


Figure 3.15: Parameter ratio comparison between dual iLQG with and without the system's multiplicative noise being compensated for in the controller.

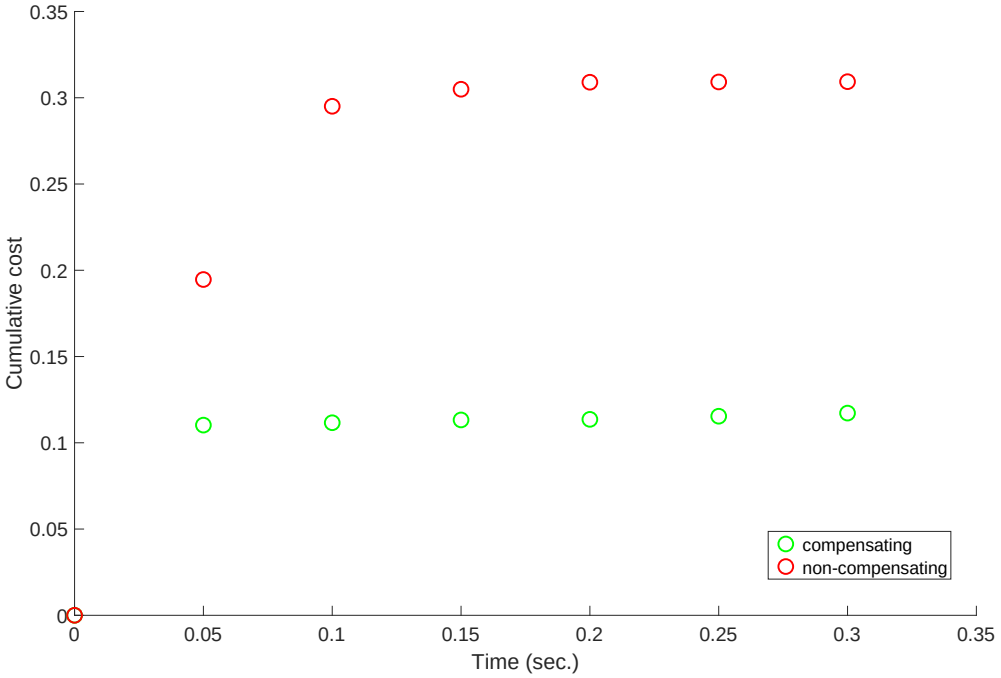


Figure 3.16: Cost comparison between dual iLQG with and without the system's multiplicative noise being compensated for in the controller.

This section demonstrates the importance of using a dual controller that can properly

compensate for multiplicative noise in systems where multiplicative noise exists. In the simple example presented, the difference in cost was significant at 62.1%.

3.7 Robustness to unmodelled dynamics

System models are often simplifications of the true dynamics of a given system, where nonlinearities or higher-order dynamics are excluded from the control model. These simplifications can be due to the cost associated with a more complete system identification or the difficulty of quantifying the mathematical relationship of a given phenomenon. The difference between the response that the algorithm expects from the system and the true response of the system can cause the values of the adapting parameters to drift over time, possibly leading to the system becoming unstable.

In Slotine and Li's Applied Nonlinear Control textbook [58], Rohr's example is used to illustrate the effect of unmodelled dynamics, showing that, unless a dead zone is used, the given adaptation law causes the first-order system to become unstable when driven to a constant reference in the face of a sinusoidal measurement noise. This simple example is used here to investigate dual iLQG's robustness to unmodelled dynamics in this case.

Rohr's example considers a desired performance that is described by a reference system that is a first-order system with a transfer function of

$$H(p) = \frac{k_m}{p + a_m} \tag{3.11}$$

where k_m and a_m are parameters both with a value of 3. The true system is a third-order system with a transfer function of

$$H(p) = \frac{k_p}{p + a_p} \frac{229}{p^2 + 30p + 229} \tag{3.12}$$

where k_p and a_p are the true system parameters with values of 2 and 1, respectively. For this example, the controller only has knowledge of the first-order portion of this model, leaving the second-order portion of the transfer function unmodelled.

These transfer functions give state space models of

$$\dot{x} = -ax + u \tag{3.13}$$

$$y = kx \tag{3.14}$$

for the first-order system, where y is the system output, and

$$\dot{x} = \begin{bmatrix} -(30+a_m)x_1 - (229+30a_m)x_2 - 229a_m x_3 + u \\ x_1 \\ x_2 \end{bmatrix} \tag{3.15}$$

$$y = 229k_m x_3 \tag{3.16}$$

for the true third-order system. The system also has a measurement noise of $n(t) = 0.5\sin(16.1t)$.

In [58], model-reference adaptive control is used to control this system and is implemented as shown in Figure 3.17. In model-reference adaptive control, the desired system behaviour is specified through the use of a reference model. The controller then attempts to make the true system's output equal to the output from the reference model by adjusting the input to the true system based on a set of known regressors and adapting parameters. This allows model-reference adaptive control to adapt a set of compensating parameters to account for the true system's behaviour and impose the desired behaviour onto the system. This is different than the dual iLQG approach which is to adapt its estimates of the true system's parameters to a sufficient level to minimize the desired cost function.

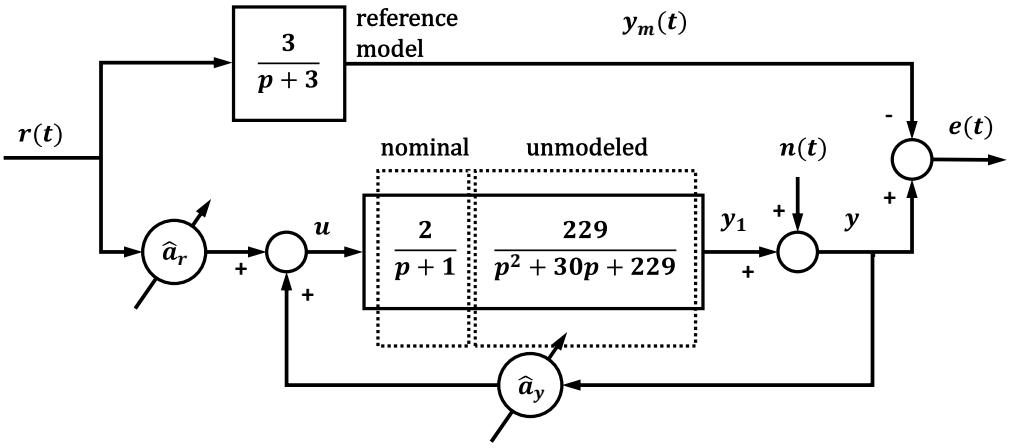


Figure 3.17: Model-reference adaptive control flowchart for Rohr's example [58].

To implement model-reference adaptive control on this example, a control law of

$$u = \hat{a}_r r + \hat{a}_y y \tag{3.17}$$

was used where r is the input reference and y is the output of the true system after the addition

of noise, and the $\hat{a}$ terms are the corresponding adaptation parameters. These parameters were updated with a adaptation law of

$$\dot{\hat{\mathbf{a}}} = -\gamma \mathbf{v} e, \tag{3.18}$$

where $\hat{\mathbf{a}}$ is the adaptation parameter vector $[\hat{a}_r,\ \hat{a}_y]^\top$, γ is a positive constant, $\mathbf{v}$ is the regressor vector which is equal to $[r,\ y]^\top$ in this case, and e is the error between y and the reference model output y_m.

To run this example with dual iLQG, the true values and initial estimates listed previously are used, with a covariance of I_2. The states are initialized at $[0, 0, 0]$ for both the true values and the estimates, and a state covariance of 10^{-15} is used as the problem does not include state uncertainties. Both the states and controls are unconstrained for this example, and the parameters are constants. A time step length of 0.1 seconds was used and the system was simulated over 700 time steps to match the period shown in [58]. The measurement noise scaling factors in $\mathcal{G}$ were 0.5 and the noise associated with the state and parameter dynamics ($\mathcal{F}$) was 10^{-15}. The sinusoidal noise from the problem was implemented on the true system, which the algorithms had no knowledge of other than the 0.5 noise scaling factor. The cost function was

$$J(\mathbf{x}, \mathbf{u}) = (y - 2)^2, \tag{3.19}$$

where the modelled and true output equations are given in equations (3.14) and (3.16).

The resulting controls from both the adaptive and dual iLQG controllers as well as the model-reference adaptive controller are shown in Figure 3.18 for the entire 70 seconds, and in Figure 3.19 for the first seven seconds. In these figures, it can be seen that both of the iLQG controllers provided similar high-frequency control actions with amplitudes that decreased over the first 1.5 seconds before evening out at a nearly steady-state value of around 1. During this initial oscillatory period, dual iLQG generally had a slightly smaller control amplitude than the adaptive iLQG, and in the steady region, both controllers continued to have small variations in their controls. In contrast, the model-reference adaptive controller starts with a lower frequency control action before oscillating around a value of 1. The amplitude of this oscillation about a control action of 1 grows until just after 60 seconds when the controller fails.

Figures 3.20 and 3.21 show the true and estimated states for all three controllers over the entire 70 seconds and for just the first seven seconds, respectively. The response to the high-frequency control actions from the iLQG controllers can be seen in the true dynamics for the

first two states, while their estimates remained at zero with increasing uncertainty because there were no dynamics for those states in the modelled system and they were unmeasured. The true state response for the third state also reflected the high-frequency input but was very small compared to the estimated state values due to the 229 scaling factor in the true system output. These large discrepancies between the true and estimated states were because they are different systems, with only the relationship between the inputs and the outputs being comparable. For the third state estimates, after some initial oscillations, there was a sustained offset between the dual and adaptive estimates. Finally, in Figure 3.20 it can be seen that the system remained stable over the entire 70 second period. As the model-reference adaptive controller does not estimate the states, only true values are shown. The low-frequency response in the first two states is due to the low-frequency controls, and the low value of the third state is again due to the 229 scaling factor.

The estimates of the uncertain parameters can be seen in Figures 3.22 and 3.23, with Figure 3.23 again being a zoomed-in version. Although the model-reference adaptive controller does not directly estimate these parameters and instead adapts the compensation parameters discussed, these parameter values can be derived from the definitions of the compensation parameters as

$$\hat{d}_1 = \frac{k_m}{\hat{a}_r} \tag{3.20}$$

$$\hat{d}_2 = \hat{a}_y \hat{d}_1 + a_m. \tag{3.21}$$

Using these relations, the parameter estimates are shown in Figures 3.22 and 3.23 were determined. After about 2 seconds these estimates stop changing as quickly and are close to the true values, but over the full simulation time, they drift until they cause the true system to become unstable just after 60 seconds. Figure 3.22 shows that for the iLQG controllers, after an initial adaptation period, the estimates of the parameters for the controllers remained largely constant and that there was no significant parameter drifting occurring although the estimates of the second parameter did decrease slightly over the 70 second simulation. Zooming in to the initial adaptation period shown in Figure 3.23, both controllers' estimates of the first parameter initially approached the true value of 2, but then gradually converged to a final value near 2.6. For the second parameter, there were no initial rapid changes in the estimates, just a gradual convergence to 1.4. Additionally, adaptive iLQG appeared to have better parameter estimates than dual iLQG, which was unexpected considering that the dual controller ended up with a lower cost.

To understand how dual iLQG performed better in this example while the adaptive controller

had better parameter estimates, a closer look at the system is required. Knowing that this system quickly reaches a steady state, an expression for the steady-state output of the first-order system can be shown to be

$$y_{ss} = \frac{ku_{ss}}{a}, \tag{3.22}$$

where the subscript ss denotes steady state. With the sole term in the objective function being the squared difference between the output and a reference of 2, it becomes clear that it is not the individual values of the parameters that are important for this problem, but the ratio of $\frac{k}{a}$. Knowing this, the ratio of $\frac{\hat{k}}{\hat{a}}$ can be plotted, as shown in Figure 3.24 for the entire time horizon, and for the first seven seconds in Figure 3.25. Although difficult to see in Figure 3.24, Figure 3.25 shows that dual iLQG had a slightly better estimate of this ratio for the first 1.5 seconds, after which adaptive iLQG is largely better. Figure 3.24 also more clearly shows that the slight drifting of the second parameter in Figure 3.22 resulted in the ratio of the parameter estimates from both controllers nearly converging to the true value by the end of the 70 second simulation. Model-reference adaptive control oscillates around the true parameter ratio, but over time diverges as the parameter drift.

Figures 3.26 and 3.27 show the full and initial plots of the modelled and true system output. In Figure 3.26, the modelled system output is shown to approach 2.01, very close to the desired value of 2, while the true system output approached 1.91. Since the modelled system had practically reached the desired output reference, the controllers' input stopped changing and the true system's output also stopped changing, but not necessarily at the desired output reference due to the unmodelled dynamics and measurement noise. Model-reference adaptive control oscillates around the desired output of 2 but diverges quickly 60 seconds into the simulation.

The stage and total costs for the adaptive and dual iLQG algorithms are shown in Figures 3.28 and 3.29, with the latter figure being only the initial seven seconds of the simulation. These plots directly reflect the square of the distance of the true system output in Figures 3.26 and 3.27 from the desired reference of 2. Although both controllers ended up with small but non-zero stage costs for the second part of the simulation, dual iLQG outperformed adaptive iLQG for the first part of the simulation, resulting in dual iLQG having an 11% lower total cost. Model-reference adaptive control is not shown as it is not based on a cost function.

For the model-reference adaptive controller in [58], the constant reference does not provide sufficient information for the parameter identification to distinguish information from noise, causing the parameters to slowly drift. After drifting significantly beyond their true values, the

parameter estimates cause the system to become unstable and suddenly diverge around 70 seconds into the simulation. On the other hand, both dual and adaptive iLQG demonstrated robustness to unmodelled dynamics and were able to get the system's output very close to the desired reference. Unlike the nonlinear controller in [58], these controllers did not cause their parameter estimates to drift and avoided the instability that caused Slotine and Li's controller to fail.

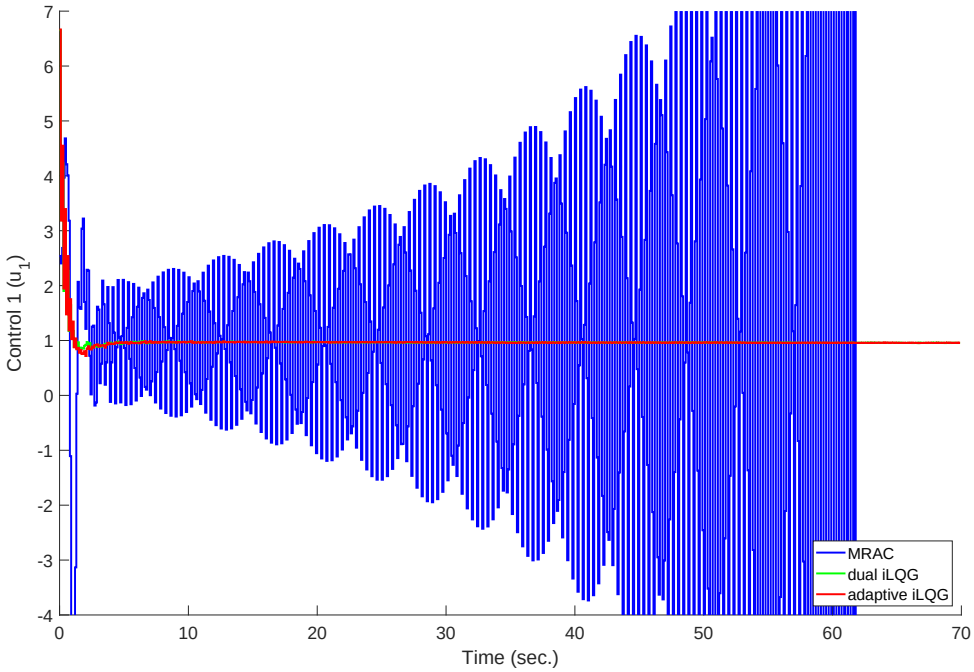


Figure 3.18: Control comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example.

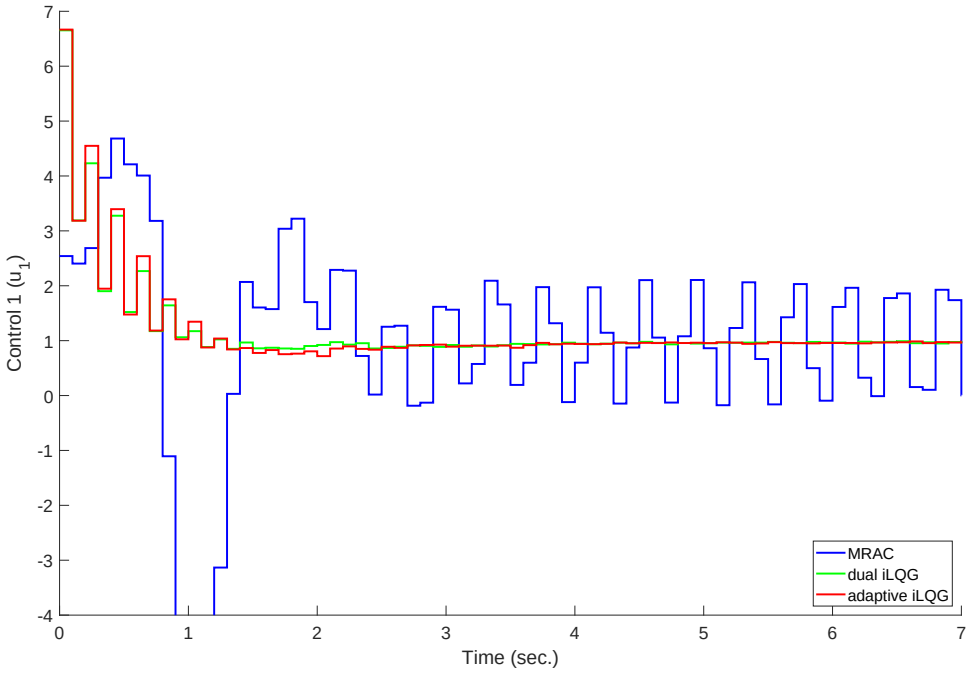


Figure 3.19: Control comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example for first seven seconds.

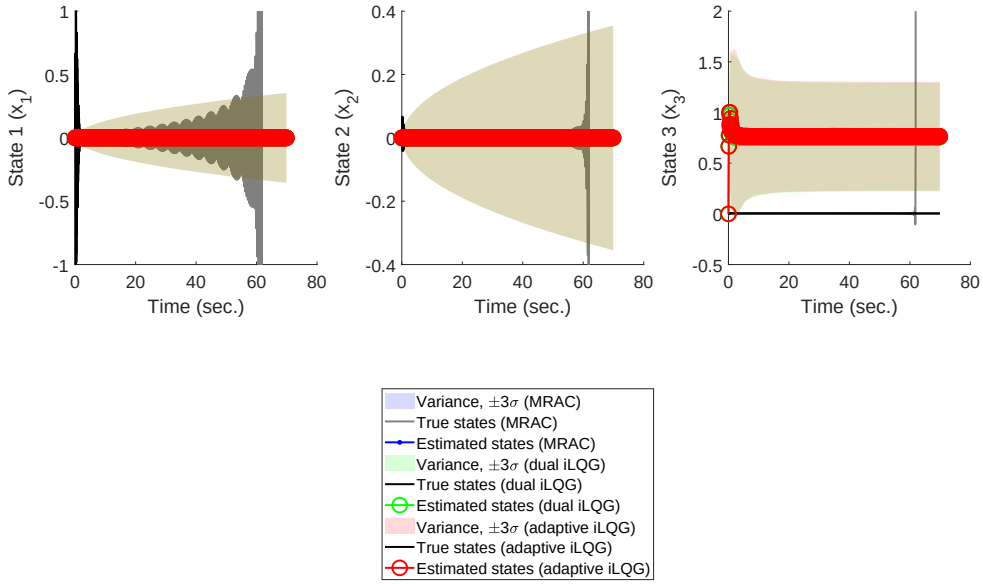


Figure 3.20: State comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example.

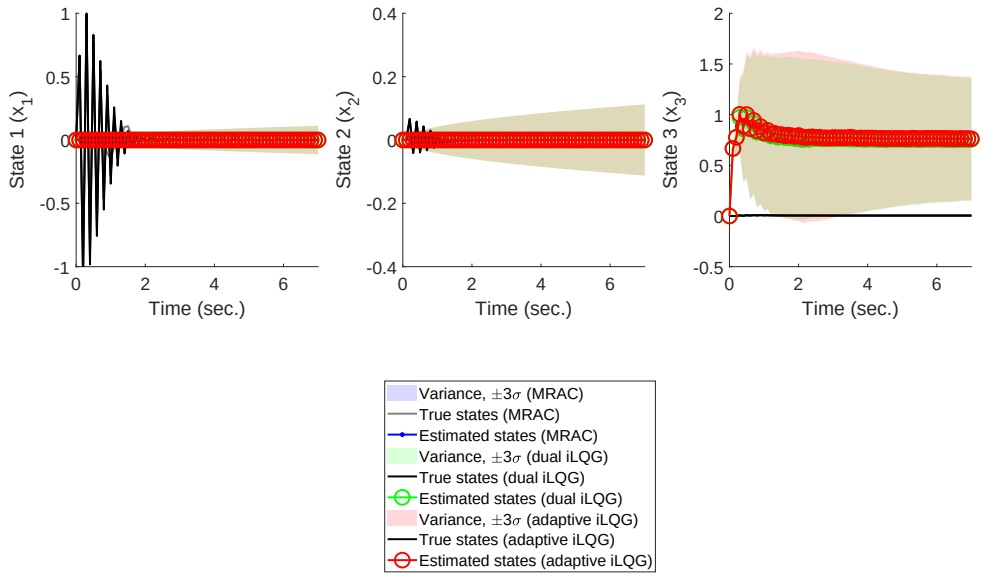


Figure 3.21: State comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example for first seven seconds.

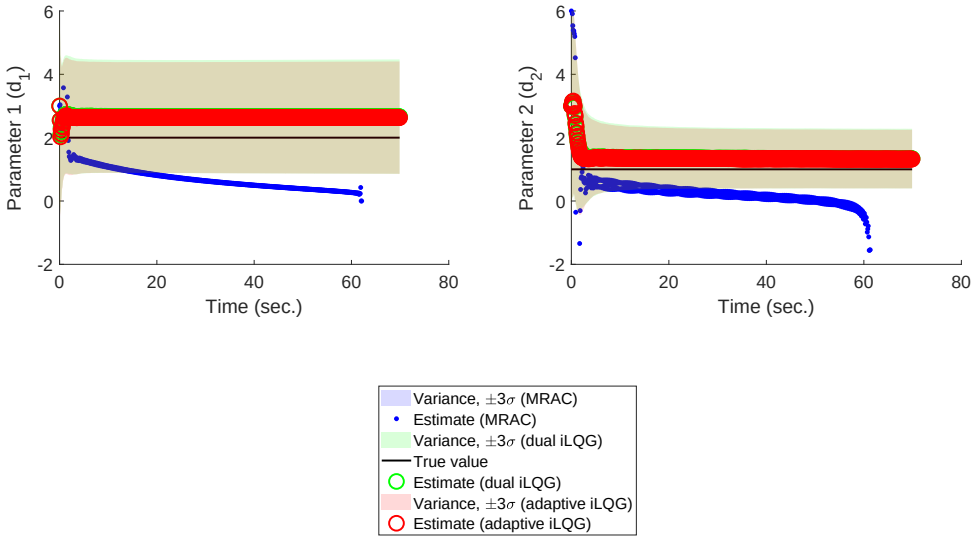


Figure 3.22: Parameter comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example.

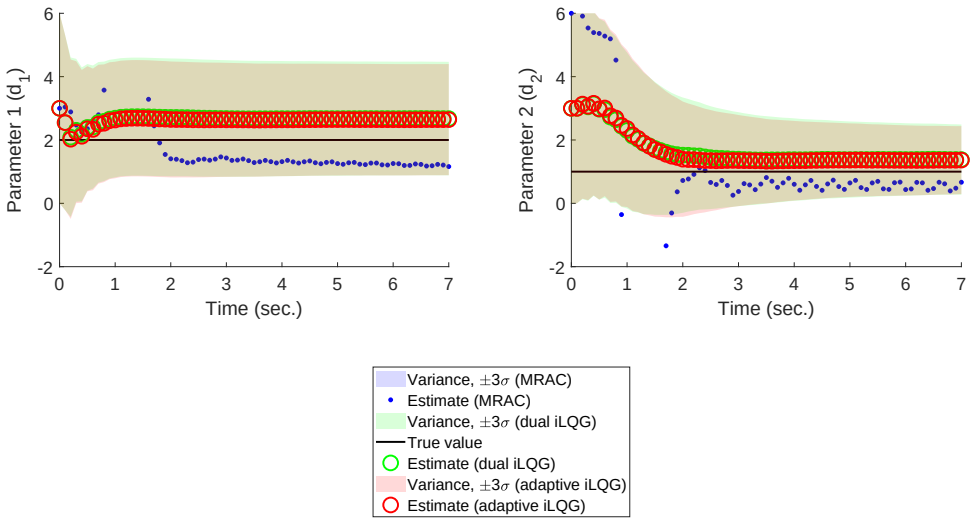


Figure 3.23: Parameter comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example for first seven seconds.

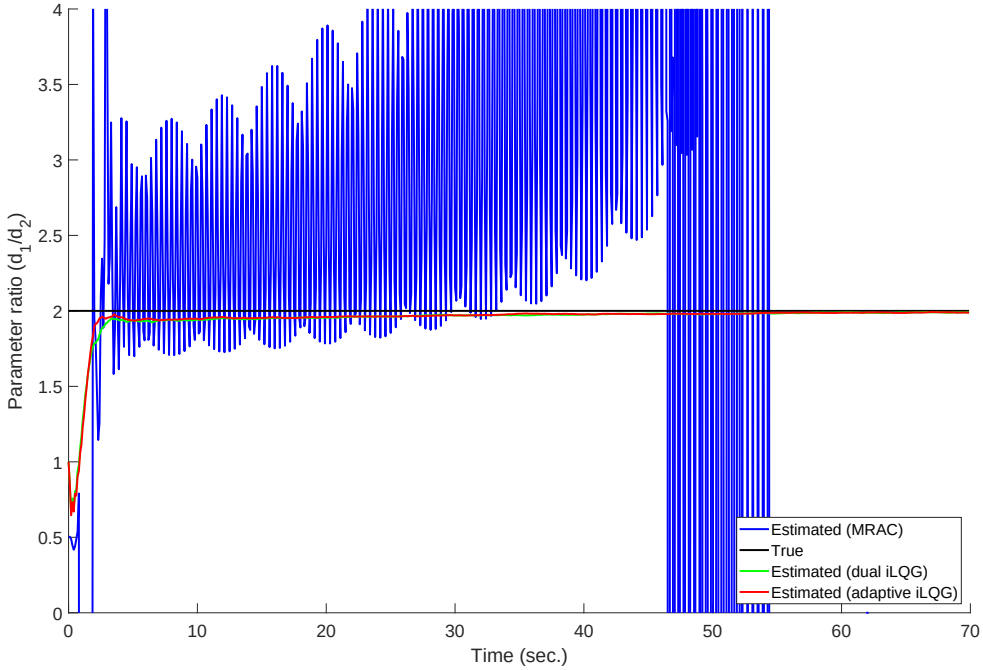


Figure 3.24: Parameter ratio comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example.

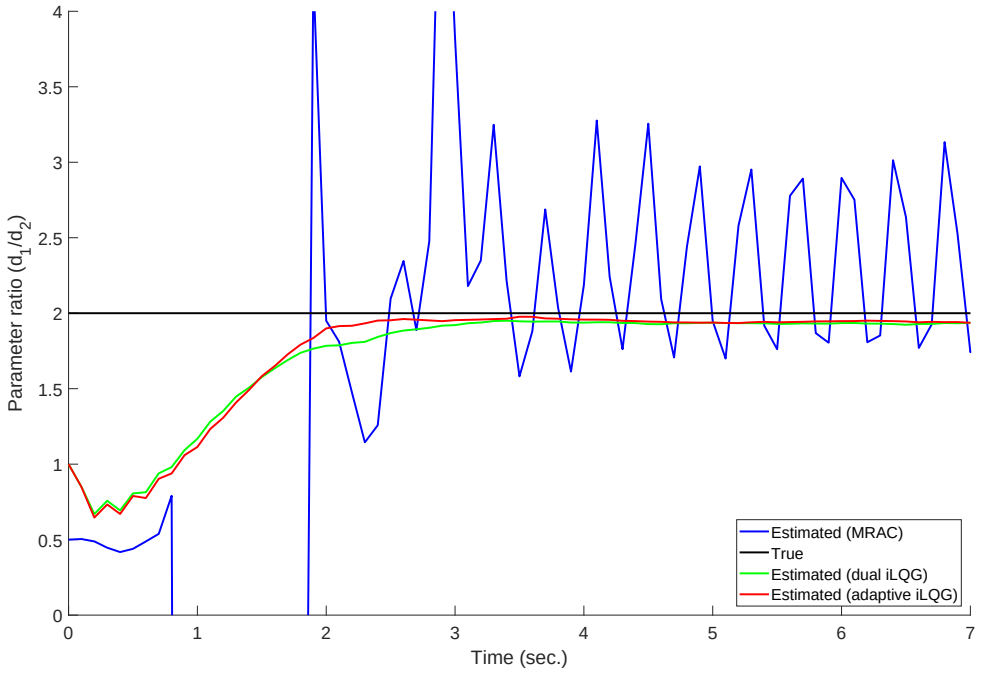


Figure 3.25: Parameter ratio comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example for first seven seconds.

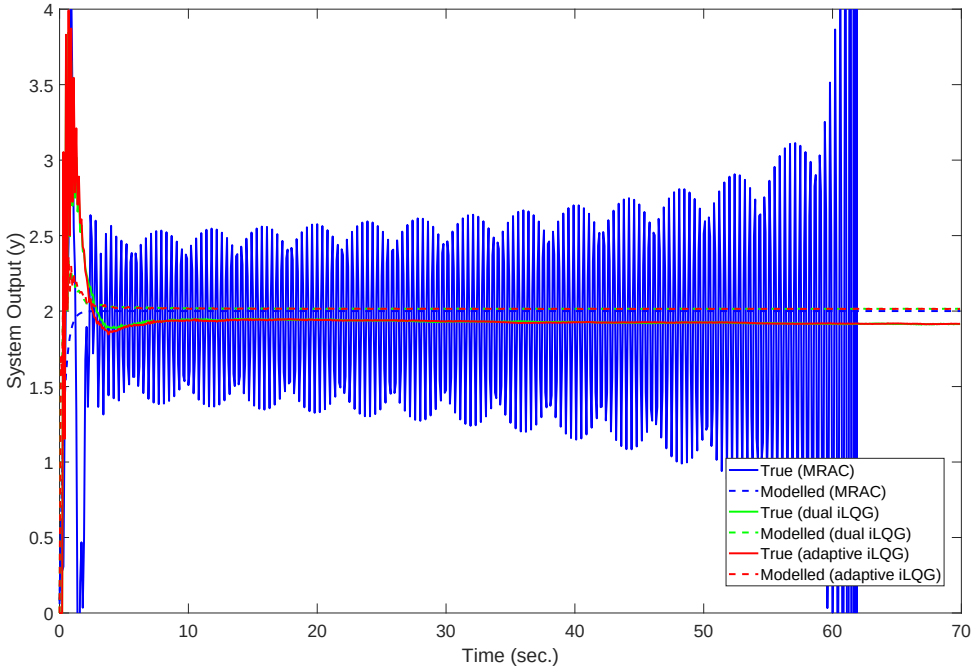


Figure 3.26: Output comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example.

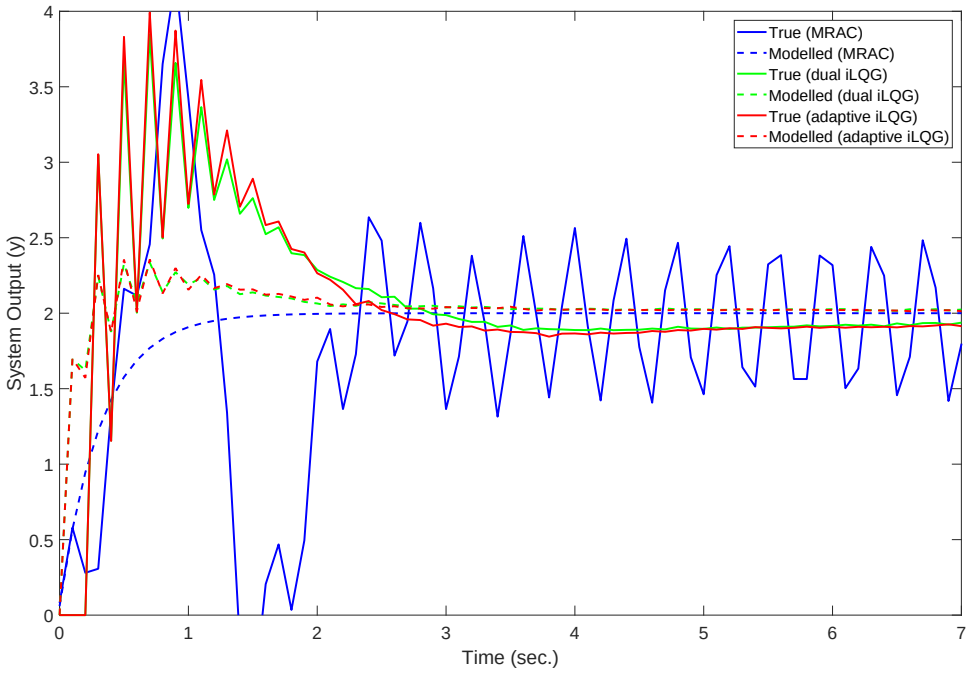


Figure 3.27: Output comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example for first seven seconds.

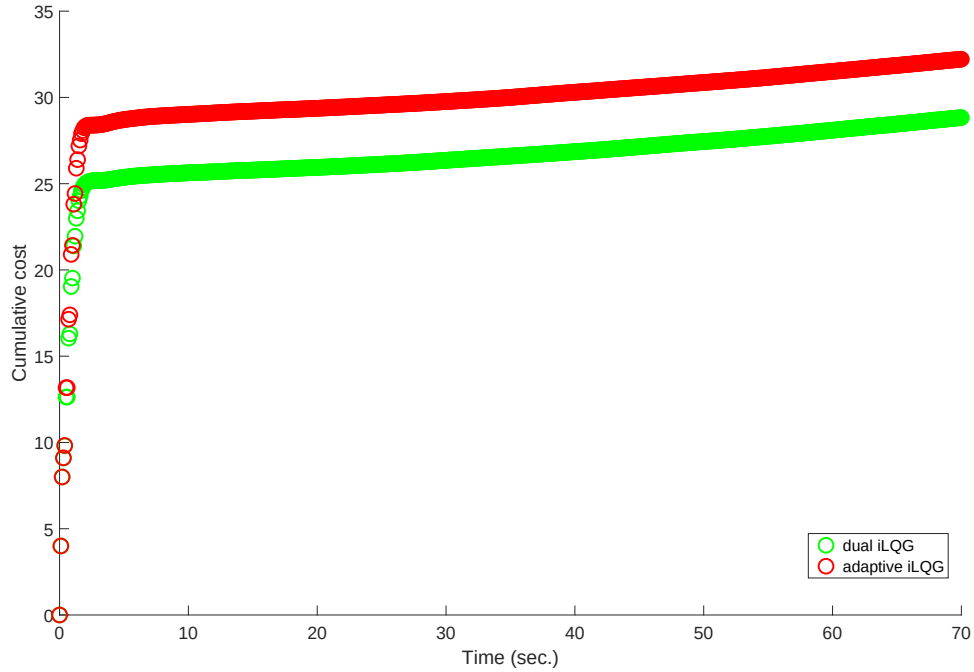


Figure 3.28: Cost comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example.

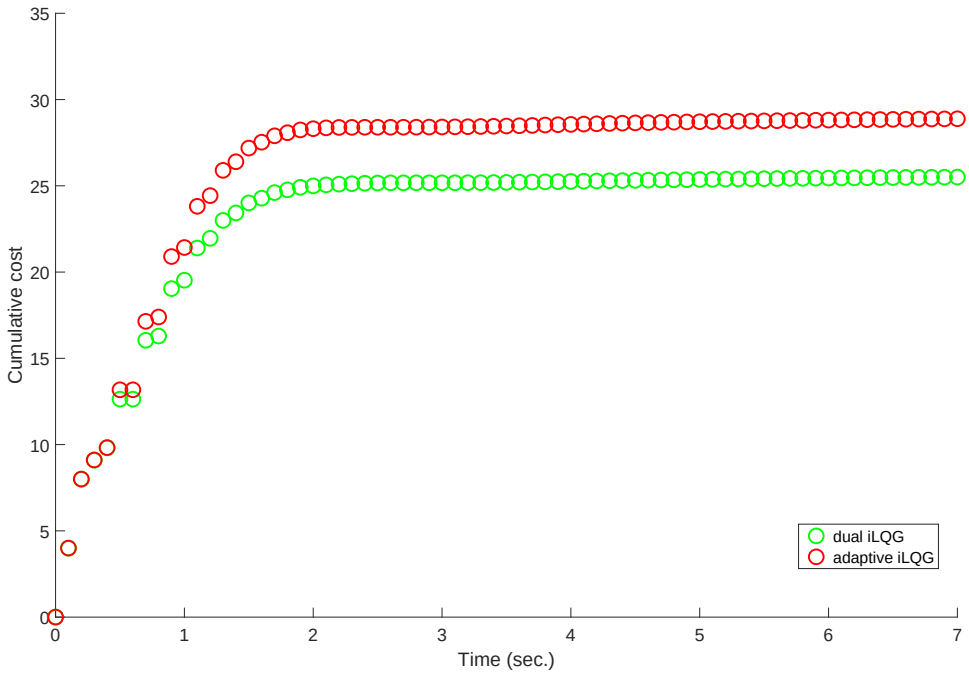


Figure 3.29: Cost comparison between dual and adaptive iLQG and model-reference adaptive control on unmodelled dynamics example for first seven seconds.

3.8 Summary

The extension of iLQG to be a dual control method was demonstrated in this chapter. Dual iLQG shows promise as a high-speed dual control algorithm that can be applied to larger problems than other dual control algorithms due to its derivative-based approach. Dual iLQG can also handle multiplicative noise where wide-sense dual control and MS-SP-NMPC cannot.

After describing how iLQG can be extended to be adaptive or dual, dual iLQG was compared with dual MS-SP-NMPC on a simple linear system where dual iLQG significantly outperformed the other controllers. The robustness of dual and adaptive iLQG to time-varying parameters and unmodelled dynamics on specific systems was then investigated. Although dual iLQG outperformed adaptive iLQG in these two examples, both controllers demonstrated robustness to time-varying parameters and unmodelled dynamics. The importance of using controllers that are designed to handle multiplicative noise was also demonstrated.

In the next two chapters, dual iLQG will be applied to a complex biochemical system, an anaerobic digestion, and a complex public policy control problem, the spread of COVID-19 in a population.

Chapter 4

Application to anaerobic digestion

4.1 Motivation

Anaerobic digestion is a chemical process where organic materials are broken down by various types of bacteria in a low-oxygen environment to produce a gas that is commonly known as biogas, which consists primarily of methane and carbon dioxide [10]. This biogas has various uses as a fuel, and can, for instance, be used directly for cooking or heating. It can also be compressed into a liquid fuel that is similar to natural gas, or it can be used to generate heat and power with a combined heat and power system [26]. Compared to other renewable energy sources, biogas can provide consistent power independent of the weather, can be located anywhere there is a consistent supply of organic waste, and the energy can easily be stored for later use. Therefore, anaerobic digestion represents a controllable renewable energy source that is free of many of the intermittency, storage, and site location issues common with other sources of renewable energy.

Anaerobic digestion has several characteristics that make it an ideal application for dual control. First of all, anaerobic digestion models have uncertain parameters, and in particular those associated with the dynamics of the bacteria populations are difficult to measure and have a significant impact on the rest of the system dynamics. Second, it is common for not all of the states to be measured, and those that are measured are noisy. Finally, although not commonly modelled, the composition of the organic feedstocks to the digester is not known precisely. This combination of uncertainties in the feedstock composition, bacteria population dynamics, and the measurements makes this application well-suited for the application of dual control. That being said, these characteristics do not uniquely lend themselves to dual control, and other control approaches such as robust or machine learning methods could also be used. As the focus of this

thesis is dual control though, that will be the focus here.

4.2 System model

Two widely used models used to represent the anaerobic digestion dynamics are Anaerobic Digestion Model No. 1 (or ADM1) [6] and AM2 [10]. ADM1 is a comprehensive model that has 24 state variables and is commonly used to simulate anaerobic digestion systems, but due to its complexity has limited use for control approaches [27]. AM2 on the other hand has six state variables while still capturing the main dynamics of the process and is commonly used for model-based control and parameter estimation [26]. For these reasons, the AM2 model was used for this work, but future work should investigate the feasibility of using dual iLQG with ADM1.

4.2.1 The AM2 model

The AM2 model represents the anaerobic digestion process as

$$\begin{aligned}
\dot{X}_1 &= (\mu_1(S_1) - \alpha D)\, X_1, && (4.1)\\
\dot{X}_2 &= (\mu_2(S_2) - \alpha D)\, X_2, && (4.2)\\
\dot{S}_1 &= D(S_1^{in} - S_1) - k_1\mu_1(S_1)X_1, && (4.3)\\
\dot{S}_2 &= D(S_2^{in} - S_2) + k_2\mu_1(S_1)X_1 - k_3\mu_2(S_2)X_2, && (4.4)\\
\dot{Z} &= D(Z^{in} - Z), && (4.5)\\
\dot{C} &= D(C^{in} - C) - q_{CO_2} + k_4\mu_1(S_1)X_1 + k_5\mu_2(S_2)X_2 && (4.6)
\end{aligned}$$

where X_1 is the concentration of acidogenic bacteria, X_2 is the concentration of methanogenic bacteria, S_1 is the organic substrate concentration, S_2 is the volatile fatty acid concentration, Z is the total alkalinity concentration, C is the total inorganic carbon concentration, and the model parameters are described in Table 4.1 [52]. The inputs to this model are the dilution rate, D, along with the inlet concentrations of organic substrate, volatile fatty acids, total alkalinity, and total inorganic carbon, S_1^{in}, S_2^{in}, Z^{in}, and C^{in} respectively.

The growth rates of the acidogenic bacteria, μ_1, are assumed to follow Michaelis-Menten kinetics,

$$\mu_1(S_1) = \bar{\mu}_1 \frac{S_1}{S_1 + K_{S_1}}, \tag{4.7}$$

and the growth rates of the methanogenic bacteria, μ_2, are assumed to follow Haldane kinetics,

$$\mu_2(S_2) = \bar{\mu}_2 \frac{S_2}{S_2 + K_{S_2} + S_2^2/K_{I_2}}, \tag{4.8}$$

where The molar flow rate of carbon dioxide, q_{CO_2}, is given by

$$q_{CO_2} = k_L a(C + S_2 - Z - K_H P_{CO_2}), \tag{4.9}$$

where $\bar{\mu}_1$ and $\bar{\mu}_2$ are the maximum respective growth rates.

$$P_{CO_2} = \frac{\phi_{CO_2} - \sqrt{\phi_{CO_2}^2 - 4K_H P_t(C + S_2 - Z)}}{2K_H}, \tag{4.10}$$

$$\phi_{CO_2} = C + S_2 - Z + K_H P_t + \frac{k_6}{k_L a}\mu_2(S_2)X_2, \tag{4.11}$$

and the molar flow rate methane, q_{CH_4}, is given by

$$q_{CH_4} = k_6 \mu_2(S_2) X_2. \tag{4.12}$$

The flow rate of biogas, q_{biogas}, is then assumed to be consisting only of carbon dioxide and methane, and is therefore given by

$$q_{biogas} = q_{CO_2} + q_{CH_4}. \tag{4.13}$$

4.2.2 AM2 with uncertain feedstocks

To model the uncertainty in the digester feedstocks, the inputs to the AM2 model were modified and parameters to characterize the feedstocks were added.

For a case with three feedstocks, the flow rates of these feedstocks become the new inputs to the model, and can be represented by q_a, q_b, and q_c. The dilution rate, therefore, becomes the sum of the individual flow rates,

$$D = q_a + q_b + q_c. \tag{4.14}$$

Parameter	Definition	Units
α	Fraction of biomass in the liquid phase	mmol/l
k_1	Yield for substrate degradation	g/g
k_2	Yield for volatile fatty acid production	mmol/g
k_3	Yield for volatile fatty acid consumption	mmol/g
k_4	Yield for carbon dioxide production	mmol/g
k_5	Yield for carbon dioxide production	mmol/g
k_6	Yield for methane production	mmol/g
μ_1	Specific acidogenic bacteria growth rate	1/day
K_{S_1}	Half-saturation constant associated with μ_1	g/l
μ_2	Specific methanogenic bacteria growth rate	1/day
K_{S_2}	Half-saturation constant associated with μ_2	mmol/l
K_{I_2}	Inhibition constant associated with μ_2	mmol/l
K_H	Henry's constant	mmol/(l atm)
k_La	Liquid/gas transfer constant	1/day
P_t	Total pressure	atm

Table 4.1: Parameters for the AM2 model

The inlet concentrations can then be represented as

$$S_1^{in} = \frac{S_1^a q_a + S_1^b q_b + S_1^c q_c}{D}, \tag{4.15}$$

$$S_2^{in} = \frac{S_2^a q_a + S_2^b q_b + S_2^c q_c}{D}, \tag{4.16}$$

$$Z^{in} = \frac{Z^a q_a + Z^b q_b + Z^c q_c}{D}, \tag{4.17}$$

$$C^{in} = \frac{C^a q_a + C^b q_b + C^c q_c}{D}, \tag{4.18}$$

where S_1^a, S_2^a, Z^a, and C^a characterize the organic substrate concentration, volatile fatty acid concentration, total alkalinity concentration, and total inorganic carbon concentration for feedstock A, and S_1^b, S_2^b, Z^b, and C^b and S_1^c, S_2^c, Z^c, and C^c do the same for feedstocks B and C respectively.

4.3 Results

4.3.1 Two-parameter comparison with dual MS-SP-NMPC

The true and estimated initial values of the state vector, $\mathbf{x} = [X_1, X_2, S_1, S_2, Z, C]$ are both $[0.5, 1, 1, 5, 40, 50]^\top$ and the state are all constrained to be greater than 0. The controls, $\mathbf{u} = [q_a, q_b, q_c]$ are all constrained to be between 0.0001 and 1. The true value of the uncertain parameter vector, $\mathbf{d} = \left[\bar{\mu}_1, Z^C\right]$, is $[1.2, 100]$, while its initial estimate is $[2, 50]$ with a covariance matrix of diag$[0.16, 1111.11]$. This system is simulated over 70 time steps that are 0.1 days long.

Only states $[S_1, S_2, C]$ are measured with an noise term $\mathcal{G}$ of $0.5I_3$. The process noise scaling matrix $\mathcal{F}$ for the dual iLQG state and parameter dynamics were set to $10^{-15}I_{23}$ to be comparable with the inability of MS-SP-NMPC to handle noise. The cost function is based on tracking a desired biogas flow rate and is given as

$$J(\mathbf{x}, \mathbf{u}) = (q_{biogas} - 100)^2. \tag{4.19}$$

Starting with Figure 4.1 which shows the three feedstock controls for the AM2 model for each of the three controllers, it can be seen that the all three algorithms took different approaches. The iLQG algorithms were similar in that they varied all three feedstock controls, where the MS-SP-NMPC algorithm focused primarily on the first feedstock control and only momentarily used the other two. As there were no terms in the cost function to explicitly penalize or encourage the use of the feedstock controls, the relative performance of these controls could only be determined by their impact on tracking the desired rate of biogas production.

The plot of the anaerobic digestion states is shown in Figure 4.2, with solid lines for the true state trajectories and markers for the state estimate means and variances. As would be expected from the difference between the controls, the state trajectories are significantly different between the three algorithms. The impact of $\bar{\mu}_1$ being uncertain can be seen in the large covariance associated with the third state, S_2, even though it was being measured.

Figure 4.3 shows the plot of the parameter estimates, with the true values shown as black lines. While the iLQG algorithms both approached the true values of the parameters, the MS-SP-NMPC algorithm's estimate of the first parameter, $\bar{\mu}_1$, was close but not as accurate, and not accurate at all for the second parameter, Z_C. The reason for the MS-SP-NMPC algorithm's estimate of Z_C being poor was that it hardly used the third control, q_C, and therefore had little opportunity to get feedback on Z_C.

Looking next at the plot of the biogas production in Figure 4.4, the MS-SP-NMPC algorithm's strategy of focusing on only one of the feedstock controls was successful, as it was largely able to maintain the biogas production level near the desired flow rate, with some variation. The dual iLQG controller performed even better though and maintained a steady level of biogas production after an initial ramp up. The the biogas production of the adaptive iLQG controller approaches the desired level, but does not reach it and fluctuates slowly.

The ability of the controllers to track the desired biogas production level was directly reflected in their costs, as shown in Figure 4.5. Comparing the total costs, dual MS-SP-NMPC had a

68.5% improvement over adaptive iLQG, while dual iLQG had a 98.6% improvement over dual MS-SP-NMPC.

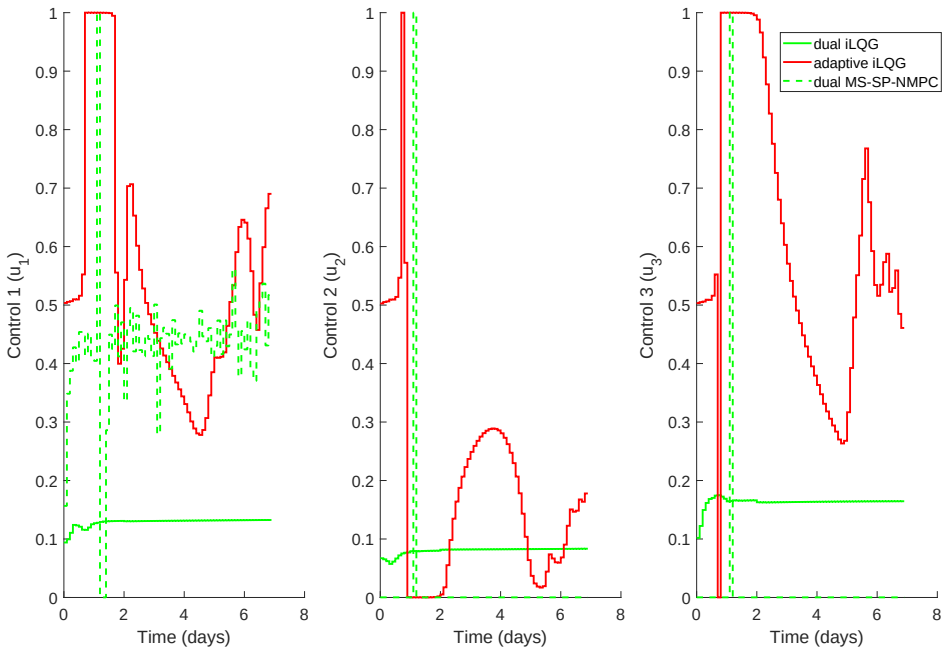


Figure 4.1: Control comparison between dual and adaptive iLQG and dual MS-SP-NMPC on AM2 anaerobic digestion model with two uncertain parameters.

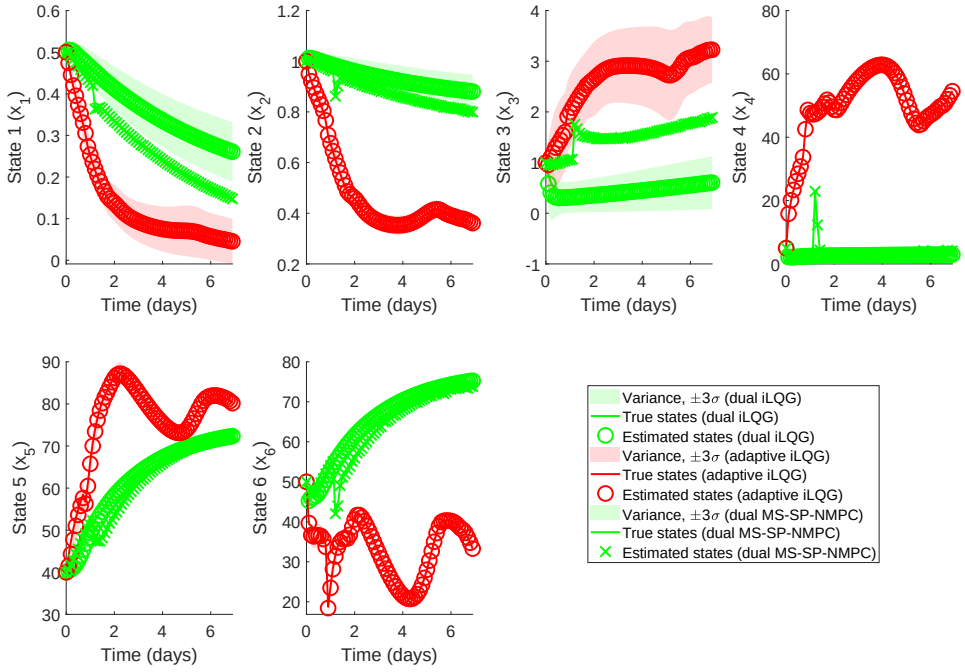


Figure 4.2: State comparison between dual and adaptive iLQG and dual MS-SP-NMPC on AM2 anaerobic digestion model with two uncertain parameters.

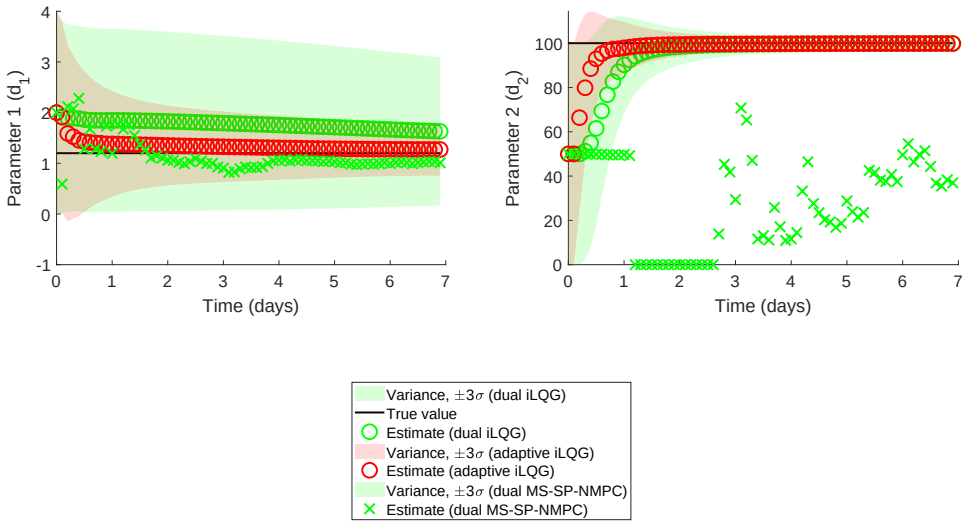


Figure 4.3: Parameter comparison between dual and adaptive iLQG and dual MS-SP-NMPC on AM2 anaerobic digestion model with two uncertain parameters.

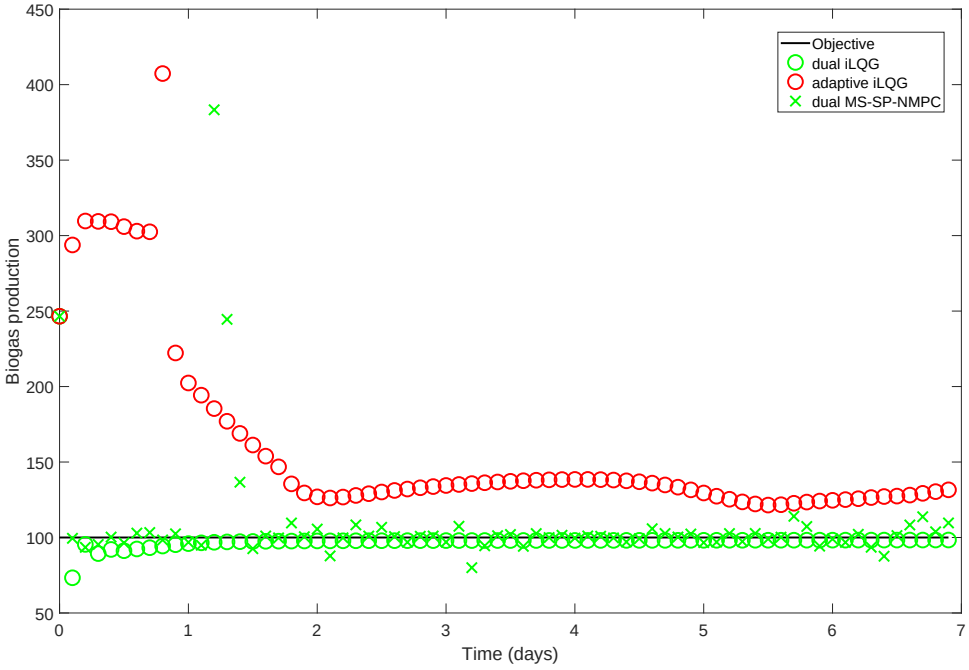


Figure 4.4: Biogas production comparison between dual and adaptive iLQG and dual MS-SP-NMPC on AM2 anaerobic digestion model with two uncertain parameters.

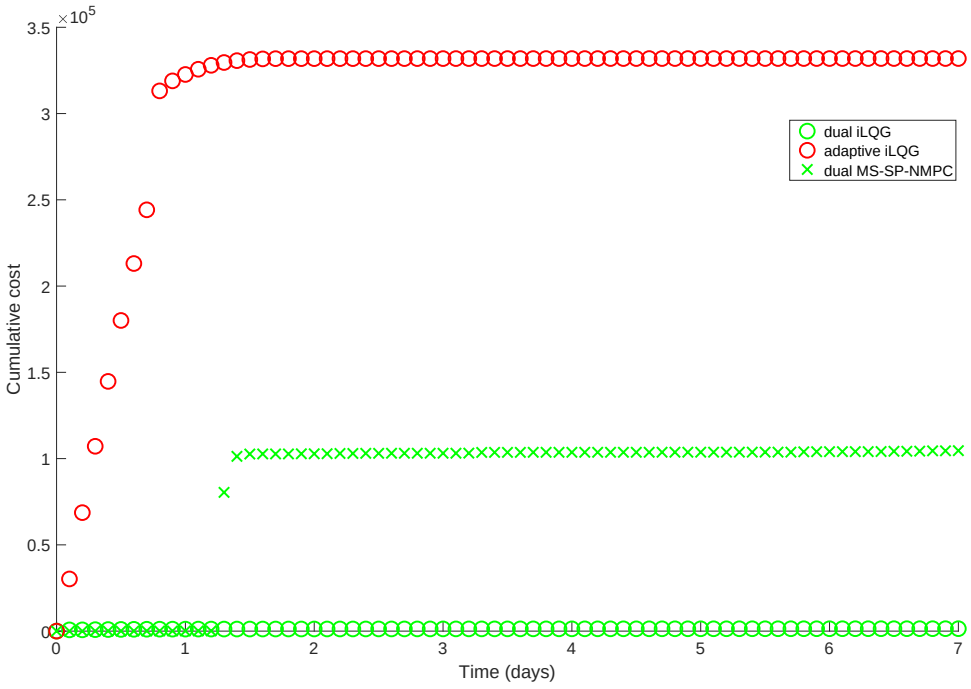


Figure 4.5: Cost comparison between dual and adaptive iLQG and dual MS-SP-NMPC on AM2 anaerobic digestion model with two uncertain parameters.

4.3.2 Seventeen-parameter comparison with adaptive iLQG

To show the ability of dual iLQG to handle many uncertain parameters, it was compared to adaptive iLQG for the AM2 system with seventeen uncertain parameters. Dual MS-SP-NMPC was not included in this comparison as it is unable to handle more than two uncertain parameters.

The true and estimated initial values of the state vector, $\mathbf{x} = [X_1, X_2, S_1, S_2, Z, C]$ are both $[0.5, 1, 1, 5, 40, 50]^\top$ and the state are all constrained to be greater than 0. The controls, $\mathbf{u} = [q_a, q_b, q_c]$ are all constrained to be between 0 and 1. The true and initial estimated values of the uncertain parameters $\mathbf{d}$ can be seen in Table 4.2, with a covariance matrix of

$$\text{diag}[0.16, 5.6, 0.06, 9.6, 7282, 1, 1111, 625, 2.8, 1.8, 2844.5, 69.5, 11, 2.8, 711, 1111, 0.1].$$

This system was simulated over 70 time steps that are 0.1 days long. Only states $[S_1, S_2, C]$ were measured with a noise term of $\mathcal{G}$ of $0.5I$. The multiplicative noise terms $\mathcal{F}$ for the dual iLQG state and parameter dynamics were set to $10^{-15}I$ to be comparable with the lack of MS-SP-NMPC to handle noise. The cost function was as given for the two-parameter example, shown in equation (4.19).

The results for the two iLQG algorithms' control trajectories are shown in Figure 4.6. As with

Parameter	True value	Estimated value
$\bar{\mu}_1$	1.2	2
K_{S_1}	7.1	10
$\bar{\mu}_2$	0.74	2
K_{S_2}	9.28	15
K_{I_2}	256	350
S_1^A	3	2.5
S_2^A	100	80
Z^A	75	50
C^A	5	2.5
S_1^B	4	2.5
S_2^B	160	80
Z^B	25	50
C^B	10	2.5
S_1^C	5	2.5
S_2^C	80	80
Z^C	100	50
C^C	1	2.5

Table 4.2: Parameter values for the AM2 comparison with seventeen uncertainties

the previous example, these results by themselves did not indicate the relative performance of the algorithms as there were no terms in the cost function to explicitly penalize or encourage the use of the feedstock controls.

Figure 4.7 shows the true and estimated values for the anaerobic digestion system states. The two iLQG algorithms had similar values for the first four states, but there was a significant difference for the last two states, as the dual controller had higher total concentrations of alkalinity and inorganic carbon.

Looking at the parameter estimates in Figure 4.8, the two iLQG algorithms gave similar results for most of the parameters with only a few that approached the true parameter values. The most notable exception was parameter 16, Z_C, where the dual controller's estimate quickly approached the true value while the adaptive controller lagged behind. Other than this, there were several instances where dual iLQG had a better estimate than adaptive iLQG and several examples where the opposite was true. It is important to keep in mind that parameter identification is not the objective of these algorithms and that a feature of dual control is being selective in identifying parameters to such a level that the objective function can be minimized.

For the plot of biogas production in Figure 4.9, the dual controller approached the desired production level by oscillating around the setpoint while the adaptive iLQG did not demonstrate that behaviour. After performing worse than adaptive iLQG for several time steps in the first half of the simulation, dual iLQG consistently outperformed adaptive iLQG over the second half of the simulation.

These results are reflected in the plot of the objective function in Figure 4.10, where it can be seen that the first and third time steps are where dual iLQG gained the most advantage over adaptive iLQG. Overall, the dual iLQG algorithm's total cost was 18.6% lower than that of the adaptive iLQG algorithm.

With the two- and seventeen-parameter comparisons of the AM2 anaerobic digestion model being the same other than the number of parameters, the simulation times for these examples can give a sense of how the dual iLQG algorithm overcomes the curse of dimensionality. The simulation times for dual iLQG in this application with two- and seventeen-parameters were 1125 and 1256 seconds respectively. The fact that these simulation times are close demonstrates that the expected exponential relationship between problem size and complexity does not hold here, and that dual iLQG can overcome Bellman's curse of dimensionality.

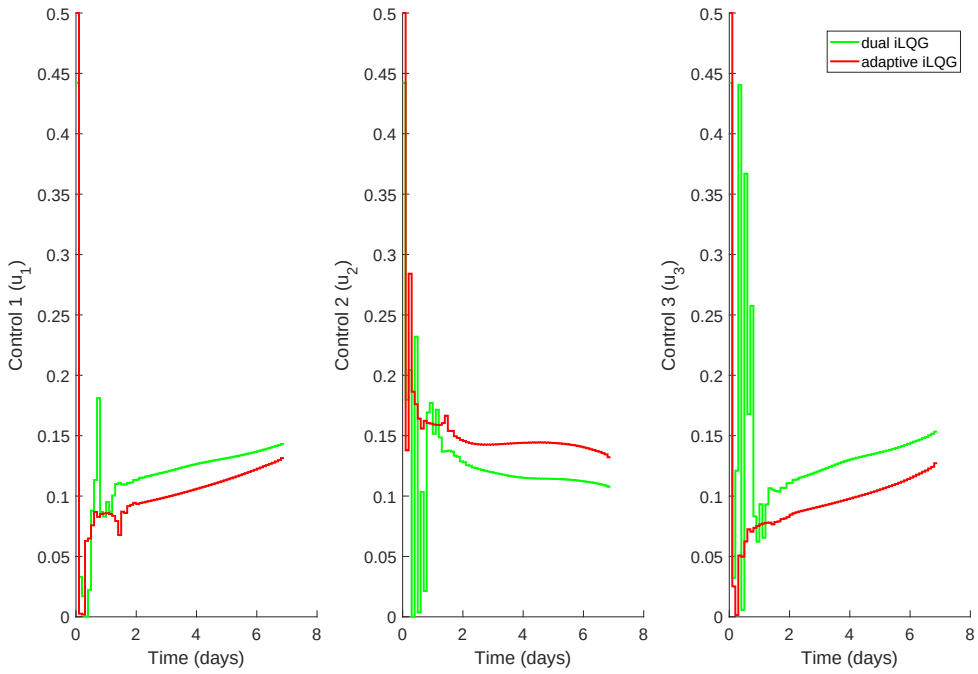


Figure 4.6: Control comparison between dual and adaptive iLQG on AM2 anaerobic digestion model with seventeen uncertain parameters.

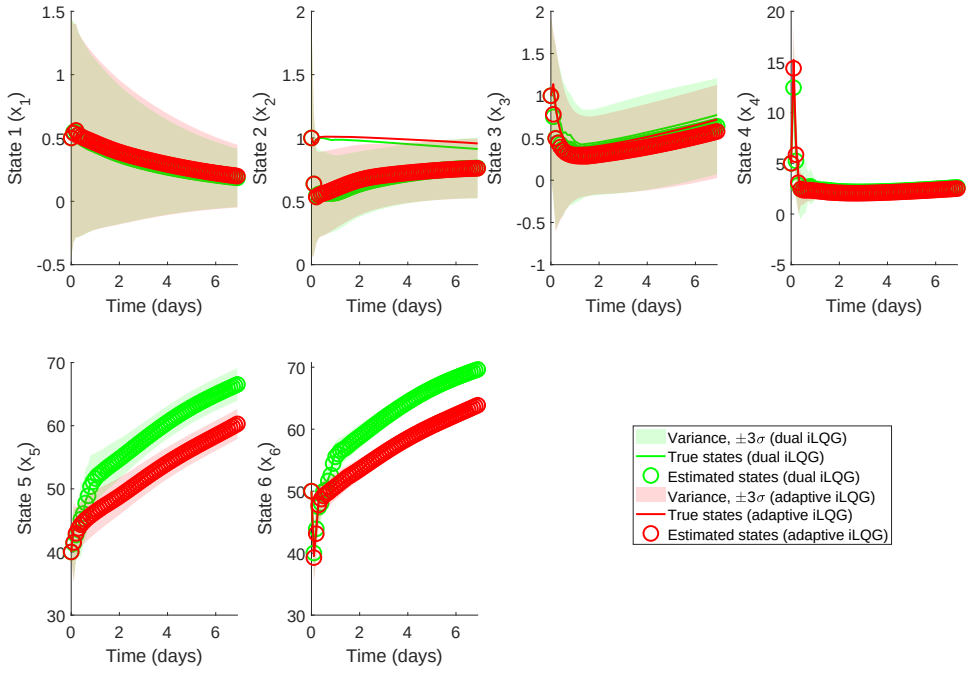


Figure 4.7: State comparison between dual and adaptive iLQG on AM2 anaerobic digestion model with seventeen uncertain parameters.

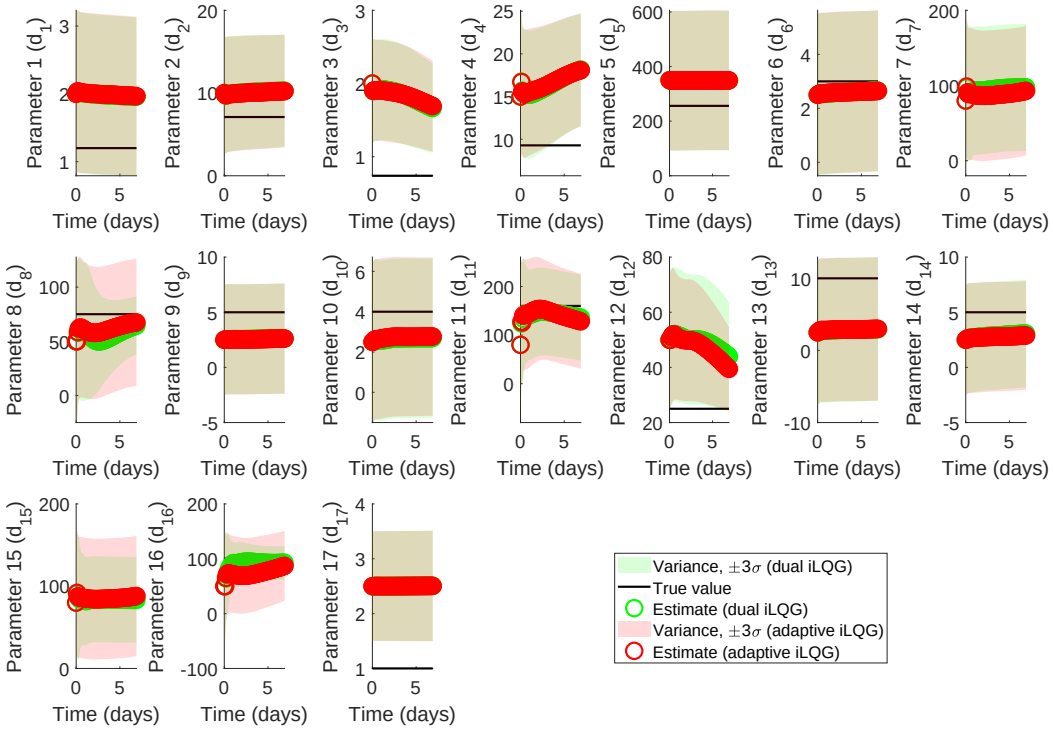


Figure 4.8: Parameter comparison between dual and adaptive iLQG on AM2 anaerobic digestion model with seventeen uncertain parameters.

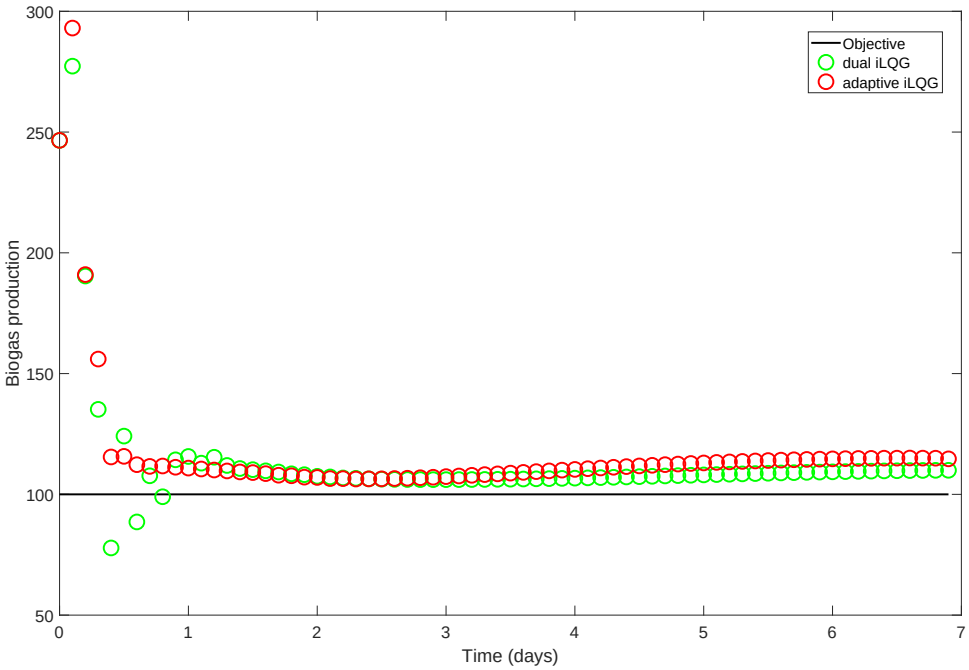


Figure 4.9: Biogas production comparison between dual and adaptive iLQG on AM2 anaerobic digestion model with seventeen uncertain parameters.

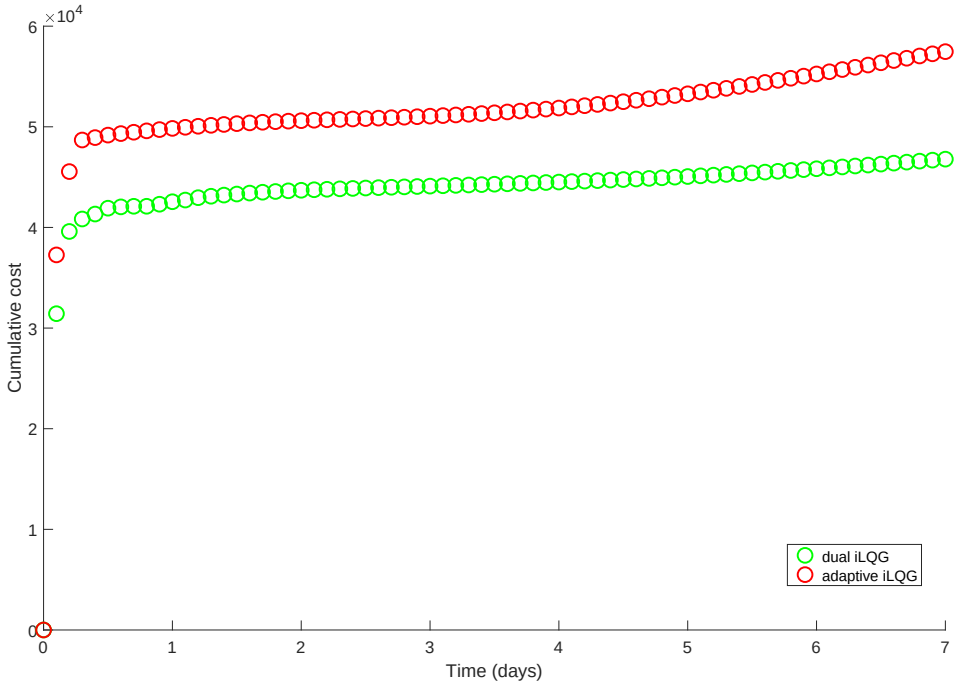


Figure 4.10: Cost comparison between dual and adaptive iLQG on AM2 anaerobic digestion model with seventeen uncertain parameters.

4.4 Summary

This application demonstrated how this work's dual iLQG controller can be used to control an uncertain nonlinear biochemical system. To allow for dual iLQG to be compared with dual MS-SP-NPMC, the AM2 anaerobic digestion system was simplified to have only two uncertain parameters. Although dual iLQG significantly outperformed dual MS-SP-NPMC in this two-parameter case, and dual MS-SP-NMPC was unable to handle the following seventeen-parameter case. In this seventeen-parameter case, dual iLQG outperformed adaptive iLQG by 18.6%. With similar simulation times between the two- and seventeen-parameter cases, the ability of dual iLQG to overcome Bellman's curse of dimensionality has been demonstrated for this application.

Chapter 5

Application to the control of the COVID-19 outbreak

5.1 Motivation

The control of the COVID-19 pandemic continues to represent an enormous challenge for governments all over the world. Although policies to limit deaths and hospitalizations have been determined, such as enforcing mask use and lockdowns, these policies have social and economic costs, and precisely how effective these policies are is uncertain [30]. Additionally, there are uncertainties associated with the dynamics of the virus's spread through populations [54]. Having an objective of balancing deaths and hospitalizations with social and economic costs in a dynamic system that has uncertain parameters makes COVID-19 a suitable application for dual control. As with the anaerobic digestion application, other control approaches such as robust or machine learning methods could also be applied to this system, but the focus of this thesis is dual control.

The purpose of using COVID-19 as an application for dual control here is not to model the spread of the virus through a real population or to suggest that dual control could have saved lives, but to apply dual control to a complex nonlinear system that people understand. To that end, where available, values for the states, parameters, and cost-weighting factors from the literature were used, and where not, illustrative values were used.

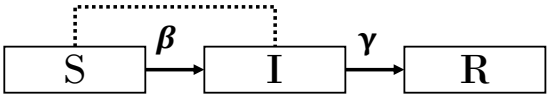


Figure 5.1: SIR model compartmental diagram [43].

5.2 System model

To model the spread of infectious diseases through populations, compartmental models have been used since 1927 [43]. These models divide a population into a series of compartments that represent stages of the disease and then describe how these groups change over time. One of the simplest of these compartment models was the SIR model [43] as shown in Figure 5.1, where a population is divided into being Susceptible (S), Infected (I), or Removed (R) (that is, deceased). The movement of the population through these states can then be represented graphically as arrows between each compartment (state), and these population flows can then be described with equations based on the states themselves as well as parameters and controls. These parameters generally represent infection and fatality rates for different populations, and the controls are methods of influencing these dynamics. For instance, the SIR model can be expressed as

$$\dot{S} = -\frac{\beta SI}{N}, \tag{5.1}$$

$$\dot{I} = \frac{\beta SI}{N} - \gamma I, \tag{5.2}$$

$$\dot{R} = \gamma I, \tag{5.3}$$

where β is the transmission rate between the infected and susceptible populations, γ is the inverse infectious period, and N is the total population ($S + I + R = N$).

Note that in the SIR model, having a population that is recovered is not considered, and neither is a hospitalized population that may have a different transmission rate.

These compartmental models have been tailored to better represent the dynamics of the COVID-19 virus, with different compartments or states being considered by different researchers. In [30], eight compartments are considered: Susceptible, Infected (those that are asymptomatic, infected, and undetected), Diagnosed (those that are asymptomatic, infected, and detected), Ailing (those that are symptomatic, infected, and undetected), Recognized (those that are symptomatic, infected, and detected), Threatened (those that are acutely symptomatic, infected, and detected), Healed (either after being detected or not, and assumed immune after being

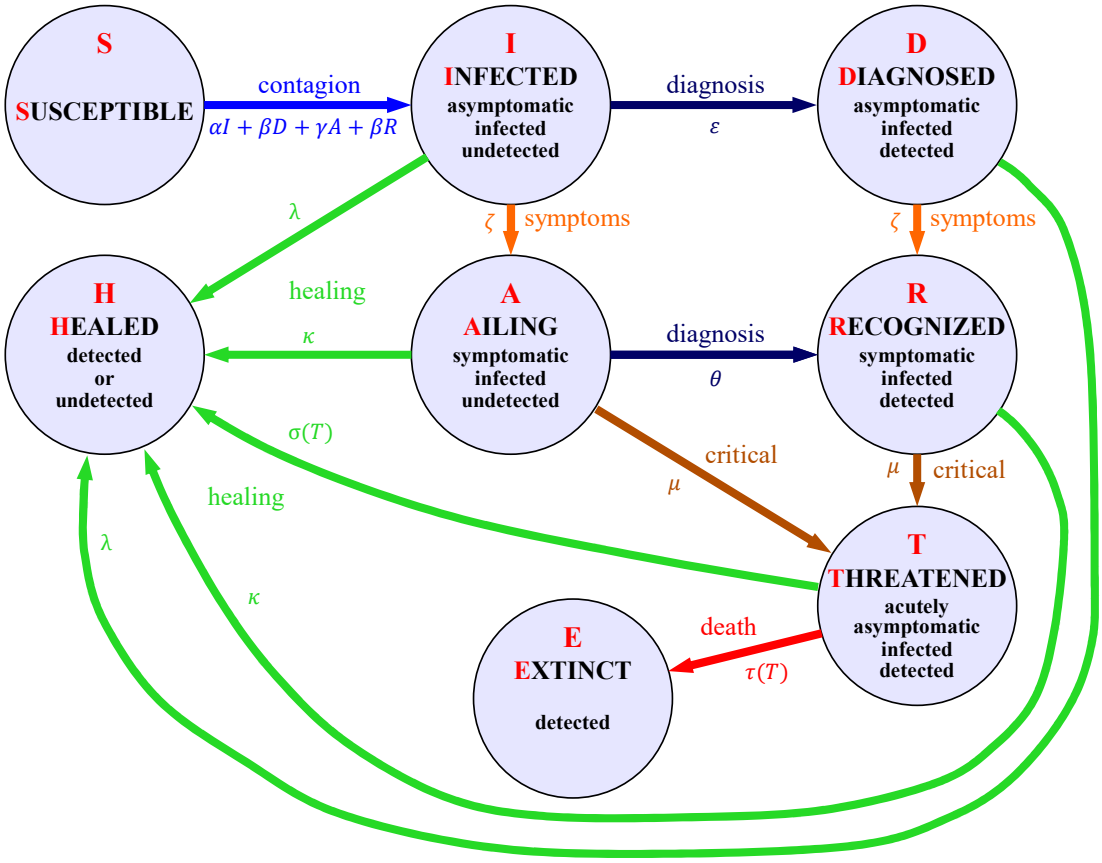


Figure 5.2: SIDARTHE model compartmental diagram [30].

infected), and Extinct (and assumed to be detected), giving the SIDARTHE model shown in Figure 5.2.

In the SIDARTHE model, the infected populations other than the threatened population infect the susceptible population with different rates of transmission. Once infected, there are 5 different transitions between populations considered, shown in different colours in Figure 5.2: developing symptoms, getting diagnosed, getting healed, becoming critical, or dying. With the parameters shown in Figure 5.2 describing these transitions between the states, the SIDARTHE

Parameter	Rate of	of state	by state	moving to state
α	infection	S	I	I
β	infection	S	D and R	I
γ	infection	S	A	I
ϵ	diagnosis	I	-	D
ζ	developing symptoms	I and D	-	A and R
θ	diagnosis	A	-	R
μ	developing acute symptoms	A and R	-	T
λ	recovery	I and D	-	H
κ	recovery	A and R	-	H
$\sigma(T)$	recovery	T	-	H
$\tau(T)$	mortality	T	-	E

Table 5.1: Parameters for the SIDARTHE model

model can be expressed as

$$\dot{S} = -S\left(\alpha I + \beta D + \gamma A + \beta R\right), \tag{5.4}$$

$$\dot{I} = S\left(\alpha I + \beta D + \gamma A + \beta R\right) - \left(\epsilon + \zeta + \gamma\right) I, \tag{5.5}$$

$$\dot{D} = \epsilon I - \left(\zeta + \lambda\right) D, \tag{5.6}$$

$$\dot{A} = \zeta I - \left(\theta + \mu + \kappa\right) A, \tag{5.7}$$

$$\dot{R} = \zeta D + \theta A - \left(\mu + \kappa\right) R, \tag{5.8}$$

$$\dot{T} = \mu A + \mu R - \left(\sigma\left(T\right) + \tau\left(T\right)\right) T, \tag{5.9}$$

$$\dot{H} = \lambda I + \lambda D + \kappa A + \kappa R + \sigma\left(T\right) T, \tag{5.10}$$

$$\dot{E} = \tau\left(T\right) T, \tag{5.11}$$

where the description of the parameters can be found in Table 5.1.

In this model, the recovery and mortality rates of the Threatened state are modelled as being dependent on the Threatened state to represent the impact of the health care system being overwhelmed. In [30], this effect was achieved in a two-step process, whereby a model was created where the Threatened population was divided into those in the limited-capacity intensive care unit (ICU) and those not, and then this model was simplified to maintain the eight states described above. The compartmental diagrams for these two steps can be seen in Figure 5.3.

Defining T_1 as the Threatened population that do not require ICU treatment and T_2 as those that do, and assuming that there are no transfers between these two populations, the dynamics of

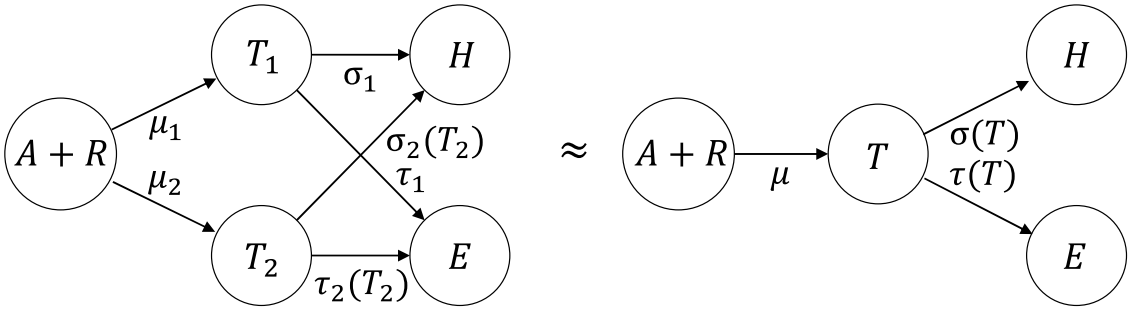


Figure 5.3: Partial compartmental diagram for considering the impact of an overwhelmed ICU [30].

these states can be represented as

$$\dot{T}_1 = \mu_1 (A+R) - (\sigma_1 + \tau_1) T_1, \tag{5.12}$$

$$\dot{T}_2 = \mu_2 (A+R) - (\sigma_2 (T_2) + \tau_2 (T_2)) T_2, \tag{5.13}$$

where σ_1 and τ_1 are independent parameters and σ_2 and τ_2 are dependent on T_2. This model is approximated to a lumped model as shown on the right of Figure 5.3, for a defined ICU capacity of T_{ICU}, by assuming that if $T_2 \leq T_{ICU}$,

$$\tau(T) = \frac{\mu_1}{\mu}\tau_1 + \frac{\mu_2}{\mu}\tau_2, \tag{5.14}$$

$$\sigma(T) = \frac{\mu_1}{\mu}\sigma_1 + \frac{\mu_2}{\mu}\sigma_2, \tag{5.15}$$

but if $T_2 > T_{ICU}$, $\tau(T)$ increases to τ_{crit} for the remaining T_2 population who require ICU treatment but cannot access it, and the recovery rate for this group also drops to 0. The $\tau(T)T$ and $\sigma(T)T$ terms can therefore be represented as

$$\tau(T)T = \frac{\mu_1}{\mu}\tau_1 T + \max\left\{\frac{\mu_2}{\mu}\tau_2 T, \tau_2 T_{ICU} + \tau_{crit}\left(\frac{\mu_2}{\mu}T - T_{ICU}\right)\right\}, \tag{5.16}$$

$$\sigma(T)T = \frac{\mu_1}{\mu}\sigma_1 + \sigma_2 \min\left\{\frac{\mu_2}{\mu}, T_{ICU}\right\}, \tag{5.17}$$

and used in equations (5.9) to (5.11).

To implement controls in this model, public health policies are seen to have a direct influence

on the infection rates α and γ, and this relationship can be modelled as

$$\alpha(t) = \alpha_{\max} + (\alpha_{\min} - \alpha_{\max})\, u(t), \tag{5.18}$$

$$\gamma(t) = \gamma_{\max} + (\gamma_{\min} - \gamma_{\max})\, u(t), \tag{5.19}$$

where $u(t)$ is constrained to [0, 1] and can be used to vary the infection rates between minimum values of $\alpha_{\min}$ and $\gamma_{\min}$ with $u = 1$ and maximum values of $\alpha_{\max}$ and $\gamma_{\max}$ with $u = 0$.

5.2.1 SIDARTHE model limitations

Although the SIDARTHE model can capture the major aspects of the dynamics of COVID-19, there are several limitations to this model. First of all, this model represents the population as static, other than deaths due to COVID-19. SIDARTHE does not include population changes due to travel, births, or non-COVID-related deaths. A more complex model that did include non-COVID-related deaths would also be an interesting application for dual control.

Additionally, the SIDARTHE model only represents the public health policies as a single control input, lumping the impact of these policies into a single value representing the severity of the restrictions. Although this makes the implementation of the model much easier, it would be difficult for health agencies to get precise recommendations from such a lumped term. Additionally, this single control action limits the potential probing that a dual control method could implement, as in reality multiple policies can be varied over time. For instance, media campaigns, enforcing social distancing and mask use, performing asymptomatic testing, performing symptomatic testing, quarantining of positive cases, increasing non-ICU hospital resources, and increasing ICU resources could all be considered independent controls, and the extension of the SIDARTHE model to include these controls will be discussed in the next section.

5.2.2 Changes to the SIDARTHE model

To extend the SIDARTHE model to have separate control inputs representing different types of public health policies, a similar approach to that of equations (5.18) and (5.19) was used. The public health policies that were considered were media campaigns (u_1), enforcing social distancing and mask use (u_2), performing asymptomatic testing (u_3), performing symptomatic testing (u_4), quarantining of positive cases (u_5), increasing non-ICU hospital resources (u_6), and increasing ICU resources (u_7). Since many of these public health policies influence more than one parameter in the SIDARTHE model to varying levels, effectiveness parameters were introduced. For

instance, both media campaigns and enforcing social distancing and mask use will lower α, but the media campaigns may do so less effectively. The impact of the control inputs on the SIDARTHE model parameters can therefore be represented as

$$\begin{aligned}
\alpha &= \alpha_{\max} + (\alpha_{\min} - \alpha_{\max})\left(\eta_{\alpha_1} u_1 + \eta_{\alpha_2} u_2\right), && (5.20)\\
\beta &= \beta_{\max} + (\beta_{\min} - \beta_{\max})\left(\eta_{\beta_1} u_1 + \eta_{\beta_2} u_2 + \eta_{\beta_5} u_5\right), && (5.21)\\
\gamma &= \gamma_{\max} + (\gamma_{\min} - \gamma_{\max})\left(\eta_{\gamma_1} u_1 + \eta_{\gamma_2} u_2\right), && (5.22)\\
\epsilon &= \epsilon_{\min} + (\epsilon_{\max} - \epsilon_{\min})\left(\eta_{\epsilon_1} u_1 + \eta_{\epsilon_3} u_3\right), && (5.23)\\
\theta &= \theta_{\min} + (\theta_{\max} - \theta_{\min})\left(\eta_{\theta_1} u_1 + \eta_{\theta_4} u_4\right), && (5.24)\\
\sigma_1 &= \sigma_{1_{\min}} + (\sigma_{1_{\max}} - \sigma_{1_{\min}})\, u_6, && (5.25)\\
\sigma_2 &= \sigma_{2_{\min}} + (\sigma_{2_{\max}} - \sigma_{2_{\min}})\, u_7, && (5.26)\\
\tau_1 &= \tau_{1_{\max}} + (\tau_{1_{\min}} - \tau_{1_{\max}})\, u_6, && (5.27)\\
\tau_2 &= \tau_{2_{\max}} + (\tau_{2_{\min}} - \tau_{2_{\max}})\, u_7, && (5.28)\\
T_{ICU} &= T_{ICU_{\min}} + (T_{ICU_{\max}} - T_{ICU_{\min}})\, u_7, && (5.29)
\end{aligned}$$

where the η terms are effectiveness factors for the controls, describing the influence they have on the SIDARTHE model parameters. For each of the SIDARTHE model parameters, the η terms sum to 1 such that the maximum and minimum values are maintained. For instance,

$$\eta_{\beta_1} + \eta_{\beta_2} + \eta_{\beta_5} = 1, \tag{5.30}$$

and therefore when these effectiveness factors are used as uncertain parameters in dual iLQG, one of the factors from each set can be determined from the others.

5.3 Results

5.3.1 Two-parameter comparison with dual MS-SP-NMPC

Since the dual MS-SP-NMPC code was only able to handle two parameters, it was compared with dual and adaptive iLQG for the original single-control SIDARTHE model. The uncertain parameters were chosen to be $\alpha_{\min}$ and $\gamma_{\min}$, as the extent to which the spread of COVID-19 can be reduced by imposing full restrictions is a critical factor in determining the trade-off between reducing cases and the socio-economic impacts of restrictions. The values of the total population

and the parameters used in these simulations were taken from [30], where the COVID-19 outbreak in Germany was modelled.

The states $\mathbf{x} = [S, I, D, A, R, T, H, E]$ were constrained to $[0, 82999999]$ and the true and estimated initial values of the state vector were both $[82998999, 1000, 0, 0, 0, 0, 0, 0]^\top$. The control input $\mathbf{u} = u(t)$ was constrained to $[0, 1]$. The true value of the constant parameter vector $\mathbf{d} = [\alpha_{\min}, \gamma_{\min}]$ was $[0.0422, 0.0422]$, while its initial estimate was $[0.15, 0.1]$ with a covariance matrix of $[0.0025, 0; 0, 0.0011]$. This system was simulated over 40-time steps that were 1.0 days long, with a rolling horizon approach. The Diagnosed, Recognized, Threatened, and Extinct states were measured with a noise term $\mathcal{G}$ of diag($[10, 7.5, 5, 2.5]$), and as the MS-SP-NMPC did not consider the states to be uncertain, the process noise scaling matrix $\mathcal{F}$ for the dual iLQG state and parameter dynamics were set to $10^{-15} I_{10}$ and the noise was not implemented in the true system in the outer loop. The cost function was set as

$$J(\mathbf{x}, \mathbf{u}) = c_x \mathbf{x} + c_u \mathbf{u}^2, \tag{5.31}$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the control vector, $c_x = [0, 0, 0, 0, 0, 0.0033, 0, 0.0267]$, and $c_u = 10$. This cost function was very similar to the one used in [53] for the suppression of COVID-19, but c_u was increased here to increase the cost of imposing restrictions. Additionally, this is a representative cost function; its values would have to be set by policymakers, but the benefits illustrated in this work are robust to various cost functions. The vector c_x places a cost on the Threatened and Extinct populations.

Instead of using a shrinking horizon approach for the SIDARTHE simulations, a rolling horizon approach was used to match how this problem would be approached by governments. Since COVID-19 is an ongoing issue, the control approach should reflect that, and ignoring this fact was shown to cause unexpected results from the controllers. When using a shrinking horizon, all of the controllers had knowledge of the fixed terminal simulation time, and as that terminal time approached, they changed their behaviour. Due to the cost of the control actions and the propagation delay between infected individuals becoming hospitalized or dying, the controllers would drop all restrictions, or controls, to zero near the end of the simulation to reduce costs, as there were no costs associated with the soon-to-be spiking hospitalization levels after the simulation ended. In essence, the controllers were behaving as if they knew the world was ending at a particular time, and there was no need for restrictions. As this behaviour was not representative of the approach governments would likely take, a rolling horizon approach was used,

where even during the last time step of the simulation, the controller was still trying to minimize costs over a time horizon that extended past the final time step. This rolling horizon approach gave the desired controller behaviour and was used for all of the SIDARTHE simulations.

Figure 5.4 shows the controls for the adaptive iLQG, dual iLQG, and dual MS-SP-NMPC algorithms. Here, the dual MS-SP-NMPC algorithm's control is plotted against the right y-axis in order to see the variations in its small control actions. Both of the iLQG algorithms used the full range of the constrained control, while the MS-SP-NMPC algorithm's control remained near zero for the entire simulation duration. The dual iLQG controls use a high degree of restrictions early on, with a small variation that likely helped with parameter identification, before tapering off over the 40 day time frame.

The effect of these controls on the system states can be seen in Figure 5.5, where the near lack of control from the MS-SP-NMPC algorithm is clear from the rapid increase in case numbers, while the iLQG algorithms did better at keeping the case numbers under control, with dual iLQG performing better than adaptive iLQG.

The estimation of the two uncertain parameters is shown in Figure 5.6. Here, with the control hardly used, the dual MS-SP-NMPC algorithm's estimates remained at the initial values, while the iLQG algorithms converged to the true values in a similar manner.

Finally, the plot of the cumulative cost for each algorithm is shown in Figure 5.7. The MS-SP-NMPC's cost increased each time step with the increasing case counts, remaining lower than the cost of the iLQG algorithms until after the 30th day. Comparing the total costs, dual iLQG outperformed adaptive iLQG and dual MS-SP-NMPC by 28% and 66% respectively.

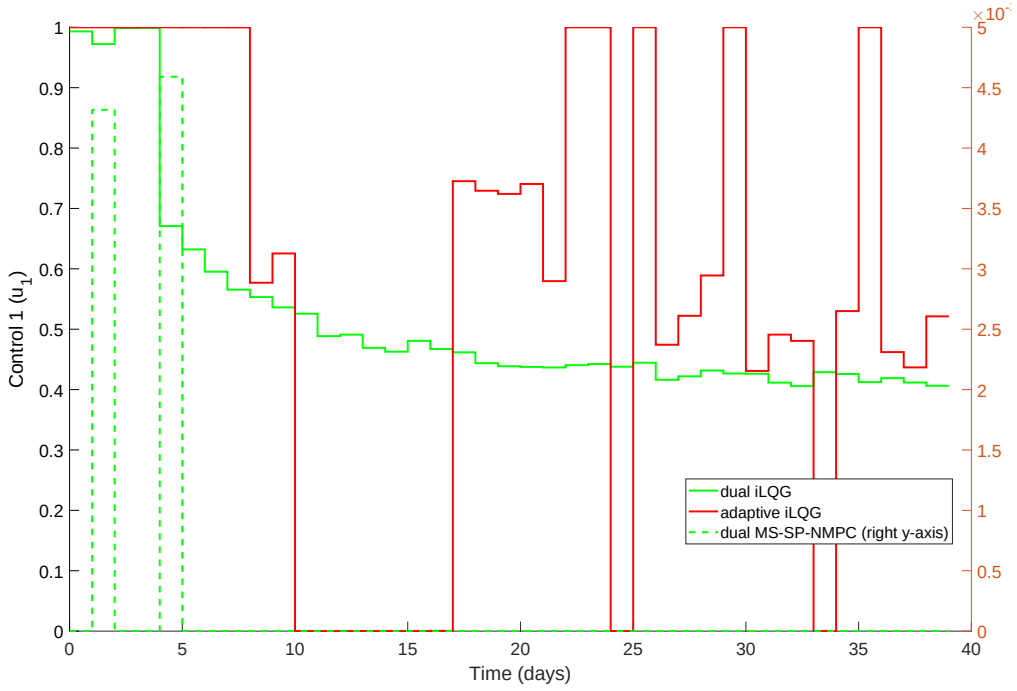


Figure 5.4: Control comparison between dual and adaptive iLQG and dual MS-SP-NMPC on SIDARTHE COVID-19 model with two uncertain parameters.

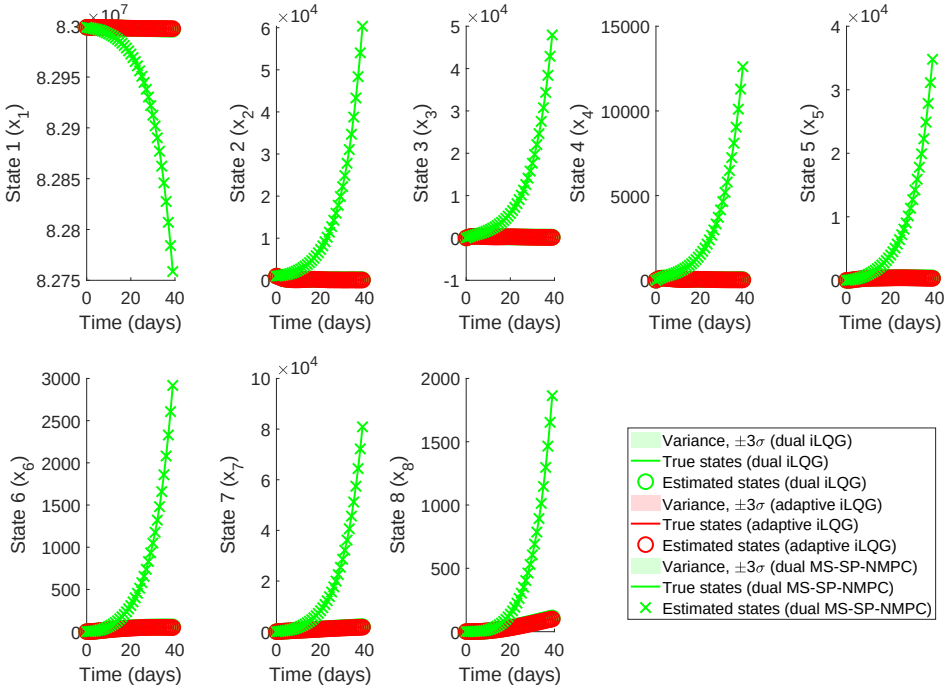


Figure 5.5: State comparison between dual and adaptive iLQG and dual MS-SP-NMPC on SIDARTHE COVID-19 model with two uncertain parameters.

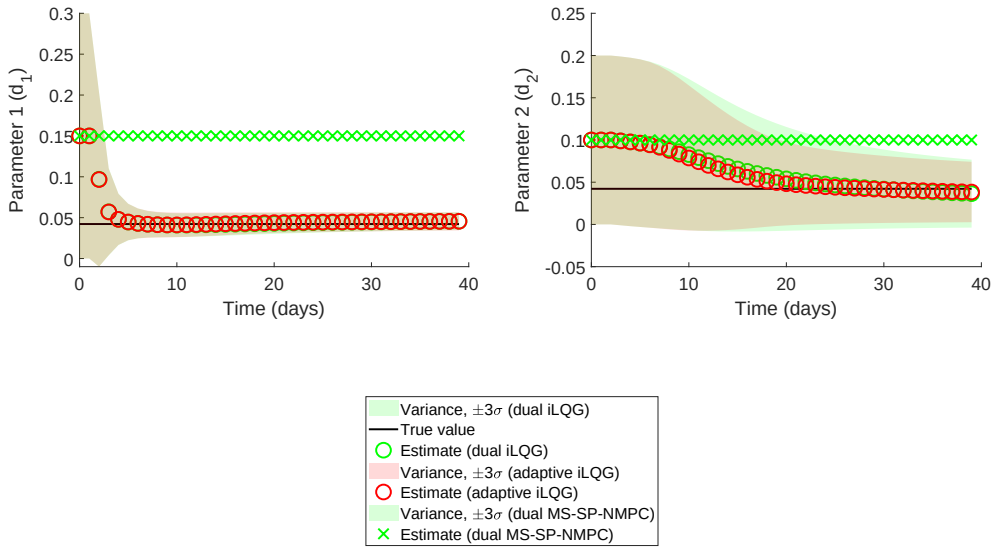


Figure 5.6: Parameter comparison between dual and adaptive iLQG and dual MS-SP-NMPC on SIDARTHE COVID-19 model with two uncertain parameters.

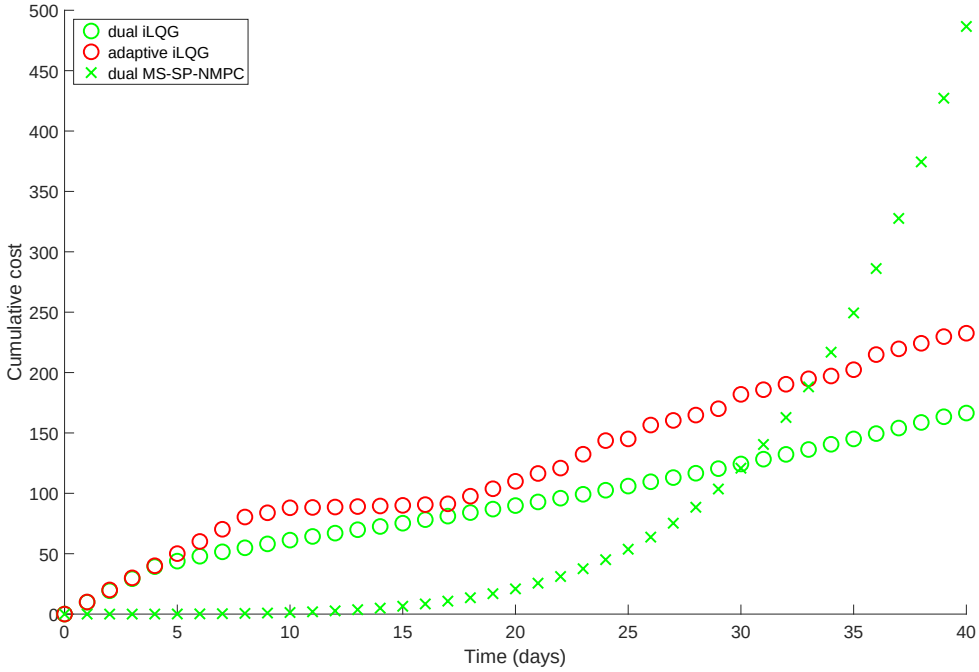


Figure 5.7: Cost comparison between dual and adaptive iLQG and dual MS-SP-NMPC on SIDARTHE COVID-19 model with two uncertain parameters.

Robustness and stability of results

To explore the robustness and stability of these results, the above simulation was run 100 times with each time step of the initial control policy being drawn randomly from a normal

distribution with a mean of 0.5 and a variance of 0.0625. This process was repeated with the iLQG, adaptive iLQG, and dual iLQG algorithms for three different rolling horizon lengths. The impact of the horizon length on the simulation time of these algorithms was of particular interest to compare with similar dual MS-SP-NMPC results. These results are shown in Table 5.2 where the abbreviations A-iLQG and D-iLQG are used for adaptive iLQG and dual iLQG.

	Min. seed cost (90% seeds within)			Median Time (Interquartile range, sec.)		
Horizon	iLQG	A-iLQG	D-iLQG	iLQG	A-iLQG	D-iLQG
5	427 (2.04%)	423 (2.97%)	423 (2.97%)	1.53 (0.0328)	2.05 (0.0607)	2.46 (0.179)
25	230 (63.8%)	169 (57.6%)	172 (59.9%)	150 (84.2)	128 (106)	166 (126)
40	246 (34.6%)	166 (43.6%)	166 (49.2%)	281 (78.6)	297 (151)	293 (177)

Table 5.2: Results from 100 seeded runs of the 2 parameter SIDARTHE COVID-19 model with several rolling horizon lengths

Looking to the left-hand side of Table 5.2, the best solution of the 100 seeds is shown along with a measure of how close 90% of the other solutions were to this minimum in brackets. For instance, the first entry reads that the minimum solution was 427 and that 90% of the 100 seeds had results within 2.04% of this minimum value (therefore the results were very consistent). With a horizon length of 5, all three of these algorithms preformed poorly compared to the longer horizon options shown, but the solutions were concentrated with 90% of them with 2-3% of their respective minimums, therefore they consistently performed poorly. The results for the other two horizon lengths were similar to one another in terms of their minimums, but the longer horizon length of 40 improved the concentration of the solutions around the best solution for all three algorithms. If the horizon length of 5 was too short, causing the algorithms to consistently given poor results, the longer horizon options do not appear to have that issue. To compare these results with individual runs of the dual MS-SP-NMPC algorithm, for horizons of 5 and 25 had results of 486.6 and 509.9, while the algorithm ran out of available memory when trying to run with the full horizon of 40. All three of the iLQG algorithms outperform dual MS-SP-NMPC in this application.

Moving to the right-hand side of Table 5.2, the median simulation times from the 100 seeds are shown along with the interquartile range of the times, both shown in seconds. Here, longer horizons are correlated with longer median run times, and in most cases, increasing complexity of the iLQG algorithm also increases the median run time. The exceptions to this last point are that

for a horizon of 25, the adaptive iLQG algorithm has a lower median time than the iLQG algorithm, and for the horizon of 40 the dual iLQG algorithm has the lowest median time. These simulation times are all within the same order of magnitude, which cannot be said for the dual MS-SP-NMPC algorithm. Individual runs of the dual MS-SP-NMPC algorithm with horizons of 5 and 25 had simulation times of 594.6 and 11301 seconds, while the algorithm ran out of available memory when trying to run with the full horizon of 40. This demonstrates the significant advantage of dual iLQG over dual MS-SP-NMPC.

To get a better idea of these results for the case of with the rolling horizon length of 40, a histogram of the costs and a box plot of the simulation times for each of the algorithms is shown in Figures 5.8 and 5.9.

In Figure 5.8, both adaptive and dual iLQG are an improvement on iLQG, but where the iLQG and adaptive iLQG solutions are strongly skewed to the left, the dual iLQG solutions do not show this same pattern. It may be that for this application the dual probing actions do not result in the cost reductions that the algorithm determined were probable. The dual iLQG algorithm does find a similar minimum solution as the adaptive algorithm though, and the two have a similar variance.

Looking at the simulation time required for these seeded runs in Figure 5.9, the results are similar between the three algorithms. Although the dual algorithm would normally be expected to have the longest times, and it does have the largest extreme times in this case, it actually has the lowest median time of the three algorithms. In this application, the complexity of the model and the length of the horizons appear to have more of an impact on the simulation times than the differences between the algorithms.

5.3.2 Sixteen-parameter comparison with adaptive iLQG

As in the anaerobic digestion application, the model expanded from the two-parameter case to sixteen parameters in this case to compare dual and adaptive iLQG. Dual MS-SP-NMPC was not included in this comparison as it is unable to handle more than two uncertain parameters. The modified SIDARTHE model used for this comparison used 5 control inputs, u_1 to u_5 as described in equations (5.20) to (5.24).

The states $\mathbf{x} = [S, I, D, A, R, T, H, E]$ were constrained to $[0, 82999999]$ and the true and estimated initial values of the state vector are both $[82998999, 1000, 0, 0, 0, 0, 0, 0]^\top$. The 5 control inputs $\mathbf{u} = [u_1, u_2, u_3, u_4, u_5]$ as described in Section 5.2.2 are constrained to $[0, 1]$. The true and

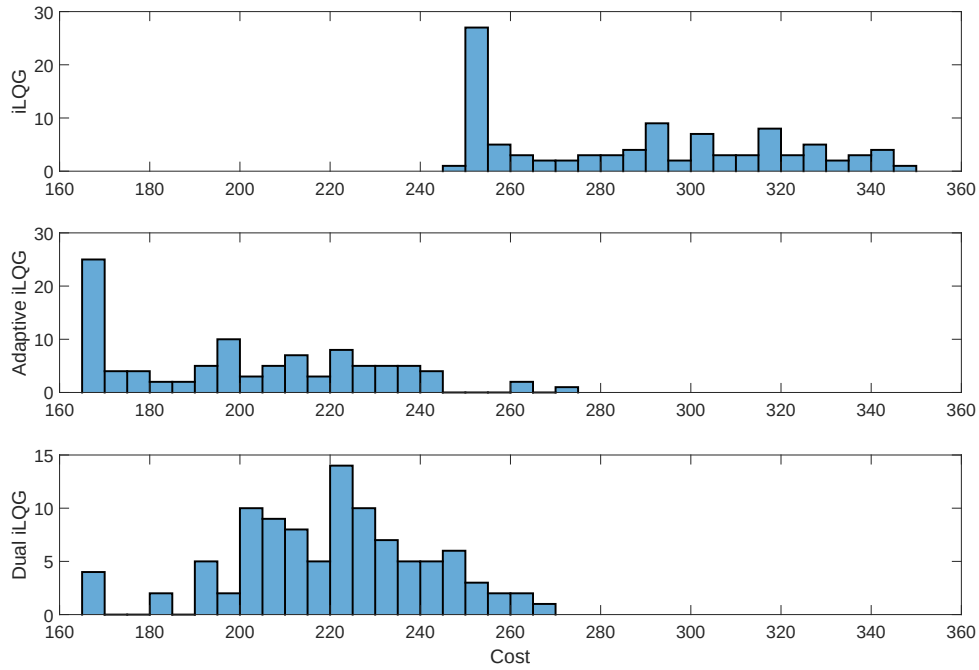


Figure 5.8: Frequency (%) of final cost of 100 seeded runs of the iLQG algorithms for the 2 parameter SIDARTHE COVID-19 model with a rolling horizon length of 40.

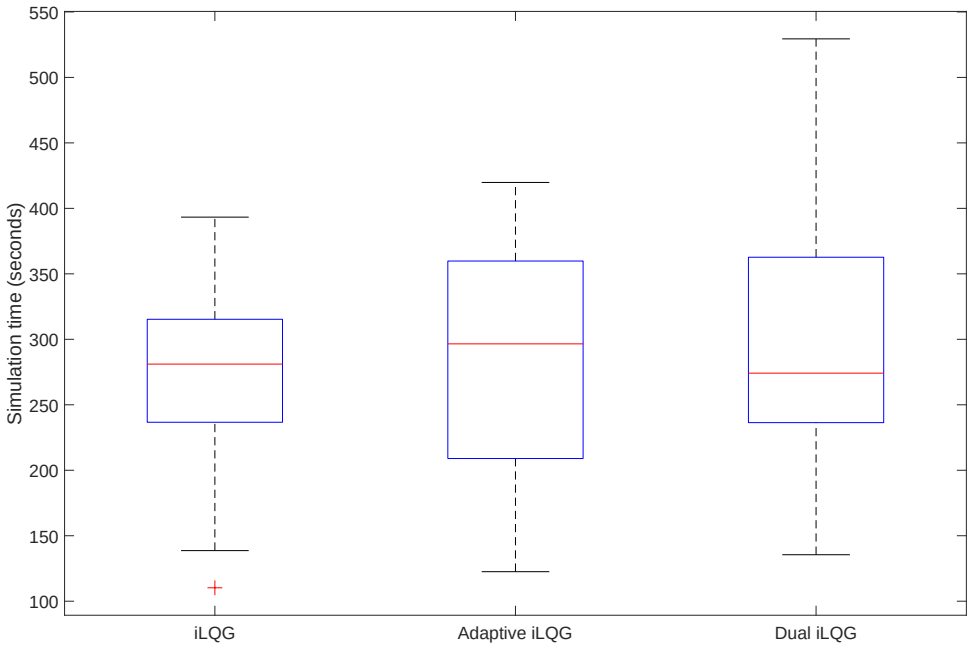


Figure 5.9: Distribution of the simulation time required for 100 seeded runs of the iLQG algorithms for the 2 parameter SIDARTHE COVID-19 model with a rolling horizon length of 40.

Parameter	True value	Estimated value	Source
$\alpha_{\min}$	0.0422	0.05	[30]
$\alpha_{\max}$	0.3614	0.5	[30]
η_{α_1}	0.2	0.5	
$\beta_{\min}$	0.0018	0.005	based on central value from [30]
$\beta_{\max}$	0.015	0.05	based on central value from [30]
η_{β_1}	0.05	0.1	
η_{β_2}	0.25	0.2	
$\gamma_{\min}$	0.0422	0.05	[30]
$\gamma_{\max}$	0.3614	0.5	[30]
η_{γ_1}	0.1	0.5	
$\epsilon_{\min}$	0.0414	0.05	based on central value from [30]
$\epsilon_{\max}$	0.3548	0.5	based on central value from [30]
η_{ϵ_1}	0.1	0.5	
$\theta_{\min}$	0.0414	0.05	based on central value from [30]
$\theta_{\max}$	0.3548	0.5	based on central value from [30]
η_{θ_1}	0.3	0.5	

Table 5.3: Parameter values $\mathbf{d}$ for the SIDARTHE comparison with sixteen uncertainties

initial estimated values of the uncertain parameters $\mathbf{d}$ used in this simulation are shown in Table 5.3. This system is simulated over 40 time steps that are 1.0 days long, with a rolling horizon approach. The Diagnosed, Recognized, Threatened, and Extinct states are measured with a noise term $\mathcal{G}$ of diag($[10, 7.5, 5, 2.5]$), and the noise term $\mathcal{F}$ for dual iLQG was set to 10^{-15} for both the state and parameter dynamics, but the noise was not implemented in the true system in the outer loop to be comparable with the 2 parameter example. The cost function is given as

$$J(\mathbf{x}, \mathbf{u}) = c_x \mathbf{x} + c_u \mathbf{u}^2, \tag{5.32}$$

where $c_x = [0, 0, 0, 0, 0, 0.033, 0, 0.267]$, and $c_u = [0.01, 10, 0.75, 0.75, 0.5]$. As with the 2 parameter example, costs are placed on the Threatened and Extinct populations. As for the cost of using the controls, media campaigns were given a low weight and enforcing social distancing and mask use was given a higher weight compared to the other three controls.

The controls resulting from the adaptive and dual iLQG algorithms are shown in Figure 5.10. Compared to the adaptive controller, the dual controller has generally lower values for the 2nd, 3rd, and 4th controls (enforcing social distancing and mask use, asymptomatic testing, and symptomatic testing) that have higher weights in the cost function, while the 1st and 5th controls are very similar to adaptive iLQG.

Figure 5.11 shows the true and estimated states for each algorithm, along with the covariance of the estimates. Although the dual controller had lower controls, the case counts for dual iLQG

are lower than or similar to the adaptive case. The dual controller has noticeably fewer case counts for the 6th and 8th states (Threatened and Extinct) that have non-zero cost terms.

Due to the significant measurement noise and large covariance of the states in this example, the parameter estimates shown in Figure 5.12 also have large covariances, and many of the estimates do not converge to the true values in the 40 day time horizon.

With generally lower restrictions the dual controller was able to reduce Threatened and Extinct cases of COVID-19, resulting in a total cost that was 5% lower than the adaptive controller, as shown in Figure 5.13. The dual controller had a similar cost to the adaptive controller for the first 6 time steps before having a higher cost for about 10 time steps, and then finally maintained a consistent advantage over the adaptive controller.

With the two- and sixteen-parameter comparisons of the SIDARTHE COVID-19 model being the same other than the number of parameters, the simulation times for these examples can give a sense of how the dual iLQG algorithm overcomes the curse of dimensionality. The simulation times for dual iLQG in this application with two- and sixteen-parameters were 293 and 1484 seconds respectively. The sixteen-parameter time is five times greater than for the two-parameter case which has eight times fewer parameters. This is not the exponential increase in simulation time that would be expected if Bellman's curse of dimensionality held.

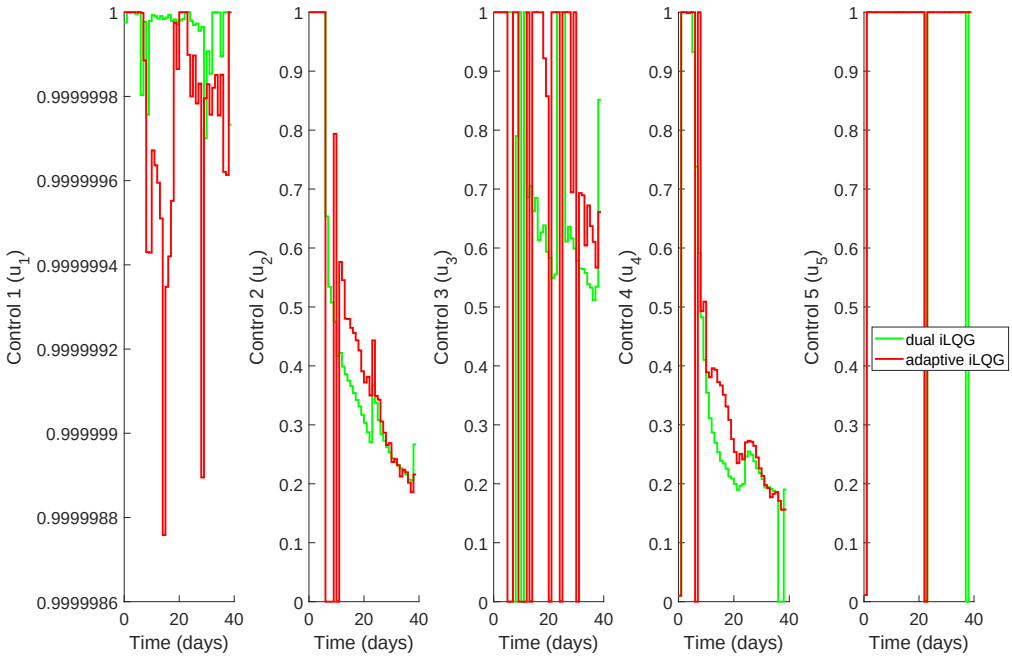


Figure 5.10: Control comparison between dual and adaptive iLQG on modified SIDARTHE COVID-19 model with sixteen uncertain parameters.

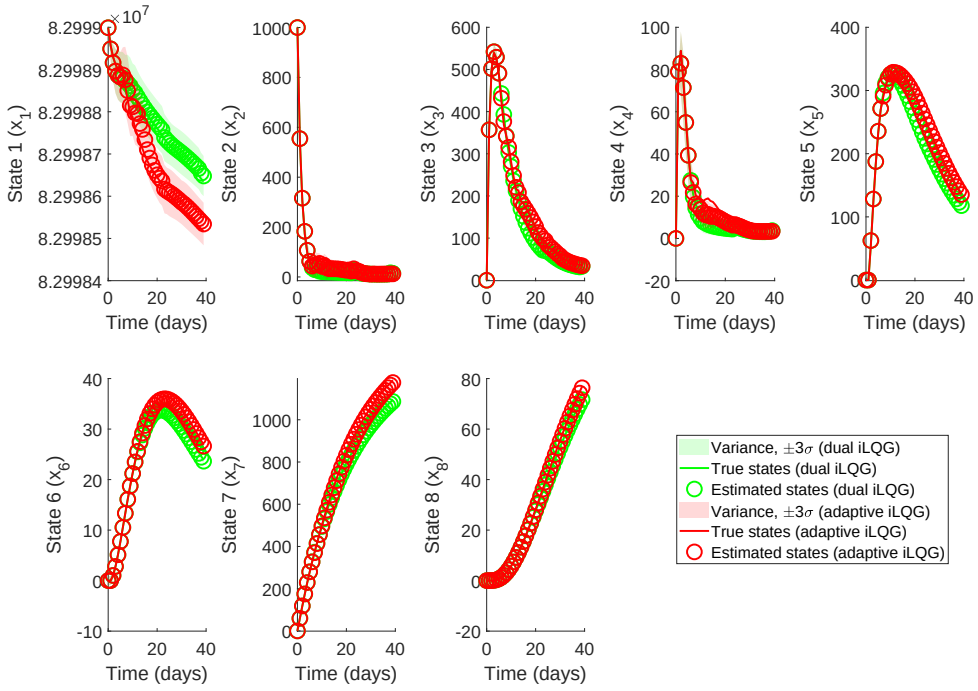


Figure 5.11: State comparison between dual and adaptive iLQG on modified SIDARTHE COVID-19 model with sixteen uncertain parameters.

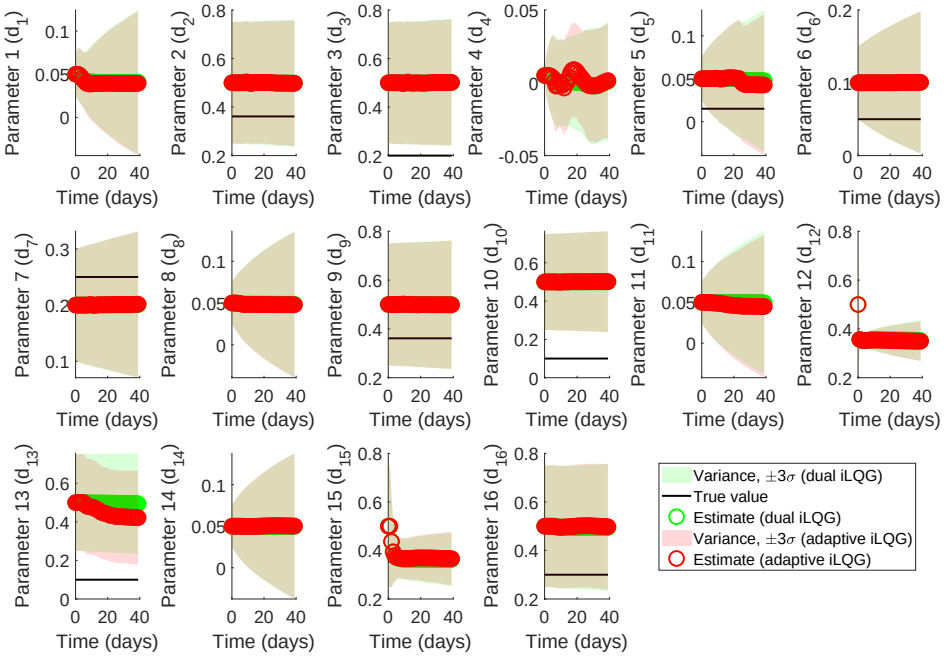


Figure 5.12: Parameter comparison between dual and adaptive iLQG on modified SIDARTHE COVID-19 model with sixteen uncertain parameters.

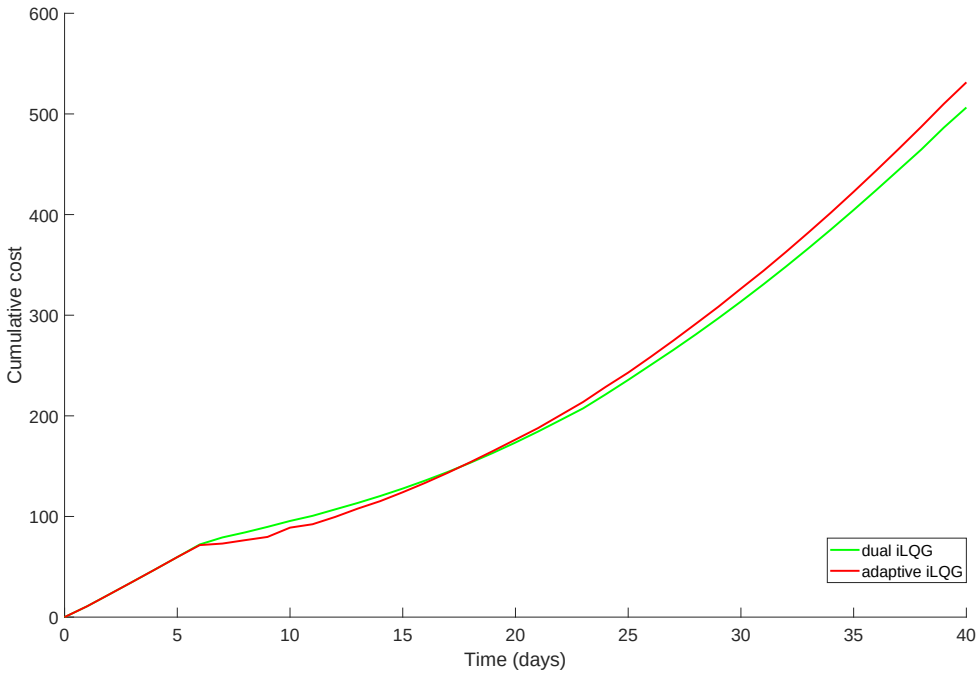


Figure 5.13: Cost comparison between dual and adaptive iLQG on modified SIDARTHE COVID-19 model with sixteen uncertain parameters.

5.4 Summary

This application demonstrates how the dual iLQG controller presented here can be used to inform government policy for uncertain nonlinear systems. To allow for dual iLQG to be compared with dual MS-SP-NPMC, the SIDARTHE model of COVID-19 was simplified to having only two uncertain parameters. Dual iLQG outperformed dual MS-SP-NMPC by 7% and adaptive iLQG by 26%, but dual MS-SP-NMPC was unable to handle the following sixteen-parameter case. With sixteen uncertain parameters, dual iLQG outperformed adaptive iLQG by 5%. With the difference in the simulation times between the two- and sixteen-parameter cases, the ability of dual iLQG to overcome Bellman's curse of dimensionality has been demonstrated.

Chapter 6

Conclusions

The focus of this thesis is the control of systems with uncertain parameters in such a way that the reduction of uncertainty is implicit in the minimization of a given cost function. These dual goals of system identification and system identification are often at odds with each other, and this tension creates a set of three features that characterize dual control. Dual control demonstrates caution, minimizing the magnitude of control actions when uncertainties are high, probing, varying the control actions to gain information about the uncertain parameters, and selectiveness, only seeking to gain information on those parameters which will are likely to cause a reduction in future costs [29].

Although the cost function can be modified to explicitly introduce one or more of these dual features, these explicit approximations of dual control require that the relative importance between these features and the original cost function is determined. Implicit approximations of dual control on the other hand consider the impact of future reductions in parameter uncertainty on the original cost function over the control horizon, increasing the flexibility of the control algorithm to balance system identification and cost minimization when compared to explicit methods. Due to this increased flexibility, implicit methods can give better results than explicit methods [8], generally at a cost of higher computational effort [23].

Several implicit dual control algorithms exist, but each only considers limited control or parameter realizations to make the control problem tractable, or are only applicable for systems with a limited number of states. For example, the optimization problem in MS-SP-NMPC grows exponentially with increasing uncertain parameters, and its current implementation can only handle two uncertain parameters. This issue of the problem size increasing exponentially with the number of states is known as Bellman's curse of dimensionality [67], and this limits existing

implicit dual methods as they do not take a derivative-based approach in continuous state space.

The dual iLQG method presented in this thesis fills this gap by extending the derivative-based iLQG method to be implicitly dual. The ability of both the dual and adaptive iLQG methods presented to be robust to time-varying parameters, unmodelled dynamics, and handle multiplicative noise was explored. Neither wide-sense dual control nor MS-SP-NMPC can handle multiplicative noise, making dual iLQG ideally suited to the wide range of control problems that inherently involve multiplicative noise.

Dual iLQG is applicable to a variety of complex nonlinear control problems with many uncertain parameters, as shown in its application to the control of anaerobic digestion and COVID-19. Since MS-SP-NMPC is only able to handle systems with two uncertain parameters, simplified versions of these systems were used to compare dual and adaptive iLQG with MS-SP-NMPC, and dual iLQG outperformed MS-SP-NMPC in all but one case. When the systems with all of the uncertain parameters were used, dual iLQG consistently outperformed adaptive iLQG.

6.1 Key contributions

A high-speed adaptive iLQG algorithm was developed that estimates individual parameters of a system instead of the entire dynamic model of the system. The parameters were treated as constants in the iLQG algorithm through the creation of an augmented constants vector, but treated as states in the outer loop filter through the creation of an augmented state vector and augmented state covariance matrix. Key contributions with adaptive iLQG include:

- developing adaptive iLQG,
- exploring robustness to time-varying parameter dynamics and unmodelled dynamics,
- demonstrating temporal feasibility,
- and comparing its performance to dual MS-SP-NMPC.

A dual iLQG algorithm was developed that can actively learn a system's uncertain parameters through probing actions, but only does so when the resulting reduction in uncertainty is expected to lead to a decrease in the objective function. The parameters were treated as states in the iLQG algorithm and the outer loop filter through the creation of an augmented state vector, augmented state covariance matrix, and an augmented dynamic model. Key contributions with dual iLQG include:

- developing implicit dual iLQG,
- overcoming Bellman's curse of dimensionality that limits conventional implicit dual control algorithms,
- exploring robustness to time-varying parameter dynamics, unmodelled dynamics, and multiplicative noise,
- demonstrating temporal feasibility,
- and demonstrating its superior performance over dual MS-SP-NMPC.

Adaptive and dual iLQG were also applied to complex nonlinear systems:

- anaerobic digestion,
- and COVID-19.

This work extends iLQG, developed by Li and Todorov [37] to be an implicit dual method, creating a more general version of Bar-Shalom and Tse's Wide-Sense Dual Control [4, 5, 68]. Unlike dual MS-SP-NMPC [64], which is also an implicit dual control approach, dual iLQG overcomes Bellman's curse of dimensionality and can handle state and measurement multiplicative noise.

6.2 Potential applications

In the 2017 review paper "Systems and Control for the future of humanity, research agenda: Current and future roles, impact and grand challenges" by Lamnabhi-Lagarrigue *et al.* [34], three requirements are listed that "call for a paramount role for data-driven modeling, which must be integrated into virtually all future complex engineering systems." Dual iLQG addresses two of these three requirements, specifically the need for models to adapt to changing parameters as well as the need for approaches that enable active learning, which is described as "probing the system/environment to generate sensor information that is suitable for model adaptation". Considering dual iLQG's ability to handle nonlinear systems and efficiently handle large systems, dual iLQG is in a favourable position to meet these requirements and could solve current and future control problems.

Since uncertainty is common in real-world systems, dual iLQG is applicable to a wide spectrum of applications. Aside from controlling mechatronic systems, dual iLQG can be applied

to inform health care systems, government policy, and financial systems. The review paper by Lamnabhi-Lagarrigue *et al.* mentions several high-impact system and control applications for the future, and dual iLQG could be applied to many of them, including automotive control, spacecraft control, renewable energy and smart grid, assistive devices for people with disabilities, and advanced building control. In this thesis, dual iLQG was applied to both a government policy problem (COVID-19) and a renewable energy problem (anaerobic digestion).

Additionally, dual iLQG can handle multiplicative noise where other dual control algorithms such as MS-SP-NMPC and wide-sense dual cannot. This feature suggests that dual iLQG should be particularly well suited for applications where multiplicative noise is common, such as in the control of prostheses using myoelectric signals [57], robot motion planning when using sensors with distance-dependent errors [15], stochastic fluid dynamics [13], and power grids where renewable energy represents a significant portion of total power generation [20, 2]. Other applications that involve multiplicative noise include financial stochastic volatility models [49], systems involving wireless communication [69], and batch chemical process control [55].

6.3 Limitations

There are several limitations to this control approach:

- Dual iLQG requires a dynamic model for each system that it is applied to, and the noise characterizes of that system need to be known. In the absence of a white box dynamic model, Gaussian Process regression can be used to provide the system dynamics [29, 35].
- Dual iLQG is not guaranteed to outperform other methods in every application as the probing nature of dual control does not always result in lower long-term costs.

6.4 Future directions

There are several directions for future research on dual iLQG. These include:

- Investigating what theoretical guarantees can be placed on the stability, optimality, and robustness of dual and adaptive iLQG. There has been little research on the stability of iLQG, but methods used to analyze the stability of multi-stage NMPC may prove useful.
- Investigating what theoretical boundaries can be placed on the robustness of dual and adaptive iLQG.

- The implementation of dual iLQG on real systems.
- The ability of state constraints to be imposed on the system could be improved by combining this work with the augmented Lagrangian method for state constraints in differential dynamic programming [25].
- With the similarities between iLQG and differential dynamic programming, this work could easily be extended to take a second order approximation of the system dynamics instead of the present first order approximation, to create a dual differential dynamic programming. Theodorou, Tassa, and Todorov's work on stochastic differential dynamic programming [65] would help with this endeavour.
- The applicability of dual iLQG to chaotic systems could be investigated.

Dual iLQG shows significant promise as a control approach that can account for and actively reduce the uncertainties that are inherent to systems while improving their performance. Since this approach is so general and requires no training prior to implementation, it may be applicable to a wide range of fields including applications that have been identified as high-impact. Dual iLQG may have the potential to solve future control challenges that humanity will face.

Appendix A

Detailed comparison of DDP-like controllers

The following tables were created by Andrew Mathis, Jon Sensinger, Anjana Arunachalam, and Ali Estarami to compare papers on the iLQR, iLQG, and DDP algorithms.

Year	Authors	Title
1966	Mayne	A second-order gradient method of optimizing non-linear discrete time systems
1976	Bar-Shalom Tse	Caution, Probing, and the Value of Information in the Control of Uncertain Systems
2004	Li Todorov	Iterative linear quadratic regulator design for nonlinear biological movement systems
2005	Todorov	Stochastic optimal control and estimation methods adapted to noise characteristics of the sensori-motor system
2005	Todorov	Code: Stochastic optimal control and estimation
2005	Todorov Li	A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems
2005	Todorov Li	Code: A generalized iterative LQG method
2006	Li Todorov	An Iterative Optimal Control and Estimation Design for Nonlinear Stochastic System
2007	Li Todorov	Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system
2012	Tassa Todorov	Synthesis and stabilization of complex behaviors through online trajectory optimization
2014	Tassa Todorov	Control limited differential dynamic programming
2014	Tassa Todorov	Code: Control limited differential dynamic programming
2023	Mathis Carretero Sensinger	Dual iterative linear quadratic Gaussian control for uncertain nonlinear systems

Table A.1: Detailed comparison of DDP-like controllers: titles

Year	Authors	Algorithm name	What is it actually?	States Augmented?	Dual?	Dynamics
1966	Mayne	DDP	DDP	No	No	Nonlinear
1976	Bar-Shalom Tse	Wide-sense dual	Explicit dual fully observable DDP with partial CE assumption	Yes (in example)	Yes, explicit	Nonlinear
2004	Li Todorov	iLQR	iLQR	No	No	Nonlinear
2005	Todorov	gLQG	gLQG	No	No	Linear
2005	Todorov	gLQG	gLQG	No	No	Linear
2005	Todorov Li	iLQG	iLQR with multiplicative noise cost terms in control law	No	No	Nonlinear
2005	Todorov Li	iLQG	iLQR	No	No	Nonlinear
2006	Li Todorov	iLQG	iLQG	No	No	Nonlinear
2007	Li Todorov	iLQG	iLQG	No	No	Nonlinear
2012	Tassa Todorov	DDP	iLQR-based DDP	No	No	Nonlinear
2014	Tassa Todorov	DDP	iLQR-based DDP	No	No	Nonlinear
2014	Tassa Todorov	DDP	iLQR-based DDP	No	No	Nonlinear
2023	Mathis Carretero Sensinger	Dual iLQG	Implicit dual iLQG	Yes	Yes, implicit	Nonlinear

Table A.2: Detailed comparison of DDP-like controllers: algorithms

Year	Authors	States fully observable?	Stochastic?	Additive noise	Multiplicative noise	States filtered?	Control law has noise terms?
1966	Mayne	Yes	No	No	No	No	Yes
1976	Bar-Shalom Tse	No	Yes	Yes	No	Yes	Yes
2004	Li Todorov	Yes	No	No	No	No	No
2005	Todorov	No	Yes	Yes	Yes	Yes	Yes
2005	Todorov	No	Yes	Yes	Yes	Yes	Yes
2005	Todorov Li	Yes	Yes	No	Yes	No	Yes
2005	Todorov Li	Yes	No	No	No	No	No
2006	Li Todorov	No	Yes	No	Yes	Yes	Yes
2007	Li Todorov	No	Yes	No	Yes	Yes	Yes
2012	Tassa Todorov	Yes	No	No	No	No	No
2014	Tassa Todorov	Yes	No	No	No	No	No
2014	Tassa Todorov	Yes	No	No	No	No	No
2023	Mathis Carretero Sensinger	No	Yes	Yes	Yes	Yes	Yes

Table A.3: Detailed comparison of DDP-like controllers: noise

Year	Authors	Cost approximation	Dynamics approximation used in cost function	In linearization of dynamics, Does A=Ac or Ad?
1966	Mayne	2nd order	2nd order	
1976	Bar-Shalom Tse	2nd order	2nd order	
2004	Li Todorov	Quadratic cost, so no approximation needed.	1st order	A=Ac
2005	Todorov	Quadratic cost, so no approximation needed.	None	N.A.
2005	Todorov	Quadratic cost, so no approximation needed.	None	N.A.
2005	Todorov Li	2nd order	1st order	A=Ad (*e.g.*, I+Ac*dt)
2005	Todorov Li	2nd order	1st order	A=Ac
2006	Li Todorov	2nd order	1st order	A=Ad (*e.g.*, I+Ac*dt)
2007	Li Todorov	2nd order	1st order	A=Ac
2012	Tassa Todorov	2nd order	2nd order	A=Ac
2014	Tassa Todorov	2nd order	2nd order	A=Ac
2014	Tassa Todorov	2nd order	2nd order	A=Ac
2023	Mathis Carretero Sensinger	2nd order	1st order	A=Ac

Table A.4: Detailed comparison of DDP-like controllers: approximations

Year	Authors	Descent algorithm	Regularization
1966	Mayne	Newton (Hessian)	??
1976	Bar-Shalom Tse		
2004	Li Todorov	Gauss-Newton (Jacobian squared)	none
2005	Todorov	Gauss-Newton (Jacobian squared)	none
2005	Todorov	Gauss-Newton (Jacobian squared)	none
2005	Todorov Li	Gauss-Newton (Jacobian squared)	Levenberg-Marquardt and eigenvalue decomposition approach
2005	Todorov Li	Gauss-Newton (Jacobian squared)	Levenberg-Marquardt
2006	Li Todorov	Gauss-Newton (Jacobian squared)	Levenberg-Marquardt
2007	Li Todorov	Gauss-Newton (Jacobian squared)	Levenberg-Marquardt
2012	Tassa Todorov	Newton (Hessian)	L.M using iterative lambda; L.M. enforcing P.D.
2014	Tassa Todorov	Newton (Hessian)	Levenberg-Marquardt enforcing P.D., in conjunction with line-search
2014	Tassa Todorov	Newton (Hessian)	Levenberg-Marquardt enforcing P.D., in conjunction with line-search. 2 options in code
2023	Mathis Carretero Sensinger	Gauss-Newton (Jacobian squared)	Levenberg-Marquardt and 3 other options

Table A.5: Detailed comparison of DDP-like controllers: regularization

Year	Authors	Are updates applied to nonlinear system?	Control-constraint method	State-constraints
1966	Mayne	Yes	N.A.	none
1976	Bar-Shalom Tse	Yes	none	none
2004	Li Todorov	Yes	none	none
2005	Todorov	N.A.	none	none
2005	Todorov	N.A.	none	none
2005	Todorov Li	Yes	Line-search early on for efficiency; Clamp	none
2005	Todorov Li	Yes	Line-search early on; Clamp	none
2006	Li Todorov	Yes	None noted in equations, although they mention the technique can accommodate it	none
2007	Li Todorov	Yes	Line-search early on; Clamp.	none
2012	Tassa Todorov	Yes	none	Augmented Langrangian
2014	Tassa Todorov	Yes	Hamiltonian box (also covers squashing. A distant third option is clamping)	Clamp in forward pass
2014	Tassa Todorov	Yes	Hamiltonian box	none
2023	Mathis Carretero Sensinger	Yes	Line-search early on; Squashing	None

Table A.6: Detailed comparison of DDP-like controllers: constraints

Bibliography

[1] Yehia Abdelsalam, Sankaranarayanan Subramanian, and Sebastian Engell, *Asymptotically Stabilizing Multi-Stage Model Predictive Control*, 2020 59th IEEE Conference on Decision and Control (CDC) (Jeju, Korea (South)), vol. 2020-Decem, IEEE, dec 2020, pp. 710–717.

[2] Fateme Aghaee, Nima Mahdian Dehkordi, Navid Bayati, and Amin Hajizadeh, *Delay and General Multiplicative Noise-Resilient Secondary Frequency and Voltage Control for an Autonomous Microgrid*, 2021 12th Power Electronics, Drive Systems, and Technologies Conference (PEDSTC), IEEE, feb 2021, pp. 1–6.

[3] Joel A. E. Andersson, Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl, *CasADi: a software framework for nonlinear optimization and optimal control*, Mathematical Programming Computation **11** (2019), no. 1, 1–36.

[4] Yaakov Bar-Shalom and Edison Tee, *Dual Effect, Certainty Equivalence and Separation in Stochastic Control.*, IEEE Transactions on Automatic Control **AC-19** (1974), no. 5, 250–257.

[5] Yaakov Bar-Shalom and Edison Tse, *Caution, Probing, and the Value of Information in the Control of Uncertain Systems*, Annals of Economic and Social Measurement **5** (1976), no. 3, 323–337.

[6] D. J. Batstone, J. Keller, I. Angelidaki, S. V. Kalyuzhnyi, S. G. Pavlostathis, A. Rozzi, W. T. Sanders, H. Siegrist, and V. A. Vavilin, *The IWA Anaerobic Digestion Model No 1 (ADM1).*, Water science and technology : a journal of the International Association on Water Pollution Research **45** (2002), no. 10, 65–73.

[7] Vinay A. Bavdekar and Ali Mesbah, *Stochastic Model Predictive Control with Integrated Experiment Design for Nonlinear Systems*, IFAC-PapersOnLine **49** (2016), no. 7, 49–54.

[8] David S. Bayard and Alan Schumitzky, *Implicit dual control based on particle filtering and forward dynamic programming*, International Journal of Adaptive Control and Signal Processing **24** (2010), 155–177.

[9] Richard Bellman, *A Markovian Decision Process*, Indiana University Mathematics Journal **6** (1957), no. 4, 679–684.

[10] Olivier Bernard, Zakaria Hadj-Sadok, Denis Dochain, Antoine Genovesi, Jean-philippe Philippe Steyer, Comore Project, and Sophia-antipolis Cedex, *Dynamical model development and parameter identification for an anaerobic wastewater treatment process. Biotechnology and bioengineering 75.4 (2001) 424-438..pdf*, Biotechnology and Bioengineering **75** (2001), no. 4, 424–438.

[11] Vladimír Bobál, Petr Chalupa, Petr Dostál, and Petr Dostál Vladimír Bobál, Petr Chalupa, *Adaptive Dual Control of Nolinear Liquid System*, IFAC Proceedings Volumes **40** (2007), no. 13, 483–488.

[12] Eric Bradford and Lars Imsland, *Economic Stochastic Model Predictive Control Using the Unscented Kalman Filter*, IFAC-PapersOnLine **51** (2018), no. 18, 417–422.

[13] Jingyi Chao and Thomas Schäfer, *Multiplicative noise and the diffusion of conserved densities*, Journal of High Energy Physics **2021** (2021), no. 1, 71.

[14] Gregory Chow, *Analysis and control of dynamic economic systems*, Wiley, New York, 1975.

[15] Noel E. Du Toit and J. W. Burdick, *Robot Motion Planning in Dynamic, Uncertain Environments*, IEEE Transactions on Robotics **28** (2012), no. 1, 101–115.

[16] Lorenzo Fagiano, Georg Schildbach, Marko Tanaskovic, and Manfred Morari, *Scenario and Adaptive Model Predictive Control of Uncertain Systems*, IFAC-PapersOnLine **48** (2015), no. 23, 352–359.

[17] M. Farrokhsiar and H. Najjaran, *A robust probing motion planning scheme: A tube-based MPC approach*, 2013 American Control Conference (Washington, DC, USA), IEEE, jun 2013, pp. 6492–6498.

[18] Morteza Farrokhsiar, *MPC-based Mobile Robots Motion Planning and Control in Uncertain Dynamic Environment*, Doctoral, The University of British Columbia, 2015, p. 114.

[19] A Feldbaum, *Dual control theory. I*, Avtomat. i Telemekh. **21** (1960), no. 9, 1240–1249.

[20] Yi Guo and Tyler H. Summers, *A Performance and Stability Analysis of Low-inertia Power Grids with Stochastic System Inertia*, 2019 American Control Conference (ACC), IEEE, jul 2019, pp. 1965–1970.

[21] Kristian G. Hanssen and Bjarne Foss, *Scenario Based Implicit Dual Model Predictive Control*, IFAC-PapersOnLine **48** (2015), no. 23, 416–421.

[22] Tor Aksel N. Heirung, Bjarne Foss, and B. Erik Ydstie, *MPC-based dual control with online experiment design*, Journal of Process Control **32** (2015), 64–76.

[23] Tor Aksel N Heirung, Joel A Paulson, Jared O'Leary, Ali Mesbah, Jared O Leary, Ali Mesbah, Jared O'Leary, and Ali Mesbah, *Stochastic model predictive control - how does it work?*, Computers & Chemical Engineering **114** (2018), 158–170.

[24] Tor Aksel N Heirung, B. Erik Ydstie, and Bjarne Foss, *An Adaptive Model Predictive Dual Controller*, 11th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing (Caen, France), vol. 46, IFAC, 2013, pp. 62–67.

[25] Wilson Jallet, Antoine Bambade, Nicolas Mansard, and Justin Carpentier, *Constrained Differential Dynamic Programming: A primal-dual augmented Lagrangian approach*, 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (Kyoto, Japan), 2022.

[26] Julie Jimenez, Eric Latrille, Jérôme Harmand, Angel Robles, José Ferrer, Daniel Gaida, Christian Wolf, Francis Mairet, Olivier Bernard, Victor Alcaraz-Gonzalez, Hugo Mendez-Acosta, Daniel Zitomer, Dennis Totzke, Henri Spanjers, Fabian Jacobi, Alan Guwy, Richard Dinsdale, Giuliano Premier, Sofiane Mazhegrane, Gonzalo Ruiz-Filippi, Aurora Seco, Thierry Ribeiro, André Pauss, and Jean Philippe Steyer, *Instrumentation and control of anaerobic digestion processes: a review and some research challenges*, Reviews in Environmental Science and Biotechnology **14** (2015), no. 4, 615–648.

[27] Hoil Kil, Dewei Li, Yugeng Xi, and Jiwei Li, *Model predictive control with on-line model identification for anaerobic digestion processes*, Biochemical Engineering Journal **128** (2017), 63–75.

[28] Edgar D. Klenske, Philipp Hennig, Bernhard Scholkopf, and Melanie N. Zeilinger, *Approximate dual control maintaining the value of information with an application to building control*, 2016 European Control Conference (ECC) (Aalborg, Denmark), IEEE, jun 2016, pp. 800–806.

[29] Edgar Dietrich Klenske, *Nonparametric Disturbance Correction and Nonlinear Dual Control*, Ph.D. thesis, ETH Zürich, 2017, p. 251.

[30] Johannes Köhler, Lukas Schwenkel, Anne Koch, Julian Berberich, Patricia Pauli, and Frank Allgöwer, *Robust and optimal predictive control of the COVID-19 outbreak*, Annual Reviews in Control **51** (2021), 525–539.

[31] Kunal Kumar, Tor Aksel N Heirung, Sachin C. Patwardhan, and Bjarne Foss, *Experimental evaluation of a MIMO adaptive dual MPC*, IFAC-PapersOnLine **28** (2015), no. 8, 545–550.

[32] H. C. La, A. Potschka, J. P. Schlöder, and H. G. Bock, *Dual Control and Information Gain in Controlling Uncertain Processes*, IFAC-PapersOnLine **49** (2016), no. 7, 139–144.

[33] Huu Chuong La, *Dual Control for Nonlinear Model Predictive Control*, Ph.d., Heidelberg University, 2016, p. 140.

[34] Francoise Lamnabhi-Lagarrigue, Anuradha Annaswamy, Sebastian Engell, Alf Isaksson, Pramod Khargonekar, Richard M. Murray, Henk Nijmeijer, Tariq Samad, Dawn Tilbury, and Paul Van den Hof, *Systems & Control for the future of humanity, research agenda: Current and future roles, impact and grand challenges*, Annual Reviews in Control **43** (2017), 1–64.

[35] Gilwoo Lee, Siddhartha S. Srinivasa, and Matthew T. Mason, *GP-ILQG: Data-driven Robust Optimal Control for Uncertain Nonlinear Dynamical Systems*, arXiv (2017).

[36] Weiwei Li and Emanuel Todorov, *Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems*, Proceedings of the First International Conference on Informatics in Control, Automation and Robotics, SciTePress - Science and and Technology Publications, 2004, pp. 222–229.

[37] ———, *An iterative optimal control and estimation design for nonlinear stochastic system*, Proceedings of the IEEE Conference on Decision and Control (San Diego, CA, USA), 2006, pp. 3242–3247.

[38] ———, *Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system*, International Journal of Control **80** (2007), no. 9, 1439–1453.

[39] Sergio Lucia, *Robust Multi-stage Nonlinear Model Predictive Control*, Ph.d., Technical University of Dortmund, 2015.

[40] Sergio Lucia and Sebastian Engell, *Potential and limitations of multi-stage nonlinear model predictive control*, IFAC-PapersOnLine **28** (2015), no. 8, 1015–1020.

[41] Sergio Lucia, Sankaranarayanan Subramanian, Daniel Limon, and Sebastian Engell, *Stability properties of multi-stage nonlinear model predictive control*, Systems and Control Letters **143** (2020), 104743.

[42] Sergio Lucia, Alexandru Tatulea-Codrean, Christian Schoppmeyer, and Sebastian Engell, *An environment for the efficient testing and implementation of robust NMPC*, 2014 IEEE Conference on Control Applications (2014), no. October, 1843–1848.

[43] Harsh Maheshwari, Shreyas Shetty, Nayana Bannur, and Srujana Merugu, *CoSIR: Optimal control of SIR epidemic dynamics by mapping to Lotka-Volterra System*, medRxiv (2021).

[44] Giancarlo Marafioti, *Enhanced Model Predictive Control: Dual Control Approach and State Estimation Issues*, Doctoral, Norwegian University of Science and Technology, 2010, p. 150.

[45] David Mayne, *A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems*, International Journal of Control **3** (1966), no. 1, 85–95.

[46] Zawadi Mdoe, Dinesh Krishnamoorthy, and Johannes Jäschke, *Adaptive horizon economic nonlinear model predictive control*, Journal of Process Control **92** (2021), 108–118.

[47] Rudolph Van Der Merwe and Eric Wan, *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*, in Proceedings of the 60th Annual Meeting of The Institute of Navigation (ION) (2004).

[48] Ali Mesbah, *Stochastic model predictive control with active uncertainty learning: A Survey on dual control*, Annual Reviews in Control **45** (2018), 107–117.

[49] Thomas Mikosch and Mohsen Rezapour, *Stochastic volatility models with possible extremal clustering*, Bernoulli **19** (2013), no. 5A.

[50] Djordje Mitrovic, Stefan Klanke, and Sethu Vijayakumar, *Adaptive optimal feedback control with learned internal dynamics models*, Studies in Computational Intelligence **264** (2010), 65–84.

[51] Altan Onat, *A Novel and Computationally Efficient Joint Unscented Kalman Filtering Scheme for Parameter Estimation of a Class of Nonlinear Systems*, IEEE Access **7** (2019), 31634–31655.

[52] Radoslav Paulen, Mario E. Villanueva, and Benoît Chachuat, *Guaranteed parameter estimation of non-linear dynamic systems using high-order bounding techniques with domain and CPU-time reduction strategies*, IMA Journal of Mathematical Control and Information **33** (2016), no. 3, 563–587.

[53] Tamás Péni, Balázs Csutak, Gábor Szederkényi, and Gergely Röst, *Nonlinear model predictive control with logic constraints for COVID-19 management*, Nonlinear Dynamics **102** (2020), no. 4, 1965–1986.

[54] Jay Prakash Tripathi and Richa Tripathi, *Adaptive nonlinear control scheme for containment of COVID-19 spread*, Simulation **98** (2022), no. 9.

[55] S.Joe Qin and Thomas A. Badgwell, *A survey of industrial model predictive control technology*, Control Engineering Practice **11** (2003), no. 7, 733–764.

[56] Martin A. Sehr and Robert R. Bitmead, *Tractable dual optimal stochastic model predictive control: An example in healthcare*, 2017 IEEE Conference on Control Technology and Applications (CCTA) (Kohala Coast, Hawaii, USA), IEEE, aug 2017, pp. 1223–1228.

[57] Jonathon W. Sensinger and Strahinja Dosen, *A Review of Sensory Feedback in Upper-Limb Prostheses From the Perspective of Human Motor Control*, Frontiers in Neuroscience **14** (2020).

[58] Jean-Jacques Slotine and Weiping Li, *Applied Nonlinear Control*, Prentice Hall, 1991.

[59] Sankaranarayanan Subramanian, Sergio Lucia, and Sebastian Engell, *Economic multi-stage output feedback NMPC using the Unscented Kalman Filter*, IFAC-PapersOnLine **28** (2015), no. 8, 38–43.

[60] Yuval Tassa, Tom Erez, and Emanuel Todorov, *Synthesis and stabilization of complex behaviors through online trajectory optimization*, IEEE International Conference on Intelligent Robots and Systems (Vilamoura, Algarve, Portugal), 2012, pp. 4906–4913.

[61] Yuval Tassa, Nicolas Mansard, and Emo Todorov, *Control-limited differential dynamic programming*, Proceedings - IEEE International Conference on Robotics and Automation (Hong Kong, China), 2014, pp. 1168–1175.

[62] S. Thangavel, S. Lucia, R. Paulen, and S. Engell, *Dual robust nonlinear model predictive control: A multi-stage approach*, Journal of Process Control **72** (2018), 39–51.

[63] Sakthi Thangavel, Sergio Lucia, Radoslav Paulen, and Sebastian Engell, *Towards dual robust nonlinear model predictive control: A multi-stage approach*, 2015 American Control Conference (ACC) (Chicago, Illinois, USA), IEEE, jul 2015, pp. 428–433.

[64] Sakthi Thangavel, Radoslav Paulen, and Sebastian Engell, *Dual multi-stage NMPC using sigma point principles*, IFAC-PapersOnLine, vol. 53, Elsevier B.V., 2020, pp. 11243–11250.

[65] Evangelos Theodorou, Yuval Tassa, and Emo Todorov, *Stochastic differential dynamic programming*, Proceedings of the 2010 American Control Conference (Baltimore, Maryland, USA), 2010, pp. 1125–1132.

[66] Emanuel Todorov, *Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system*, Neural Computation **17** (2005), no. 5, 1084–1108.

[67] ———, *Optimal Control Theory*, Bayesian Brain, MIT Press Scholarship Online, 2006.

[68] Edison Tse, Yaakov Bar-Shalom, and Lewis Meier, *Wide-Sense Adaptive Dual Control for Nonlinear Stochastic Systems*, IEEE Transactions on Automatic Control **18** (1973), no. 2, 98–108.

[69] Vyacheslav Tuzlukov, *Generalized Approach to Signal Processing in Wireless Communications: The Main Aspects and some Examples*, Wireless Communications and Networks - Recent Advances, InTech, mar 2012.

[70] Andreas Wächter and Lorenz T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Mathematical Programming **106** (2006), no. 1, 25–57.

[71] Avishai Weiss and Stefano Di Cairano, *Robust dual control MPC with guaranteed constraint satisfaction*, Proceedings of the IEEE Conference on Decision and Control (2014), 6713–6718.

Vita

Candidate's full name: Andrew Craig Mathis
University attended:

- 2022, Diploma in University Teaching, University of New Brunswick (UNB)
- 2016 - 2018, M.Sc.E., Mechanical Engineering, UNB
- 2014 - 2016, Diploma in Technology Management and Entrepreneurship, UNB
- 2011 - 2016, B.Sc.E., Mechanical Engineering, UNB

Publications:

- Andrew Craig Mathis, *A high packing ratio linear actuator for use in cable driven parallel manipulators*, Master's thesis, 2018.
- Andrew Mathis and Jon Sensinger, *Implicit Dual Control for Uncertain Stochastic Systems*, US63/537,243 filed September 8, 2023. Provisional patent.
- Andrew Mathis, Juan Antonio Carretero, and Jon Sensinger, *Beyond adaptive control: A control method for nonlinear systems with uncertainties, applied to COVID-19*, Submitted to Nature Scientific Reports (2023).

Conference Presentations:

- Andrew Craig Mathis, Juan Antonio Carretero, and Yves Losier, *Towards a High Packing Ratio Linear Actuator for use in Cable Driven Parallel Manipulators*, CCToMM (2019).
- Andrew Mathis, Juan Antonio Carretero, and Jon Sensinger, *Actively adaptive control applied to COVID-19*, MEGSC (2022).
- Andrew Mathis, Juan Antonio Carretero, and Jon Sensinger, *A learning algorithm applied to the control of COVID-19*, NBHRF Health Research Week (2022).

Patent:

Andrew Mathis and Jon Sensinger, *Implicit Dual Control for Uncertain Stochastic Systems*, US63/537,243 filed September 8, 2023. Provisional patent.