NTNU

Norwegian University of
Science and Technology

Solving Adaptive Optimal Control Problems with Dynamic Programming

Hilde Fuglstad

Master of Science in Cybernetics and Robotics

Submission date:	June 2017
Supervisor:	Bjarne Anton Foss, ITK
Co-supervisor:	Tor Aksel N. Heirung, University of California, Berkeley

Norwegian University of Science and Technology
Department of Engineering Cybernetics

NTNU
Norwegian University of Science and Technology

Faculty of Information Technology,
Mathematics and Electrical
Engineering
Department of Engineering Cybernetics

Master project

Name of candidate: Hilde Fuglstad

Subject: Engineering Cybernetics

Title: Solving adaptive optimal control problems with dynamic programming

Title (in Norwegian): Løsning av adaptive optimale reguleringsproblemer med dynamisk programmering

Background

Adaptive optimal control problems (AOCP) is an interesting class of controllers since they balance the use of control and active excitation as defined through the principle of dual control (DC). Dynamic programming (DP) provides a general approach for solving AOCP, and the purpose of this project is to explore the use of DP for solving AOCP by evaluating alternative formulations, noise model assumptions and parallelization. The numerical experiments should assess the capabilities of typical hardware configurations, like a standard laptop, in the context of AOCP and DP through numerical experiments on selected examples.

Task description

1. Provide an overview of AOCP and the use of DP for solving such problems. Furthermore, an overview of approximate techniques for achieving active excitation in controllers should be included.
2. Define an appropriate AOCP formulation and an associated DP solution approach. Explain also how an estimator is embedded in the solution approach.
3. Define goals for the numerical experiments and select a few systems (2-3) with appropriate uncertainty descriptions.
4. Implement a framework for solving DP on the systems defined in item 3.
5. Perform a comprehensive set of numerical experiments on the selected systems to explore the effect of
    a. reformulations as for instance is possible in the minimum-time case.
    b. different noise descriptions and in particular robustness towards errors in noise models.
    c. parallelization options.
    d. discretization accuracy.
6. Assess the applicability of DP given current typical hardware configurations and suggest approaches for embedding active excitation into controllers.

The candidate may extract key findings and summarize these in a scientific paper format as part of the master thesis report.

Starting date: 16.01.2017

End date: 12.06.2017

Co-supervisor: Tor Aksel N. Heirung, University of California, Berkeley

Trondheim, 16.01.2017
Bjarne Foss
Professor/supervisor

Abstract

In optimal control of uncertain systems, lack of crucial information about the system can lead to unacceptable performance like the violation of constraints. In these, or similar situations where it is important to reduce uncertainty quickly, excitation can be used for learning purposes. The optimal balance between learning and control is achieved with dual control. This concept was introduced over seventy years ago and is still relevant. It has been shown that dynamic programming (DP) can be used to solve these problems, along with a number of approximate methods. Analytical solution of the problems are in most cases impossible and it is therefore necessary to solve them numerically.

The purpose of this thesis is to provide an overview of adaptive optimal control problems (AOCP) and the use of DP for solving them. The method is explored through several illustrating examples and the dual control algorithm is evaluated through computer simulations. The main examples considered are a simple integrator problem with unknown gains, and a minimum-time problem with an unknown breaking coefficient. The unknown parameters and noise in the systems are modelled as stochastic variables with known statistical distributions that are utilized by the dual controller.

It is shown how the different AOCP can be formulated, and the DP algorithms can be implemented. Different noise model assumptions are evaluated to see how this can affect the problem. Numerical experiments assess the capabilities of typical hardware configurations and parallelization options explore the possibility of reduced runtime. Results from simulations certainly demonstrate how the dual controller manage to both control the process and learn about it simultaneously. The controller is also compared to a certainty equivalent (CE) and cautious controller to further emphasize the advantages it has to these heuristic, adaptive controllers. Despite the well-known problems related to the curse of dimensionality, it is shown that it is possible to solve the given AOCP using DP with a desired accuracy, within reasonable time.

Sammendrag

I optimal regulering av usikre systemer kan mangel på essensiell informasjon om systemet lede til uakseptabel oppførsel, som brudd på begrensninger. I slike situasjoner hvor det er nødvendig å redusere usikkerheten umiddelbart, kan det være fordelaktig å bruke eksitasjon for å lære mer om systemet før det er for sent. Den optimale balansen mellom læring og regulering oppnås med dual regulering. Dette konseptet ble introdusert for over sytti år siden, men er fortsett like aktuelt i dag. Det har blitt vist at dynamisk programmering kan brukes for å løse denne typen problemer, ved siden av en mengde metoder som baserer seg på approksimasjoner og tilnærminger. Analytisk løsning av problemene er i de fleste tilfeller umulig og det er derfor nødvendig å løse de numerisk.

Formålet med denne oppgaven er å gi en oversikt over adaptive, optimale reguleringsproblemer og bruken av dynamisk programmering for å løse de. Metoden blir utforsket gjennom noen illustrerende eksempler og den duale regulatoren blir evaluert gjennom datasimuleringer. Eksemplene som blir benyttet er et enkelt integrator problem hvor avstanden fra en gitt referanseverdi skal minimeres og et minimum-tid problem. Begge problemene kan ha ukjente parametere og støy som kan modelleres som stokastiske variabler. Kjennskap til deres sannsynlighetsfordelinger kan dermed brukes av den duale regulatoren til å forbedre reguleringen.

Det blir vist hvordan de adaptive, optimale reguleringsproblemene kan formuleres og hvordan algoritmene for dynamisk programmering kan implementeres. Ulike antakelser omkring sannsynlighetsfordelingene blir evaluert for å se hvordan det kan påvirke problemet. De numeriske eksperimentene viser hvilke muligheter vi har for å løse denne typen problemer med dagens datamaskiner, og parallellisering blir utforsket for å se på muligheten for å redusere kjøretiden. Resultater fra simuleringene demonstrerer tydelig hvordan regulatoren evner å både regulere prosessene og å lære om de. Den duale regulatoren blir også sammenliknet med en CE regulator og en forsiktig regulator for å framheve dens fordeler relativt til disse heuristiske, adaptive regulatorene. Til tross for velkjente skaleringsproblemer relatert til dynamisk programmering, vises det her at det for enkle systemer er mulig å oppnå resultater med relativt god nøyaktighet innen rimelig tid.

Preface

The work documented in this thesis is carried out at the Department of Engineering Cybernetics, at the Norwegian University of Science and Technology, the spring of 2017. It is part of the fulfillment for the degree Master of Science in Cybernetics and Robotics and is done with great help from Professor Bjarne A. Foss and Postdoctoral Tor Aksel N. Heirung. I am grateful for the interest you have taken in my work and for all the instructive conversations we have had. Thank you for prioritising my work in your busy schedules. I would also like to thank my family for always being there for me. I am grateful for the support we have in each other. Lastly, I would like to thank my boyfriend Håkon for all his help and support.

Trondheim, June 2017

Hilde Fuglstad

Table of Contents

Abstract i

Summary iii

Preface v

Table of Contents viii

List of Tables ix

List of Figures xii

Abbreviations xiii

1 Introduction 1

1.1 Structure of the Thesis 4

2 Dynamic Programming 5

2.1 The Dynamic Programming Algorithm 6

2.2 Interpolation 7

2.3 Approximate Dynamic Programming 8

3 The Adaptive Optimal Control Problem 9

3.1 Problem Formulation 9

3.2 Parameter Estimation 11

3.2.1 Parameter Estimation with Gaussian Random Variables . . . 11

3.2.2 Parameter Estimation with Uniform Random Variables . . . 12

3.3 Solving the AOCP with Dynamic Programming 14

3.3.1 Calculation of Expected Cost to go 17

3.4 Approximate Techniques to Dual Control 19

4 Dual Control of an Integrator with Unknown Gain 21
4.1 One Unknown Parameter . . . 21
4.1.1 AOCP Formulation . . . 21
4.1.2 Solving the AOCP with DP . . . 23
4.2 Two Unknown Parameters . . . 25
4.2.1 AOCP Formulation . . . 25
4.2.2 Solving the AOCP with DP . . . 26

5 Time-Optimal Dual Control of a Cart System with an Unknown Breaking Coefficient 27
5.1 Cart Model . . . 28
5.2 The Optimal Control Problem . . . 28
5.3 AOCP Formulation . . . 29
5.4 Solving the AOCP with DP . . . 31

6 Testing Environment 33
6.1 Implementation of the DP Algorithms . . . 33
6.2 Integrator Specifications . . . 34
6.3 Cart System Specifications . . . 36
6.4 Parallelization . . . 38
6.4.1 Parallel for-Loops . . . 38
6.5 Experiments . . . 39
6.5.1 Overview . . . 39
6.5.2 Integrator with One Unknown Parameter . . . 40
6.5.3 Integrator with Two Unknown Parameters . . . 40
6.5.4 Cart System without Noise . . . 41
6.5.5 Cart System with Noise . . . 41

7 Results and Discussion 43
7.1 Integrator with One Unknown Parameter . . . 43
7.1.1 The Dual Control Law . . . 44
7.1.2 Simulation . . . 49
7.2 Integrator with Two Unknown Parameters . . . 51
7.3 Runtime for the DP Algorithms for the Integrator System . . . 52
7.4 Cart System without Noise . . . 54
7.5 Cart System with Noise . . . 58
7.6 Runtime for the DP Algorithm for the Cart System . . . 60
7.7 Common Observations . . . 60

8 Conclusion and Future Work 63

A Statistics 65
A.1 Uniform Distribution . . . 65
A.2 Normal Distribution . . . 66

Bibliography 66

List of Tables

6.1 Computer specifications 33

7.1 Integrator: Comparison of dual, CE and cautious controller. Mean cost over 10 000 simulations for $\hat{\theta}(0) = 2$. 49

7.2 Integrator: Comparison of dual, CE and cautious controller. Mean cost over 10 000 simulations for $P(0) = 1$. 51

7.3 Integrator: Runtime of DP algorithm for original variables with one and two unknown parameters, respectively. Grids used are $16\times16\times9$ and $16 \times 16 \times 16 \times 9 \times 9 \times 5$. $\Delta u = 0.2$ and $|u_{max}| = 5$. 52

7.4 Integrator: Runtime of DP Algorithms for different grid sizes. The result is for a system with one unknown parameter. $\Delta \nu = \Delta u = 0.01$, $|u_{max}| = 10$ and 5 points is used to represent P for the original variables. 54

7.5 Minimum-time: Deviation from desired end state for different values of t_f and $\hat{\theta}(0)$, without noise. The result shown is for $P(0) = 1$ and $\theta = 1$ 55

7.6 Minimum-time: Mean deviation from desired end state over 5000 simulations, for different values of t_f and probability distributions for v. $\hat{\theta}(0) = 0.25$, $P(0) = 1$ and $\theta = 1$. 59

7.7 Minimum-time: Runtime of DP Algorithm. $\Delta u = 0.1$, $|u_{max}| = 1$ and grid is given by $x_1 \times x_2 \times \hat{\theta} \times P$. 60

List of Figures

1.1 The optimal control family tree from Diehl and ESAT-SCD (2011). . 2

2.1 Illustrating the *principle of optimality* through optimal path calculation 5
2.2 Illustration of cubic vs. linear interpolation 7

3.1 Block diagram of an adaptive optimal regulator from Chapter 7 in Åström and Wittenmark (2013), Figure 7.1 10
3.2 Example of parameter estimation with uniform noise: non-linear algorithm vs. Kalman filter in one simulation 15
3.3 Mean estimation error and covariance: non-linear algorithm vs. Kalman filter for uniform noise 15
3.4 Simulation of two uniformly distributed variables x_1 and x_2 and their sum. 19

5.1 Illustration of the Cart problem: Cart moves from position x_0 to x_f 27
5.2 Illustration of non-dual control for cart system with unknown breaking coefficient. 30

7.1 Integrator: Dual control laws $\nu(\eta = 0, \beta)$ for different time horizons T. 44
7.2 Integrator: Surface plot of dual control laws $\nu(\eta = 0, \beta)$ for different time horizons T. 45
7.3 Integrator: Dual control laws for $T = 31$ (upper) and $T = 3$ (lower) found with normalized variables. 46
7.4 Integrator: Dual control laws for $T = 3$ with $P = 1$ (upper) and $P = 0.25$ (lower) found with original variables. 47
7.5 Integrator sim 1: System with one unknown parameter simulated with the dual control law. 50

7.6 Integrator sim 2: System with two unknown parameters simulated with the dual control law. (Dotted lines are true parameter values.) 53

7.7 Cart sim 1: System without noise simulated with dual control law. Initial estimate $\hat{\theta}(0) = 2$ and variance $P(0) = 1$. 56

7.8 Cart sim 2: System without noise simulated with dual control law. Initial estimate $\hat{\theta}(0) = 0.25$ and variance $P(0) = 1$. 57

7.9 Cart: Dual control law $u^*(y_1 = 0, y_2 = 0.1, \hat{\theta}, P = 1)$ at $t = 0.1$ for $t_f = 2.1$ 58

Abbreviations

ADP	=	Approximate Dynamic Programming
AOCP	=	Adaptive Optimal Control Problem
CE	=	Certainty Equivalent
DP	=	Dynamic Programming
HJB	=	Hamilton-Jacobi-Bellman
MPC	=	Model Predictive Control
MRAC	=	Model Reference Adaptive Controller
OCP	=	Optimal Control Problem
STC	=	Self-Tuning Controller

Chapter 1

Introduction

Adaptive optimal control problems involve optimal control of an uncertain system. In real-life control problems we rarely posses complete knowledge of the systems. More often we have to deal with uncertainties related to the system models, or unknown influence from the environment. In these situations we have to make choices without knowing exactly how these decisions will affect our system. Since the early 1950s, there have been a lot of research on adaptive control. Åström and Wittenmark (2013) loosely defines an adaptive controller as *a controller that has adjustable controller parameters and a method for adjusting them.* Some of the earliest and most well-known heuristic, adaptive controllers are the model reference adaptive controller (MRAC) and the self-tuning controller (STC). These controllers use parameter estimates as if they are the true ones and do not take action to influence the uncertainty related to the estimates. Therefore, we can say that the controllers are based on the *certainty equivalence principle.* A different adaptive controller that do take parameter uncertainties into account is the *cautious controller.* It does however not try to reduce this uncertainty. All of these controllers therefore have some limitations, and are in general not optimal.

As an attempt to design adaptive control laws to optimal control problems, a different approach was later proposed. Through the use of nonlinear stochastic control theory, an interesting class of controllers was developed, that balanced the use of control and excitation, defined through the principle of dual control. The main difference between these controllers and the ordinary adaptive controllers is the modelling of the unknown parameters as stochastic variables. This makes it possible for the controller to exploit the knowledge of the uncertainty related to the system.

The term *dual* refers to the dual goals of the controller. These are to control the process in an optimal way at the same time as it should work actively to learn about the system to reduce the uncertainty. An optimal decision will consequently

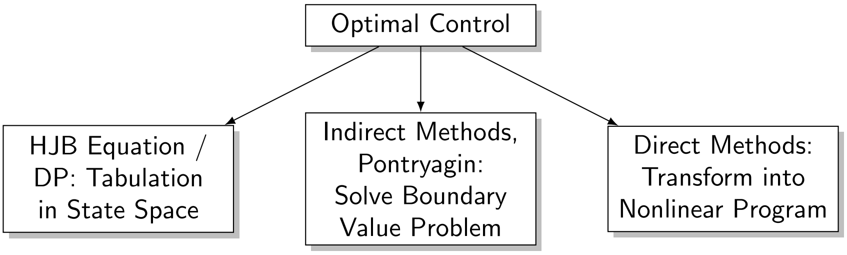

Optimal Control
HJB Equation / DP: Tabulation in State Space
Indirect Methods, Pontryagin: Solve Boundary Value Problem
Direct Methods: Transform into Nonlinear Program

Figure 1.1: The optimal control family tree from Diehl and ESAT-SCD (2011).

rely upon both current data and expected future data resulting from the current decision. As Heirung et al. (2017) argue, these goals are not conflicting, but rather complementary when the uncertainty is relevant for the decision making. This does not however imply that the optimal course of action is to always use the control for learning. In some situations, where for instance the uncertainty does not affect the decision (or it is very low), experimentation will not be optimal. This means that a dual control problem needs to possess the ability to use the control input for the purpose of learning, but that it is not necessarily utilized. Bar-Shalom and Tse (1974) describe this *dual effect* as a property that allows the control to affect both the state itself and the uncertainty related to it. If the control cannot affect the uncertainty, the system is *neutral*.

The approach with dual control was first developed by Feldbaum (1960), but it is not the only method used for uncertainty reduction. It can also be achieved with non-dual methods. Two standard approaches are passive learning and dedicated experiments. The first corresponds to learning as a side effect of normal operating conditions as is the case with the ordinary CE adaptive controllers. For the second approach, it is achieved through customized experiments that are carried out as a separate process. However, as Heirung (2016) defines in his doctoral thesis, the dual control signal is the *optimal* control if the system possesses the dual effect and the uncertainty is decision relevant.

A general optimal control problem concerns identification of the optimal decisions to be made over a given control horizon. What characterizes an optimal decision is that it minimizes (maximizes) a specified objective while possibly satisfying some given constraints. This objective will depend on the purpose of the control. It can for instance be to keep the output of a given system as close as possible to a reference value, or to reach a given end state in minimum time. Typically there will be some natural constraints on the available input and the states of the system.

There are different approaches for solving optimal control problems. Diehl and ESAT-SCD (2011) divide them into the three main classes seen in Figure 1.1. The first is the Hamilton-Jacobi-Bellman (HJB) equation or dynamic programming approach. This is based on the principle of optimality, which says that any subarc of an optimal trajectory is also optimal, as defined by Bellman (1954). The second approach is indirect methods. This involves deriving a boundary value problem and then solving this numerically. For this reason it is often referred to as "first optimize, then discretize". The well known calculus of variations and the Pontryagin Maximum Principle belong to this class. Lastly, we have direct methods. They are based on transforming an infinite optimal control problem into a finite dimensional nonlinear programming problem that is solved. In this thesis, we will mainly be concerned with dynamic programming, since this is shown by Feldbaum (1960) to give a conceptually simple, optimal solution to the AOCP.

Despite the method of dual control being known since the 1960s, and a great deal of research being conducted, there are still a lot to be learned. Due to the known difficulties related to the optimal solution obtained with dynamic programming, there have not been a huge development on the area. At the moment there seem to be more resources used on the field of approximate methods, either in the form of approximations of the optimal solution or by reformulation of the problem (Lee and Lee (2009); Heirung et al. (2015)). This will be briefly presented later in the thesis.

Åström and Helmersson (1986) did however solve a dual control problem with one unknown parameter with DP. This thesis will revisit the problem and recreate their solution. Moreover, it will look at a different formulation of the problem and the extension to two unknown parameters. A minimum-time dual control problem with a double integrator with one unknown parameter will also be investigated. Through these examples we want to evaluate the DP approach to dual control to further understand the principles, and learn about both the strengths and limitations. The problems are implemented and solved numerically, and the behaviour of the dual controllers are evaluated primarily through computer simulations. For comparison, simple CE and cautious controllers will also be implemented and compared to the dual controllers.

Multiple experiments will be conducted to asses the capabilities of typical hardware configurations in the context of AOCP and DP for the examples, and to see what can be achieved with today's equipment. A method for parallelization will also be implemented to analyse the possibility of reducing the computation time for the algorithms. The impact of different noise descriptions on the minimum-time problem is also studied. We want to see past the Gaussian assumption on the system, and investigate the effect of applying noise from a uniform probability distribution. Moreover, we want to see if it is possible to use customized algorithms for uniformly distributed uncertainty. Based on the note by Servi and Ho (1981), a nonlinear recursive parameter estimation algorithm for a one dimensional system with uniformly distributed noise and parameter is derived. This is further explored in relation to the dual controller.

1.1 Structure of the Thesis

The rest of the thesis is organized as follows. First we will present some background that is necessary for solving the adaptive optimal control problems. Chapter 2 gives a short presentation of dynamic programming. Here the principle of optimality is used to derive the Bellman equation and interpolation is introduced. Chapter 3 shows how a general AOCP is formulated mathematically and how it can be solved with DP. Different methods for estimating the unknown parameters are derived and quadrature formulas for calculating the approximated cost is given. Then the two applications will be presented. The first, described in Chapter 4, is a continuation of the integrator problem in Åström and Helmersson (1986), while the second presented in Chapter 5 is a minimum-time optimal control problem with an unknown breaking coefficient. Chapter 6 presents the testing environment for these examples, while the results and a discussion of the findings are given in Chapter 7. Finally, Chapter 8 concludes the thesis and gives some suggestions for future work.

Chapter 2

Dynamic Programming

In this chapter the principles of dynamic programming will be presented. We will look at the Bellman equation built from the principle of optimality and introduce interpolation as a useful aid. At last, a short presentation of approximate methods is given. All this is necessary to understand how dynamic programming can be used to solve the adaptive optimal control problems we will encounter later. The derivation in this chapter is partly based on what was written in my project report.

Similar to what was done in this project, I will in my thesis work with discrete systems where decisions are made in stages. Dynamic programming is a method that can be used for these kinds of optimization problems. The continuous counterpart is the Hamilton-Jacobi-Bellman equation. Dynamic programming is based on dividing a complex system into simpler subproblems which are easier to solve. The concept is simple, but it rarely produces nice analytical solutions. Therefore, the problems are rather solved numerically and results from calculations at each stage of the algorithm are stored in a table to avoid recalculation. Often we deal with continuous systems that are approximated in a discrete space. Due to this approximation, the method will produce a globally optimal solution only on the discrete approximation of the space. If there are multiple solutions giving the same

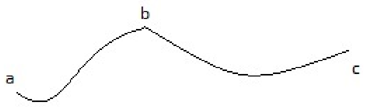


Figure 2.1: Illustrating the *principle of optimality* through optimal path calculation

optimal value, it will return one of these.

The method is based on the *principle of optimality* as Bellman defined in the following way (Bellman, 1954)

Definition 1. *An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.*

As illustrated in Figure 2.1, this means that if you have an optimal path from a to c and this goes through b, then you also know that there is no better way to get from b to c. If that had been the case, there would also have been a better way to get from a to c. A simple proof by contradiction can be found in Kirk (1967).

When this principle holds, it is not necessary to search over all possible solutions. The clear advantage is seen in the comparison between dynamic programming and direct enumeration in Kirk (2012), Section 3.9. Here, it is investigated what will happen when the number of stages N in the process is increased. The result is that the growth in direct enumeration is exponential, while it is linear for dynamic programming. Unfortunately, DP also has a serious drawback known as the *curse of dimensionality*, that prevents us from using it in larger problems. This is based on the fact that the problem size grows exponentially with the number of states x and control inputs u. If we have m control inputs and n states that are described by d_m and d_n discrete point respectively, we get $d_m^m \times d_n^n$ possibilities at each stage. Hence, the grids used to represent the states and control inputs greatly affect the run time. How many discrete points we use is therefore a trade-off between how accurate the solution will be, and how much time it takes to solve the problem and how much memory we need.

2.1 The Dynamic Programming Algorithm

Solving an optimal control problem with dynamic programming involves finding the optimal control at each stage to minimize the total cost over all stages. This is done by considering both the immediate cost and the future cost resulting from the decision. By using the principle of optimality stated earlier, the DP-algorithm can be defined. Bertsekas (2005) gives a good description of the algorithm. It is built on the idea that we can start at the last stage and move backward in time. At each stage calculating the optimal control by minimizing a specified cost function where the results from the previous stage is used during each calculation. For a deterministic system, the future cost resulting from a decision can be found by a simple table lookup and possibly interpolation. When there are uncertainties however, we cannot know precisely what will be the next state of the system. Therefore, calculating the cost to go for a stochastic system is more complicated.

There are especially two methods used for optimization with uncertainties. The first is based on finding the expected value, while the second method considers

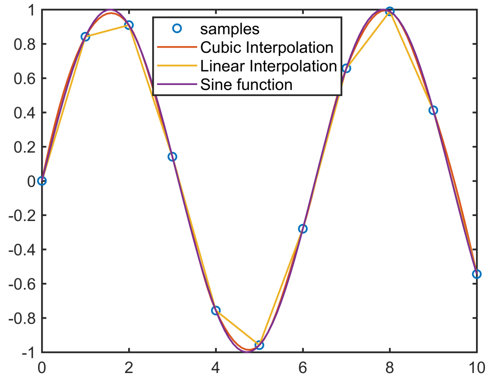


Figure 2.2: Illustration of cubic vs. linear interpolation

worst-case situations (Shapiro et al., 2013). Here we will use the first method and formulate the DP-algorithm as an expected value. The result of the optimal control problem will be the control sequence u^* that is most likely to give the minimum cost. A typical optimal control objective is mathematically stated as

$$V(x(t),t) = \min_{u(k),\ k=t,\ldots,N-1} E[\sum_{k=t}^{N-1} F(x(k+1),u(k))] \tag{2.1}$$

with value function V equal to the total cost over the given time horizon, state x, control input u and stage cost F. The relationship between $x(t+1)$, $x(t)$ and $u(t)$ is described through a known dynamic system equation and the cost at the last stage $V(x(N),N)$ is also assumed known.

By applying the principle of optimality to the sum in this value function, we can find that

$$V(x(t),t) = \min_{u(k),\ k=t,\ldots,N-1} E[F(x(t+1),u(t)) + V(x(t+1),t+1)] \tag{2.2}$$

This is known as the Bellman equation. To find the optimal control to be applied at each stage from $x(0)$, you have to start the algorithm at $t = N-1$ and calculate $V(x(N-1,N-1))$, and go all the way back to $V(x(0),0)$. At each stage checking all possible inputs at all possible states to get a complete list containing optimal inputs to apply at a certain time, depending on the state you are in.

2.2 Interpolation

When dealing with continuous systems, the states and control inputs have to be sampled at distinct values before DP can be used. In the simplest examples, the values can be carefully selected to avoid ending up at a value for $x(k+1)$ in between the sampled values. For other problems it may not be a good idea to sample that often considering shortage in time or memory. More distinct values means that there will be more calculations and values to be stored and this may not be desirable. On the other hand it may happen that one is to calculate the cost of applying a control input and the next state is not one of the states considered one step earlier. It is in these situations that interpolation can be used.

There are different interpolation methods. When choosing the method to use, one have to decide if it is important to save memory and computation time or to get a smooth result. Two common methods are linear and cubic interpolation. Out of these two, linear uses the least memory and has the fastest computation time and cubic gives a smoother curve. An example with the two methods can be seen in Figure 2.2. Here we have sampled a sine function at 11 different time instances and tried to recreate it by using interpolation. For this function it is clear that cubic interpolation gives a result that is very similar to the original function. However, in certain situations it may be superfluous.

When implementing the DP algorithm in MATLAB, the function *griddedInterpolant* (MathWorks, a) can be used. It takes the grid and corresponding sample values as inputs and optionally the interpolation (and extrapolation) method. With linear interpolation, the function value is found as

$$f(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \tag{2.3}$$

where the goal is to find the function value at a value of x that is not a point in the grid, given the values $f(x_0)$ and $f(x_1)$ at the closest lower x_0 and upper x_1.

2.3 Approximate Dynamic Programming

The application of approximate dynamic programming (ADP) may often be necessary in practice. Approximate schemes are available to overcome challenges related to the curse of dimensionality, or unavailable or difficult models that makes it hard to compute expectation values. The importance is especially evident in situations where you are dealing with time sensitive on-line calculations. There are several approaches to ADP. One important idea is that a system can learn about its own behaviour through simulation forward in time. The classical approach is based on replacing the value function with an approximation. The approximation can be improved through parameter adjustment based on results from the simulation. The approximate function is not easy to derive and should depend on the problem at hand. Another possibility is to adjust the parameters of the input policy directly. Typically, the ADP methods save computation time and memory by exploring only a subset of the space. See Bertsekas (2011) or Powell (2011) for an orientation on some different methods.

Chapter 3

The Adaptive Optimal Control Problem

In this chapter the AOCP will be formulated mathematically. Methods for estimating the parameters in linear systems influenced by noise that is both normally and uniformly distributed will be presented. Further, it is shown how the problem can be solved numerically with DP. In the end, different approximated techniques to dual control will be discussed. Parts of this chapter is a continuation of what was written in my project.

3.1 Problem Formulation

In this thesis, I look at adaptive optimal control problems for systems with uncertainty related to the model parameters and not the model structure. This parameter uncertainty will be described through statistical distributions and it is assumed that it can be reduced by manipulation of the system input. The optimal control problem (OCP) in discrete time can generally be stated mathematically as

$$\min_{u(k),\ k=t,\ldots,N-1} E[\sum_{k=t}^{N-1} F(x(k+1), u(k)) | \gamma(t)] \tag{3.1a}$$

$$x(t+1) = f(x(t), u(t), \theta(t), v(t)) \tag{3.1b}$$

$$\theta(t+1) = g(\theta(t), e(t)) \tag{3.1c}$$

with x, u and θ being the states, control inputs and parameters, respectively. e and v are independent random variables with known probability distributions. F is the stage cost and the total cost is calculated as an expectation $E[\cdot]$ with respect

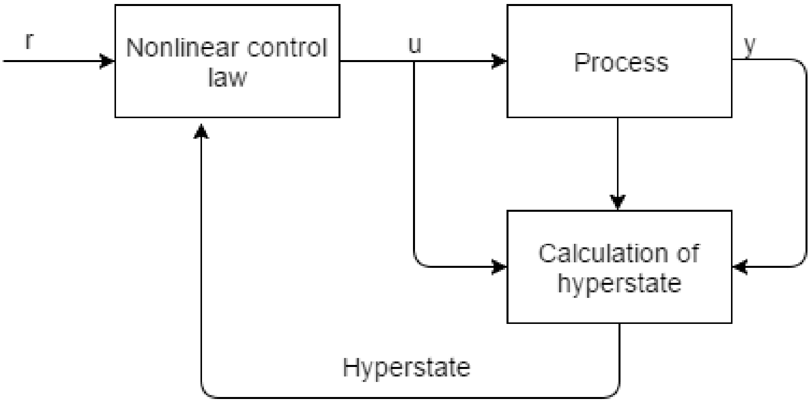


Figure 3.1: Block diagram of an adaptive optimal regulator from Chapter 7 in Åström and Wittenmark (2013), Figure 7.1

to the random variables, conditioned on recorded data γ. Initial conditions are assumed to be known along with bounds on variables. The control horizon N can either be a known constant, or a decision variable. The problem consists of finding the optimal control sequence u^* that minimizes the objective (3.1a) for the system described by (3.1b) and (3.1c).

The control problem is said to be *adaptive*, since there also will be a way to estimate and update the parameters θ. Methods for parameter estimation will be shown in Section 3.2. The parameters will behave as states in the resulting system and it is therefore nonlinear. In this thesis we will assume that the system (3.1b) is linear in the parameters and can be expressed as

$$x(t+1) = \phi^T(t)\theta(t) + v(t) \tag{3.2}$$

where ϕ is an m-dimensional vector with known functions of the measured input u and output x and parameters

$$\theta(t) = [\theta_1(t), \dots, \theta_m(t)]^T \tag{3.3}$$

We will also assume that the parameters are constant and write

$$\theta(t+1) = \theta(t) = \theta \tag{3.4}$$

Since it rarely is possible to model their variation in practice, this can be a fair assumption in many situations.

To simplify the structure of the dual control problem, we introduce a *hyperstate* $\xi(t)$, similar to what Åström and Wittenmark (2013) have done. This hyperstate is fed back to the controller. It can contain the states, control inputs and statistical moments describing the system at time t. The resulting structure can be seen in Figure 3.1. This is similar to the ordinary STC, except for the statistical moments. They add more complexity to the problem.

3.2 Parameter Estimation

When the parameters are unknown, we need a way to estimate them. For our system (3.2), we want to find a recursive estimate of the unknown parameter θ. There are multiple methods available, and the most appropriate method will depend on the system and the specific uncertainty assumptions. It is unquestionably most common to assume a Gaussian distribution for the uncertain parameters and noise. The reason for this is that many real life systems fits this distribution and also that it can simplify calculations considerably in certain situations, as we will see shortly. In this section we will look at parameter estimation for both the Gaussian and uniformly distributed variables. The different uncertainty descriptions lead to different methods to estimate the parameters, and also calculation of the expected cost to go in the resulting Bellman equation. Later we want to investigate the effect of different uncertainty descriptions on a dual control problem and in particular robustness towards possible errors in the noise assumptions.

3.2.1 Parameter Estimation with Gaussian Random Variables

When the uncertainty in the system is Gaussian, we can use the well-known recursive least squares method (Ioannou and Sun, 2012). The idea behind the method is to fit a mathematical model to observed data by minimizing the sum of squares between the observed and computed data. When the sequence of random variables, $v(t)$ in (3.2) is Gaussian with zero mean and variance R, the estimate can be described by the following system of equations

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)(x(t) - \phi^T(t)\hat{\theta}(t-1)) \tag{3.5a}$$

$$K(t) = P(t-1)\phi(t)(R + \phi^T(t)P(t-1)\phi(t))^{-1} \tag{3.5b}$$

$$P(t) = (I - K(t)\phi^T(t))P(t-1) \tag{3.5c}$$

where $\hat{\theta}(t)$ is the current parameter estimate. This can be interpreted as a Kalman filter for the process described by (3.2) and (3.4) and it minimizes the variance for the given system (Brown and Hwang, 2012).

When we assume that the parameters θ can be described by a Gaussian conditional distribution given $\gamma(t)$, we find (Åström and Wittenmark, 2013)

$$\begin{aligned} E[\theta|\gamma(t)] &= \hat{\theta}(t) \\ E[(\theta - \hat{\theta}(t))(\theta - \hat{\theta}(t)^T)|\gamma(t)] &= P(t) \end{aligned} \tag{3.6}$$

where $\hat{\theta}(t)$ is the conditional mean and $P(t)$ the conditional covariance at time t. Furthermore, we can derive that the conditional distribution of $x(t+1)$ from (3.2), given $\gamma(t)$ is Gaussian with conditional mean and covariance given by

$$\begin{aligned} \mu_x &= E[x(t+1)|\gamma(t)] = \phi^T(t)\hat{\theta}(t) \\ \sigma_x^2 &= E[(x(t+1) - \hat{x}(t))^2|\gamma(t)] = R + \phi^T(t)P(t)\phi(t) \end{aligned} \tag{3.7}$$

This is a neat property that can simplify the problem significantly.

3.2.2 Parameter Estimation with Uniform Random Variables

It is less common to assume a uniform probability distribution for the parameters. When the parameters and noise are uniformly distributed, the state $x(t+1)$ does not have the same conditional distribution as θ and v, as was the case with the Gaussian distribution. This makes the problem a bit more intricate, but we will look at a possible method for parameter estimation for the simple problem we are studying in this thesis. Refer Appendix A to view the basis for the statistical calculations in this section.

Recursive estimation of a parameter with a uniform probability distributions is not as straightforward as with the Gaussian distribution described earlier. However, when we only have noise in the state and the parameter is constant, we know that the posterior distribution of the parameter conditioned on the observations is also uniformly distributed (Servi and Ho, 1981). The main difference in this case is that the update equations will be non-linear. The Kalman filter that is optimal for the Gaussian system is also the best minimum variance *linear* estimate for the system corrupted by uniformly distributed noise.

For our system made up of (3.2) and (3.4), we assume a uniform a priori probability distribution for θ

$$f_\theta(\theta) = \begin{cases} \frac{1}{b_\theta - a_\theta} & a_\theta \leq \theta \leq b_\theta, \\ 0 & \text{else} \end{cases} \tag{3.8}$$

and the noise $v(t)$ is uniformly distributed according to

$$f_v(v(t)) = \begin{cases} \frac{1}{2r_v} & -r_v \leq v(t) \leq r_v, \\ 0 & \text{else} \end{cases} \tag{3.9}$$

From these known probability distributions, the uniform a posterior probability distribution of θ conditioned on the observations, $f_\theta(\theta|\gamma(t))$, can be found. A proof on the uniformity assumption of this distribution can be found in Servi and Ho (1981). With this knowledge of the posterior distribution of the parameter, update equations for an estimate of $\hat{\theta}$ can be found.

The posterior distribution of the parameter will have conditional mean and covariance

$$\begin{aligned} E[\theta|\gamma(t)] &= \hat{\theta}(t) \\ E[(\theta - \hat{\theta}(t))(\theta - \hat{\theta}(t)^T)|\gamma(t)] &= P(t) \end{aligned} \tag{3.10}$$

where the conditional expectation will represent the current estimate of the parameter. An algorithm that updates the current estimate at each time instant can be found. For our simple system with a constant parameter, we have a disturbance v

that is uniformly distributed in $[-r_v, r_v]$ and an a priori knowledge that θ will be in the interval $[a_\theta, b_\theta]$. Equivalently we can have a priori information of the mean $\hat{\theta}(0)$ and covariance $P(0)$ of θ. This means that $\theta \in [\hat{\theta}(0) - \sqrt{3P(0)}, \hat{\theta}(0) + \sqrt{3P(0)}]$ is all the information we have about the parameter initially. When we receive a new measurement of $x(t)$ at time t, we can use it to improve the estimate of $\hat{\theta}$. This can be done by noting that θ also has to lie in the set $[\phi^{-1}(t)(x(t) - r_v), \phi^{-1}(t)(x(t) + r_v)]$. Using the fact that the centre of the intersection of the two sets equals the conditional expectation of θ, we can find the minimum error variance estimate.

Let $\Theta(t)$ denote the set containing $\theta(t)$ consistent with the observations of x up until time t and the a priori knowledge

$$\Theta(t) := [a_\theta(t), b_\theta(t)] = [\hat{\theta}(t) - \sqrt{3P(t)}, \hat{\theta}(t) + \sqrt{3P(t)}] \tag{3.11}$$

where the last part comes from the knowledge of a uniform a posteriori conditional distribution of θ. Let $X(t)$ denote the set containing $\theta(t)$ consistent only with the latest observation $x(t)$

$$X(t) := [\phi^{-1}(t)(x(t) - r_v), \phi^{-1}(t)(x(t) + r_v)] \tag{3.12}$$

Then we can find an update of $\Theta(t)$ from the intersection of the previous set $\Theta(t-1)$ and the set $X(t)$ obtained from the measurement of x at time t.

$$\begin{aligned} \Theta(t) &= \Theta(t-1) \cap X(t) \\ &= [\hat{\theta}(t-1) - \sqrt{3P(t-1)}, \hat{\theta}(t-1) + \sqrt{3P(t-1)}] \cap \\ & [\phi^{-1}(t)(x(t) - r_v), \phi^{-1}(t)(x(t) + r_v)] \end{aligned} \tag{3.13}$$

Further, the intersection of the two sets can be found from

$$\begin{aligned} \Theta(t) &= [\max(\hat{\theta}(t-1) - \sqrt{3P(t-1)}, \\ & \phi^{-1}(t)(x(t) - r_v)), \min(\hat{\theta}(t-1) + \sqrt{3P(t-1)}, \phi^{-1}(t)(x(t) + r_v))] \\ &= [a_\theta(t), b_\theta(t)] \end{aligned} \tag{3.14}$$

which results in the following parameter estimate

$$\hat{\theta}(t) = \frac{b_\theta(t) + a_\theta(t)}{2} \tag{3.15}$$

and variance

$$P(t) = \frac{(b_\theta(t) - a_\theta(t))^2}{12} \tag{3.16}$$

This means that the interval θ can be in, is narrowed down as new measurements become available. If we assume for our simple system in (3.2) and (3.4), that $\theta = 1$, $\phi(t) = 1$, $v(t) \in [-1, 1]$ and a priori information about θ is $\Theta(0) = [0, 4]$. Now, if you for instance get a measurement $x(1) = 0$, then you know that θ will be in the set $X(1) = [-1, 1]$. Since you also know that θ has to be larger than zero from $\Theta(0)$, you can find the new interval as $\Theta(1) = [0, 1]$. If you next receive a measurement $x(2) = 2$, you can find that $X(2) = [1, 3]$. This results in $\Theta(2) = [1, 1]$ and therefore nothing more can be learned about the parameter. If you had observed $x(2) = 1$ on the other hand, you would not have been able to improve Θ.

Example: Comparison of Non-linear Algorithm and Kalman Filter for Uniformly Distributed Variables

As mentioned earlier, the Kalman filter is the best minimum variance linear estimate independent of the noise distribution. It is however not guaranteed to minimize the variance if the noise is not Gaussian. The non-linear algorithm outlined above on the other hand, will minimize the variance for the simple system given here. Figure 3.2 shows the result from one simulation of the two different methods for the system

$$\begin{aligned} \theta(t+1) &= \theta(t) = \theta \\ x(t+1) &= \phi(t)\theta + v(t) \end{aligned} \tag{3.17}$$

with random noise v uniformly distributed according to (3.9) and a priori distribution of θ from (3.8). The parameter estimation for the non-linear algorithm is found from (3.15) and (3.16), while the parameter estimation with the Kalman filter is found from (3.5), where R is the variance of v. In the simulation, we have $\theta = 1$, $\phi(t) = 1$, $a_\theta(0) = 0$, $b_\theta(0) = 4$, $\hat{\theta}(0) = 2$, $P(0) = \frac{4}{3}$, $r_v = 1$ and $R = \frac{1}{3}$. It can be seen from the figure that the non-linear method appears to be superior to the linear in this simulation. Furthermore, it can be seen from Figure 3.3 that the method produces a mean result that has a lower variance than the Kalman filter. Here the algorithms are simulated thousand times and the figure shows the calculated mean at each time t of the estimation error squared and covariance for the two methods. The blue and black lines represent the non-linear algorithm, while the red and magenta lines show the results for the Kalman filter. The figure also shows that the difference between the two algorithms increases along with the strength of the noise signal. From these simulations, it is clear that the Kalman filter is a good approximation when the system is under the influence of uniformly distributed noise, but also that it is possible to achieve even better results.

3.3 Solving the AOCP with Dynamic Programming

As described earlier, the solution to the AOCP made up of (3.1) and the parameter estimation for θ, can be found through the concept of dual control. Feldbaum (1960) showed that the theoretical solution to this dual control problem could be derived with dynamic programming as defined by Bellman. In this section, we will show how this can be done. Since the resulting expression rarely has an analytical solution, it will be necessary to discretize the value function in the variables of the hyperspace and use a quadrature formula to evaluate the integral.

To derive the Bellman equation for the AOCP, we define the value function from

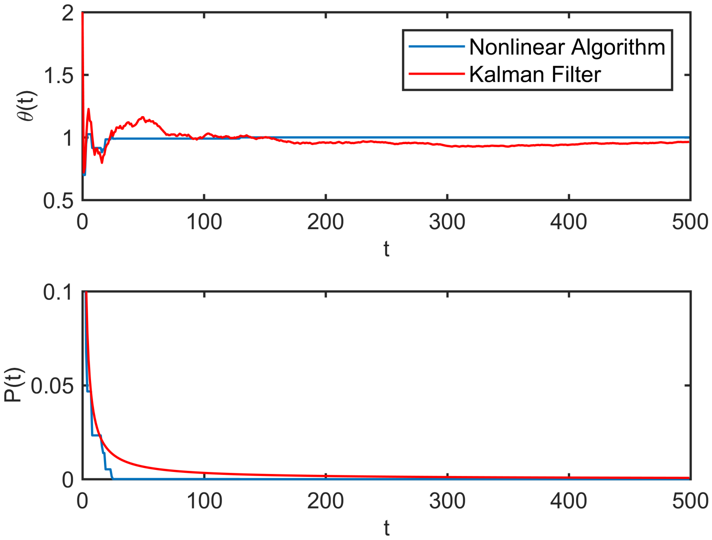


Figure 3.2: Example of parameter estimation with uniform noise: non-linear algorithm vs. Kalman filter in one simulation

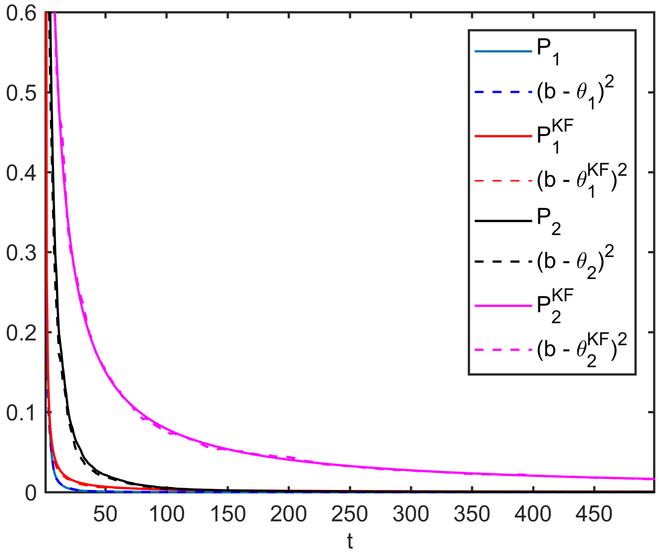


Figure 3.3: Mean estimation error and covariance: non-linear algorithm vs. Kalman filter for uniform noise

the objective function (3.1a) to be

$$V(\xi(t),t) = \min_{u(k),\ k=t,\ldots,N-1} E[\sum_{k=t}^{N-1} F(x(k+1),u(k))|\gamma(t)] \tag{3.18}$$

This can be interpreted as the minimum expected loss from the current time t to the end of the control horizon (N), given measurements up until time t. These measurements are expressed by γ. The value function depends on the hyperstate and current time. If we reformulate the expression on the right-hand side to obtain

$$V(\xi(t),t) = \min_{u(k),\ k=t,\ldots,N-1} E[F(x(t+1),u(t)) + \sum_{k=t+1}^{N-1} F(x(k+1),u(k))|\gamma(t)], \tag{3.19}$$

we can use the principle of optimality to derive the Bellman equation

$$V(\xi(t),t) = \min_{u(k),\ k=t,\ldots,N-1} E[F(x(t+1),u(t)) + V(\xi(t+1),t+1)|\gamma(t)] \tag{3.20}$$

Here, the optimal solution gives the optimal control inputs u^* that minimize the expected loss over the control horizon. One of the greatest difficulties with this expression is the unknown next hyperstate $\xi(t+1)$. The first expression on the right-hand side will depend on the objective of the minimization, while the second part requires some more consideration.

The recursive Bellman equation (3.20) is very computationally expensive to resolve. Analytical evaluation of the expression is in most cases impossible. The numerical solution will typically involve approximating the continuous space by a discrete space and iterating through all possible values in this grid. The algorithm will start at the last stage N and then move backward until reaching the initial stage. At each stage identifying the optimal control inputs at each possible combination of state values, that minimizes the expected loss. We have implicit knowledge of future information from calculations done in previous steps of the recursion. This future information is however not exact, since the discrete approximation only considers distinct values in the grid. This implies that the accuracy of the solution is limited by the discretization, but the distance between the points in the grid can not be too small, since this will lead to a large amount of calculations and memory needed.

Due to great limitations in computational power, the method was less useful in practice when it was first introduced by Feldbaum. Some decades after the dual control problem was derived however, it was revisited again by Åström and Helmersson (1986). At this time the authors decided to give computation of dual control laws a new chance with the current available computer power. They successfully solved a simple integrator problem with unknown gain numerically with dynamic programming. We will look more carefully at this problem later in this thesis.

3.3.1 Calculation of Expected Cost to go

In order to calculate the expected cost to go in (3.20), we have to use the probability function for the random variable and this will lead to an integral that needs to be approximated. This can be done with a number of different quadrature formulas, as for instance the well known Simpson's rule. In this subsection we will show some other methods that can simplify the approximations for the problems with the two specific uncertainty descriptions (Zarowski, 2004).

Gaussian Random Variables

When we are working with a Gaussian distribution, we can find the expression for the conditional expectation of the cost to go from

$$E[V(\xi(t+1), t+1)|\gamma(t)] = \int_{-\infty}^{\infty} V(\xi(t+1), t+1)\frac{1}{\sqrt{2\sigma_x^2\pi}}e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}}dx \tag{3.21}$$

With mean μ_x and variance σ_x^2 from (3.7). The resulting integral must be approximated for numerical solution. This can be done with a quadrature formula, like the Gauss-Hermite quadrature. This method requires that we first do a change of variables to get it on standard form. We set $y = \frac{x-\mu_x}{\sqrt{2}\sigma_x}$ and find

$$E[V(\xi(t+1), t+1)|\gamma(t)] = \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}}e^{-y^2}V(\xi(t+1), t+1)dy \tag{3.22}$$

with $x = y\sqrt{2}\sigma_x + \mu_x$. This can be approximated as

$$E[V(\xi(t+1), t+1)|\gamma(t)] \approx \frac{1}{\sqrt{\pi}}\sum_{i=1}^{n} w(i)V(\xi(t+1), t+1) \tag{3.23}$$

with

$$w(i) = \frac{2^{n-1}n!\sqrt{\pi}}{n^2[H_{n-1}(y(i))]^2} \tag{3.24}$$

and $y(i)$ are the roots of the Hermite polynomial

$$H_n(y(i)) = (-1)^n e^{y^2}\frac{\partial^n}{\partial y^n}e^{-y^2} \tag{3.25}$$

Uniform Random Variables

In the case with a Gaussian noise assumption, it was shown that a Gauss-Hermite quadrature could be used to obtain points and weights. When the system (3.2) is assumed to have both parameter θ and noise v with a uniform distribution, then the state $x(t+1)$ does not have a simple distribution with an obvious way to choose points and weights. If however, either the parameters are known or there are no noise in the system, then the state will have a known uniform distribution. In this case it will be sufficient to use a Gauss-Legendre quadrature to approximate the integral, since there is no exponential term, only a constant probability density function.

If we assume that the random variable X is in fact uniformly distributed, then the integral (see Appendix A)

$$E[V(\xi(t+1),t+1)|\gamma(t)] = \int_{\mu_x-\sigma_x\sqrt{3}}^{\mu_x+\sigma_x\sqrt{3}} V(\xi(t+1),t+1)\frac{1}{2\sqrt{3\sigma_x^2}}dx \tag{3.26}$$

can be approximated with a Gauss-Legendre quadrature after a change of variables, since this method is based on an interval $[-1,1]$. We set $x = \sigma_x\sqrt{3}y + \mu_x$ and find that $dx = \sigma_x\sqrt{3}dy$. If this is put into (3.26), we get

$$E[V(\xi(t+1),t+1)|\gamma(t)] = \frac{1}{2}\int_{-1}^{1} V(\xi(t+1),t+1)dy \tag{3.27}$$

The approximation is further found as

$$E[V(\xi(t+1),t+1)|\gamma(t)] \approx \frac{1}{2}\sum_{i=1}^{n} w(i)V(\xi(t+1),t+1) \tag{3.28}$$

with

$$w(i) = \frac{2}{(1-x^2(i))(\frac{\partial}{\partial x}P_n(x(i)))^2} \tag{3.29}$$

and $y(i)$ are the roots of the Legendre polynomial

$$P_n(x) = \frac{1}{2^n n!}\frac{\partial^n}{\partial y^n}(y^2-1)^n \tag{3.30}$$

If both the parameter is unknown and the system is corrupted by noise, the expectation will be over a random variable with a non-uniform distribution. In Figure 3.4 the result from a simulation in MATLAB of two uniformly distributed random variables $X_1 \sim U(a_{x_1}, b_{x_1}) = U(1,6)$ and $X_2 \sim U(a_{x_2}, b_{x_2}) = U(2,4)$ and their sum is shown. From this it can be seen that the distribution of the sum of two uniformly distributed variables has the shape of a trapeze. The most common approach to this problem is to treat the random variable X as if it had a Gaussian distribution. Then the calculation can be done according to the procedure outlined above. An alternative to this could be to manually choose points and weights from the distribution of X.

The probability density function for a function $X = X_1 + X_2$, that is the sum of two independent uniform random variables can be found as

$$f_x(x) = \int_{-\infty}^{\infty} f_{x_1}(x_1)f_{x_2}(x_2)dx_1 = \int_{-\infty}^{\infty} f_{x_1}(x_1)f_{x_2}(x-x_1)dx_1 \tag{3.31}$$

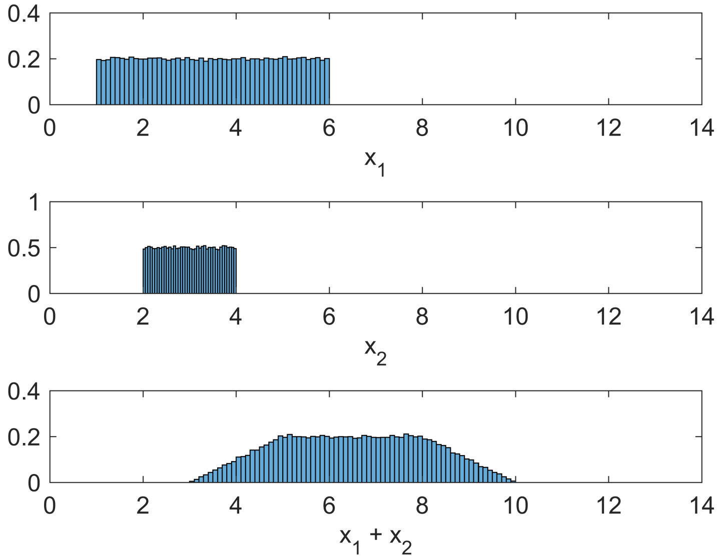


Figure 3.4: Simulation of two uniformly distributed variables x_1 and x_2 and their sum.

and by inserting the probability density functions for f_{x_1} and f_{x_2} we find

$$f_x(x) = \int_{a_x}^{b_x} \frac{1}{(b_{x_1} - a_{x_1})(b_{x_2} - a_{x_2})} dx_1 \tag{3.32}$$

where

$$a_x = \max(a_{x_1}, x - b_{x_2}) \tag{3.33}$$

and

$$b_x = \min(b_{x_1}, x - a_{x_2}) \tag{3.34}$$

Since the Gauss-Hermite approximation seems to produce an accurate result in calculations however, we do not proceed to produce any similar points and weights.

3.4 Approximate Techniques to Dual Control

Despite the rapid increase in computational capacity, it is still challenging in general to obtain the optimal solution with dynamic programming because of the *curse of dimensionality* and difficult calculations of probability distributions. To obtain the true dual solution with dynamic programming, we would need an infinite amount of memory and computational power. This has inspired the development of approximate methods.

Wittenmark (2002) proposes different approaches to derive what he refers to as suboptimal dual controllers. These avoid the issues related to the solution with dynamic programming, while dual features still remain. He suggests that they can be constructed either by approximations of the optimal dual control problem or by reformulating the problem. This can be achieved for example through approximations of the loss function or by adding a perturbation signal to a cautious controller to increase learning. A different approach considers modification of the loss function. Here the idea is to add terms to the loss function that reflect the quality of the parameter estimates. This can be done through the use of the covariance matrix for instance, which can be seen as a measure of the parameter uncertainty.

Over the years there have been some attempts to produce controllers with performance close to the optimal dual controller at a lower cost by the use of ADP. Thompson and Cluett (2005) show how the dual control problem can be solved without complete discretization of the hyperstate space or nested numerical integration. They do this by using a combination of iterative dynamic programming and Monte Carlo methods. Lee and Lee (2009) present another method that solves a stochastic optimal control problem with dynamic programming, but only approximately and within limited regions of the hyperstate space. Both approaches illustrate the benefits of the approximated algorithms on the same simple integrator example.

Another area with current development is dual control with model predictive control (MPC). Heirung et al. (2015) work on finding the exact solution to approximations of the dual control problem instead of finding approximations to the exact problem as for instance is done with the ADP methods. They look at an MPC-based approach to dual control with on-line experimental design. They present two approaches on how to excite the plant to generate informative data. One where the controller actively tries to reduce the error covariances, and the other where it maximizes the information in the signals. The algorithms presented both reduce parameter uncertainty and increase the information content of the closed-loop signals. Very recently it has also been proposed a different approach that avoids heuristic additions to the cost function (Heirung et al., 2017).

Chapter 4

Dual Control of an Integrator with Unknown Gain

In this chapter we will present a problem similar to the one studied in Åström and Helmersson (1986). This is a simple, yet nontrivial adaptive control problem, consisting of a simple integrator with an unknown and constant parameter. Despite the problem being known, it is interesting to see what is possible with today's computational resources. We will also extend the problem to include two unknown and constant parameters. The AOCP for the problem will be formulated and the solution with DP presented. In Chapter 6 it will be shown how the systems are implemented and tested.

4.1 One Unknown Parameter

In this section the problem with one unknown parameter will be presented. Two different formulations and their solution form will be given.

4.1.1 AOCP Formulation

The discrete-time system is described by

$$x(t+1) = x(t) + bu(t) + v(t) \tag{4.1}$$

where x, u and v are output, control input and random noise with a known Gaussian probability distribution with zero mean and variance R, respectively. b is a constant, unknown parameter with a known a priori probability distribution. The purpose of the control is to minimize the expected cost given by

$$\min_{u(k),\ k=t,\ldots,N-1} E[\sum_{k=t}^{N-1} x^2(k+1)|\gamma(t)] \tag{4.2}$$

where γ denotes the observed outputs and inputs available at time t and N is a known time horizon. There are no constraints specified on x and u. The goal is to find the optimal control sequence u^* that minimizes this cost function. This corresponds to keeping the state at the reference value $x(t) = 0$. It can easily be extended to include a desired non-zero reference $r(t)$ by writing $(x(k+1) - r(k))^2$ in the sum instead.

To get the system on familiar form (3.2), we set $\theta = b$ and $\phi(t) = u(t)$, to get

$$z(t) = x(t+1) - x(t) = \phi(t)\theta + v(t) \tag{4.3}$$

If we combine (4.1) and (4.2), we get the following OCP

$$\boxed{\begin{aligned} &\min_{u(k),\ k=t,\ldots,N-1} E[\sum_{k=t}^{N-1} x^2(k+1)|\gamma(t)] \\ &x(t+1) = x(t) + \phi(t)\theta + v(t) \\ &\theta(t+1) = \theta(t) = b \end{aligned}} \tag{4.4}$$

The hyperstate can be set to

$$\xi(t) = [x(t), \hat{\theta}(t), P(t)]^T \tag{4.5}$$

where $\hat{\theta}(t)$ is the current estimate of θ or the conditional mean, and $P(t)$ is the conditional covariance. (3.5) is used to find the recursive parameter estimate.

Normalized Variables

The problem can be simplified by using normalized variables and recognizing that the value function will not depend explicitly on the covariance P. This is done by using the following new variables similar to Åström and Helmersson (1986).

$$\eta(t) = \frac{x(t)}{\sqrt{R}} \tag{4.6}$$

$$\beta(t) = \frac{\hat{\theta}(t)}{\sqrt{P(t)}} \tag{4.7}$$

$$\zeta(t) = \frac{1}{\sqrt{P(t)}} \tag{4.8}$$

$$\nu(t) = \frac{u(t)\sqrt{P(t)}}{\sqrt{R}} \tag{4.9}$$

$$\epsilon(t) = \frac{z(t) - \phi(t)\hat{\theta}(t)}{\sqrt{R + \phi^2(t)P(t)}} \tag{4.10}$$

where η, ζ and β are state variables, ν is the control and ϵ is the normalized innovation. ϵ will be a random variable with mean value equal the mean value for the noise v, and variance one. Note that ζ is used instead of ξ to avoid confusion with the hyperstate.

The problem can then conveniently be written as

$$\boxed{\begin{aligned} &\min_{u(k),\ k=t,\dots,N-1} E[\sum_{k=t}^{N-1} \eta^2(k+1)|\gamma(t)] \\ &\eta(t+1) = \eta(t) + \beta(t)\nu(t) + \sqrt{1+\nu^2(t)}\epsilon(t) \\ &\beta(t+1) = \beta(t)\sqrt{1+\nu^2(t)} + \nu(t)\epsilon(t) \\ &\zeta(t+1) = \zeta(t)\sqrt{1+\nu^2(t)} \end{aligned}} \tag{4.11}$$

with the hyperstate set to

$$\xi_n(t) = [\eta(t), \beta(t)]^T \tag{4.12}$$

Since the cost function does not depend explicitly on ζ, this variable is not included. Åström and Helmersson (1986) show this by using induction.

The equations for $\eta(t+1)$, $\beta(t+1)$ and $\zeta(t+1)$ are similar to equations (2.5), (2.6) and (2.7) in Åström and Helmersson (1986). Note that there most likely is a typo in their equation, where $\beta(t+1)$ is replaced by $\chi(t+1)$. Similarly there appear to be some error in their Bellman equation for the original variables in the arguments of W_{T-1} in the integral. First, for $y(t+1)$, where I would suggest to write $y + \hat{b}u + \sqrt{\sigma^2 + u^2P}\epsilon$ instead. Second for $\hat{b}(t+1)$, where I believe it should be $\hat{b} + \frac{uP}{\sqrt{\sigma^2+u^2P}}\epsilon$.

4.1.2 Solving the AOCP with DP

In Section 3.3 it was shown how the solution to the AOCP can be found with DP. Here we will show how this can be done for the given integrator problem. First it can be noticed that if the parameter b in (4.1) is known and different from zero, the problem has a simple minimum variance optimal solution given by

$$u^*(t) = -\frac{x(t)}{b} \tag{4.13}$$

This means that the optimal cost when the parameter is known is just equal to the sum of the noise squared.

$$V(\xi(t),t) = \min_{u(k),\ k=t,\ldots,N-1} E[\sum_{k=t}^{N-1} x^2(k+1)|\gamma(t)] = \sum_{k=t}^{N-1} v^2(k) \tag{4.14}$$

If $\hat{\theta}$ is different from zero, the parameter in (4.13) can be changed with its estimate to produce a simple certainty equivalence controller.

$$u^*_{ce}(t) = -\frac{x(t)}{\hat{\theta}} \tag{4.15}$$

For the dual controller however, the optimal control can be identified from the solution of the Bellman equation (3.20). Here the stage cost F equals the output squared as seen in (4.2). This gives the following Bellman equation for this specific problem

$$V(\xi(t),t) = \min_{u(k),\ k=t,\ldots,N-1} E[x^2(t+1) + V(\xi(t+1),t+1)|\gamma(t)] \tag{4.16}$$

The first expression on the right side can be found from

$$\begin{aligned} E[x^2(t+1)] &= E[(x(t)+\phi(t)\theta+v(t))^2] \\ &= (x(t)+\phi(t)\hat{\theta}(t))^2 + R + \phi^2(t)P(t) \end{aligned} \tag{4.17}$$

where R is the variance of v and v is independent of ϕ, x and θ. The last expression can be found from

$$E[V(\xi(t+1),t+1)|\gamma(t)] = \int_{-\infty}^{\infty} V(\xi(t+1),t+1) f_z(z) dz \tag{4.18}$$

where $f_z(z)$ is the probability density function for the random variable Z in (4.3). The calculation of this integral can be done as shown in Subsection 3.3.1. This will lead to an expression on the form

$$E[V(\xi(t+1),t+1)|\gamma(t)] = \sum_{i=1}^{n} w(i) V(\xi(t+1),t+1)] \tag{4.19}$$

Normalized Variables

For the normalized variables, the Bellman equation will be

$$V(\xi_n(t),t) = \min_{\nu(k),\ k=t,\ldots,N-1} E[\eta^2(t+1) + V(\xi_n(t+1),t+1)|\gamma(t)] \tag{4.20}$$

with

$$\begin{aligned} E[\eta^2(t+1)] &= E[(\eta(t)+\beta(t)\nu(t)+\sqrt{1+\nu^2(t)}\epsilon(t))^2] \\ &= (\eta(t)+\beta(t)\nu(t))^2 + 1 + \nu^2(t) \end{aligned} \tag{4.21}$$

The last expression on the right side is calculated similar to the case with the original variables. To avoid the need for P, the expectation is taken over the random variable ϵ. This leads to the following expression

$$E[V(\xi_n(t+1), t+1)|\gamma(t)] = \int_{-\infty}^{\infty} V(\xi_n(t+1), t+1) f_\epsilon(\epsilon) d\epsilon \tag{4.22}$$

where ϵ is the random variable from (4.10).

4.2 Two Unknown Parameters

In this section we will give the problem with two unknown parameters a short presentation.

4.2.1 AOCP Formulation

The extension to two unknown parameters is simple. It leads to the discrete-time system described by

$$x(t+1) = ax(t) + bu(t) + v(t) \tag{4.23}$$

To get the system on familiar form (3.2), we set $\theta = [\theta_1,\ \theta_2]^T = [a,\ b]^T$ and $\phi(t) = [x(t),\ u(t)]^T$, to get

$$z(t) = x(t+1) = \phi^T(t)\theta + v(t) \tag{4.24}$$

When the uncertainty is Gaussian and the random variables are independent, we still have a Gaussian conditional distribution with conditional mean and covariance from (3.7).

The OCP is similar to (4.4)

$$\boxed{\begin{gathered} \min_{u(k),\ k=t,\ldots,N-1} E[\sum_{k=t}^{N-1} x^2(k+1)|\gamma(t)] \\ x(t+1) = \phi^T(t)\theta + v(t) \\ \theta_1(t+1) = \theta_1(t) = a \\ \theta_2(t+1) = \theta_2(t) = b \end{gathered}} \tag{4.25}$$

The hyperstate is still

$$\xi(t) = [x(t), \hat{\theta}(t), P(t)]^T \tag{4.26}$$

with $\hat{\theta}(t)$ the conditional mean and $P(t)$ is the conditional covariance. The main difference is the dimension of the hyperstate. $\hat{\theta}$ now contains two variables, while P contains three variables.

4.2.2 Solving the AOCP with DP

The CE controller for two unknown parameters corresponding to (4.15) is

$$u_{ce}^{*} = -\frac{\hat{\theta}_1 x(t)}{\hat{\theta}_2} \tag{4.27}$$

The Bellman equation for the system is equal the system with one unknown parameter (4.16). The only difference in the solution is that the first expectation can be written

$$\begin{aligned} E[x^2(t+1)] &= E[\phi^T(t)\theta + v(t)] \\ &= (\phi^T(t)\hat{\theta})^2 + R + \phi^T(t)P(t)\phi(t) \end{aligned} \tag{4.28}$$

Chapter 5

Time-Optimal Dual Control of a Cart System with an Unknown Breaking Coefficient

Similar to the problem studied in the project report, we will investigate the use of dual control on a time-optimal control problem. Here the focus will be on a system with an unknown breaking coefficient. The problem to be studied is a time optimal problem, since the cart shall be moved from one position x_0 to a final predefined position x_f in minimum time t_f. In this example the cart shall be stationary at the end position and there are bounds on the control inputs. Only constant desired end states are considered. An illustration of the problem can be seen in Figure 5.1. Also, as it is shown here, only movement in one direction is considered.

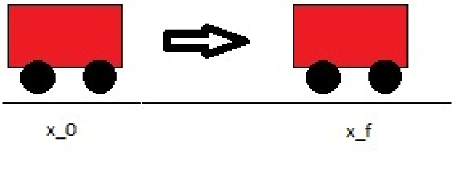


Figure 5.1: Illustration of the Cart problem: Cart moves from position x_0 to x_f

5.1 Cart Model

The cart system has the following simple model in continuous time

$$\ddot{x}(t) = b_1 u_1(t) - b_2 u_2(t) + v(t) \tag{5.1}$$

with x, v, u_1 and u_2 representing position, disturbance, gas and break, respectively. b_1 and b_2 are constant coefficients. They can be either known or unknown. In this thesis the system with known b_1 and unknown b_2 will be considered. By introducing new variables for position $x_1 = x$ and velocity $x_2 = \dot{x}$, this can be written on standard form

$$\dot{x}_1(t) = x_2(t) \tag{5.2a}$$

$$\dot{x}_2(t) = b_1 u_1(t) - b_2 u_2(t) + v(t) \tag{5.2b}$$

The disturbance v can for instance be drag. This can be modelled as

$$v(t) = c x_2^2(t) + w(t) \tag{5.3}$$

with a constant coefficient c and uncertainty w with expected value of zero. A discrete approximation to the system can be found with Euler's method to be

$$x_1(t+1) = x_1(t) + \Delta t x_2(t) \tag{5.4a}$$

$$x_2(t+1) = x_2(t) + \Delta t(b_1 u_1(t) - b_2 u_2(t) + v(t)) \tag{5.4b}$$

where Δt represents the time step between t and $t+1$.

5.2 The Optimal Control Problem

Minimum-time problems are well known optimal control problems. The goal of these types of problems is to minimize the expected end time t_f for when the system reaches a given end state x_f. This can be expressed by the following objective function

$$\min_{u(k),\ k=t,\ldots,N-1} E[\sum_{k=t}^{N-1} \Delta t] \tag{5.5}$$

where $t_f = (N-1) \times \Delta t$. The problem can also be written on a more common form, with a state in the objective function. This is done by augmenting a state x_3 to the system (5.4), with

$$x_3(t+1) = x_3(t) + \Delta t \tag{5.6}$$

and $x_3(0) = 0$. Then we can find an objective written on Mayer-form (Diehl and ESAT-SCD, 2011)

$$\min_{u(k),\ k=t,\ldots,N-1} E[x_3(N)] \tag{5.7}$$

In addition, the system usually has to satisfy some constraints on the states and control inputs. We want to find the optimal control u^* that minimizes the control horizon N, while keeping the states and control within legal values. The solution to the minimum-time problem in continuous time will be of type bang-bang control when there are bounds on the input and no uncertainties in the system. The term bang-bang control means that the control input will be at the maximal or minimal values at all time. Therefore, the problem only consists of finding the switching times. The most common way to solve these minimum-time problems is by the use of Pontryain's minimum principle (Kirk, 2012). The solution to the discrete-time system however, is not necessarily bang-bang control even without uncertainties in the system. This is because of the imperfections that comes with the discrete sampling of the continuous state space.

The analytical solution to the bounded, deterministic cart problem was derived in my project. Since the system consists of two states, we know that the control input can shift at most one time. Below the result is given, with expressions for finding the optimal final time t_f and the switching time t_s.

$$\boxed{t_f = t_s(1 + \frac{b_1\bar{u}_1}{b_2\bar{u}_2})} \tag{5.8}$$

$$\boxed{t_s = \sqrt{\frac{2x_f}{b_1\bar{u}_1(1 + \frac{b_1\bar{u}_1}{b_2\bar{u}_2})}}} \tag{5.9}$$

Where $\bar{u}_1$ and $\bar{u}_2$ are maximum gas and break, respectively. These equations can also be used as guidelines for how the solution should look for a stochastic system, but are not directly applicable.

As a motivation for using dual control, I described in my project the following problem with non-dual control, similar to La et al. (2016). Figure 5.2 shows what will happen in a situation where we have an uncertain breaking coefficient. Here we have an initial estimate of $\hat{b}_2 = 2$, a known parameter $b_1 = 1$, final position $x_f = 1$ and maximum control input $\bar{u_1} = \bar{u_2} = 1$. By using the above formula for t_s in (5.9), we see that this will lead to a switching time of $t_s \approx 1.15$. When this time is reached and the breaking period is initiated, the parameter can be estimated and we will discover that it will not be possible to apply enough breaking to stop at the desired end position x_f. This will lead to unacceptable violation of the constraints.

5.3 AOCP Formulation

In this thesis we will be working with the stochastic, discrete time approximation of the problem in (5.4) and (5.5), with b_2 being an unknown constant. We want

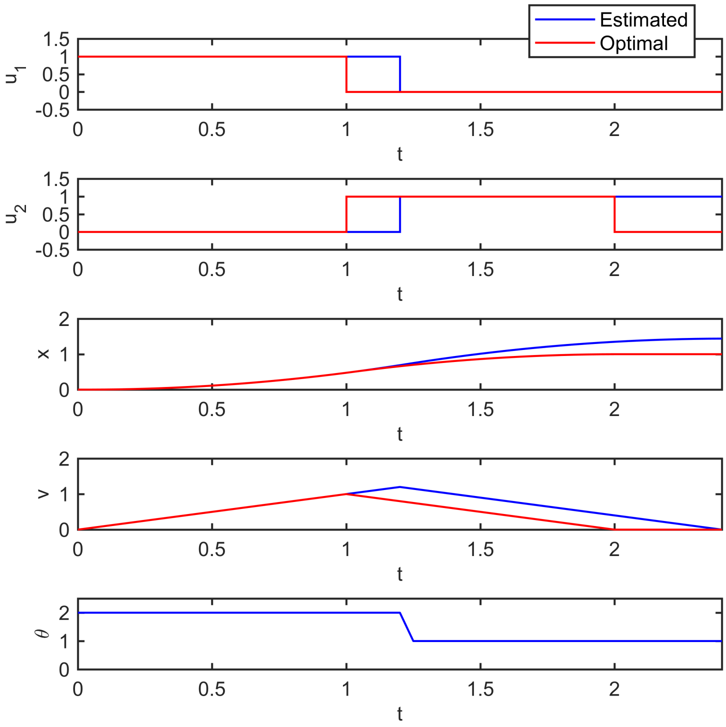


Figure 5.2: Illustration of non-dual control for cart system with unknown breaking coefficient.

to find an optimal control sequence u^* that brings us to the end state as fast as possible. The overall problem made up of (5.4) and (5.5) gives the OCP for the cart system

$$\boxed{\begin{aligned} \min_{u(k),\ k=t,\dots,N-1} & E[\sum_{k=t}^{N-1} \Delta t] \\ x_1(t+1) &= x_1(t) + \Delta t x_2(t) \\ x_2(t+1) &= x_2(t) + \Delta t(b_1 u_1(t) - b_2 u_2(t) + v(t)) \\ \theta(t+1) &= \theta(t) = b_2 \\ u_{min} \leq & u(t) \leq u_{max} \\ x(t_0) &= x_0,\ x(t_f) = x_f \end{aligned}} \tag{5.10}$$

Due to the uncertainties, we do not have complete knowledge of the process. Therefore, we describe both the unknown parameter and the process by stochastic models. To get the system on the familiar form (3.2), we can set $\theta = b_2$, $\phi(t) = -u_2(t)$ and find

$$z(t) = \frac{x_2(t+1) - x_2(t)}{\Delta t} - b_1 u_1(t) = \phi(t)\theta + v(t) \tag{5.11}$$

z has conditional mean and covariance given by

$$\begin{aligned} \mu_z &= \phi(t)\hat{\theta}(t) \\ \sigma_z^2 &= R + \phi^2(t)P(t) \end{aligned} \tag{5.12}$$

where $\hat{\theta}(t)$ is the current estimate of θ or the conditional mean, and $P(t)$ is the conditional covariance.

The hyperstate is set to

$$\xi(t) = [x(t), \hat{\theta}(t), P(t)]^T \tag{5.13}$$

The parameter estimation is done with one of the methods derived in Subsection 3.2, depending on the probability distributions for θ and v. For a Gaussian system we will use the system of equations in (3.5), while for a uniform system we can also use (3.15) and (3.16).

5.4 Solving the AOCP with DP

For this specific problem, the stage cost F is simply the time it takes to get from where we are now and until the next stage. The only problem is that we do not know exactly where this will be, due to the uncertain breaking coefficient and noise that influences the system. This leads to the following value function from (5.10)

$$V(\xi(t), t) = \min_{u(k),\ k=t,\dots,N-1} E[\sum_{k=t}^{N-1} \Delta t | \gamma(t)] \tag{5.14}$$

where $\gamma(t)$ represents the measurements available at time t. The expectation is conditioned on these measurements, and the corresponding Bellman equation can be found by applying the principle of optimality

$$V(\xi(t),t) = \min_{u(k),\ k=t,\dots,N-1} E[\Delta t + V(\xi(t+1),t+1)|\gamma(t)] \tag{5.15}$$

This can be interpreted as the expected total time to get from hyperstate ξ at time t, and is equal to the expected time it takes to get to the next hyperstate plus the remaining time from this hyperstate and until the final state.

The last expression on the right side can be found from (Subsection 3.3.1)

$$E[V(\xi(t+1),t+1)|\gamma(t)] = \int_{-\infty}^{\infty} V(\xi(t+1),t+1) f_z(z)dz \tag{5.16}$$

where $f_z(z)$ is the probability density function for the random variable Z in (5.11). To calculate this expectation of the cost to go, we can use a quadrature formula as described in Subsection 3.3.1. This will lead to an expression on the form

$$V(\xi(t),t) = \min_{u(k),\ k=t,\dots,N} E[\Delta t + \sum_{i=1}^{n} w(i) V(\xi(t+1),t+1)] \tag{5.17}$$

Chapter 6

Testing Environment

In this chapter it is shown how the dual controllers introduced in Chapter 4 and 5 were implemented and tested. First, the framework for the DP algorithms is given and some related issues for the two examples discussed. Later, a method for parallelization of the code is presented and in the end we go into more detail on the goals for the numerical experiments and how they were conducted.

6.1 Implementation of the DP Algorithms

This section should provide a brief description of the implementation of the DP algorithms. All implementations are done in MATLAB and executed on a computer with specification given by Table 6.1.

Table 6.1: Computer specifications

Manufacturer	Dell
Processor	Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
Installed memory (RAM)	32 GB

Prior to running the DP algorithm you need to choose the following system parameters:

- Grid for hyperstates in ξ
- Control law grid U
- Time horizon N
- Time step Δt
- Variance R for the noise v

A general pseudo code for the implementation can be seen below. Here V represents the value function. It is a matrix containing the optimal costs. μ^* is a matrix containing the optimal control inputs. The size of the two matrices will depend on the grids of hyperstates. The total number of loops will depend on the dimension of the hyperstate. In the pseudo code x represents the states, $\hat{\theta}$ the parameter estimates and P the covariance matrix. Each of these can have a dimension larger than one, depending on the problem at hand. Therefore, each of the variables may in practice correspond to multiple for-loops.

Pseudo code

```
for $t = N - 1 : -\Delta t : 1$ do
    for all $x$ do
        for all $\hat{\theta}$ do
            for all $P$ do
                for all $u$ do Calculate expected cost $J(\xi(t), t)$
                    if $J(\xi(t), t) < V(\xi(t), t)$ then
                        $V(\xi(t), t) = J(\xi(t), t)$
                        $\mu^*(\xi(t), t) = u$
                    end if
                end for
            end for
        end for
    end for
end for
```

At each iteration in time, the optimal cost and control input are calculated for each possible combination of the hyperstates. The calculation of the expected cost J will involve the approximation of an integral and interpolation as described in Chapter 2 and 3.

The implementation of the deterministic DP algorithm was compared to the explicit solution to a linear quadratic control problem (Foss and Heirung, 2013) in my project. This gave a good indication on the correctness of the algorithm.

6.2 Integrator Specifications

In this section some properties of the implementation of the DP algorithm for the integrator system will be discussed. Due to symmetry properties for V and μ^* in this problem, it is only necessary to store the values for one quadrant. V is symmetric in the hyperstates, while μ^* is antisymmetric.

As described earlier, the cost function does not depend explicitly on the covariance for the system with one unknown parameter and it is therefore possible to reduce the implementation with one loop with normalized variables. Hence, the implementation will differ slightly depending on if you want to use the normalized variables or not. In terms of the normalized variables, you get a system consisting of four loops. These are time, the two normalized hyperstates and control input, respectively. For the original variables you will have the additional loop representing covariance.

For the system with two unknown parameters in (4.25), the hyperstate is of dimension six. This corresponds to eight loops in the DP algorithm. One for time, one for the state, two for the parameter estimates, three for the elements of the symmetric covariance matrix and one for the control input.

Quadrature

As explained by Åström and Helmersson (1986), there are discontinuities in the first derivative of the cost function V for $\eta(t+1) = x(t+1) = 0$ and $\beta(t+1) = \frac{\hat{\theta}(t+1)}{\sqrt{P(t+1)}} = 0$. These points are

$$\epsilon_1(t) = -\frac{\eta(t) + \beta(t)\nu(t)}{\sqrt{1+\nu^2(t)}} = -\frac{x(t) + \hat{\theta}(t)u(t)}{\sqrt{R + u^2(t)P(t)}} \tag{6.1}$$

and

$$\epsilon_2(t) = -\beta(t)\frac{\sqrt{1+\nu^2(t)}}{\nu(t)} = -\frac{\hat{\theta}(t)\sqrt{R+u^2(t)P(t)}}{u(t)P(t)} \tag{6.2}$$

This have to be considered when approximating the integral in (4.22) (or equivalently for the original variables (4.18)). When there are discontinuities it is important to have enough quadrature points around ϵ_1 and ϵ_2 to capture the true shape of the function.

Since we only will look at Gaussian noise for this example, the approximation is done with a Gauss-Hermite quadrature, as explained in Subsection 3.3.1.

Interpolation

As described in Section 2.2, interpolation has to be used in the DP algorithm. The method used for this problem is cubic interpolation. The state variables are transformed by the mapping $x \to \frac{x}{x+1}$ for the interpolation function. This is done to be able to use a close grid in small values of the variables, and only a few large values. The transformation makes this possible, while still obtaining a uniform grid for interpolation that is in the interval $[0, 1]$. The grid sizes for the variables were varied in the computations.

Note that if you are using MATLAB function *griddedInterpolant*, this is very expensive and therefore the function should be called as few times as possible. This can be ensured by sending in a vector with all Hermite points in one call. This is much faster than going through all points individually and is possible for this problem.

Initialization of Cost Matrix

For the integrator problem, the end conditions are simply

$$V(\xi(N), N) = 0 \tag{6.3}$$

while all other stages are set to a large value.

6.3 Cart System Specifications

In this section some properties of the implementation of the DP algorithm for the cart system will be discussed. This problem has two states and therefore one more loop than the integrator system. The implementation of the DP algorithm consequently consists of six for-loops. The time horizon N in the outermost loop has to be set to a value that is larger than the optimal end time. This can be done with an educated guess based on the solution from a deterministic problem. The inner loops represent the hyperstates and the control input.

The Stopping Criteria

In the minimum-time problem, the goal is to minimize the end time t_f. This means that the control horizon N is an unknown optimization variable. Therefore, it has to be set to a value larger than the optimal end time before running the DP algorithm. For a deterministic minimum-time problem it is easy to obtain a stopping criteria for the algorithm, since we then know with certainty that the first iteration where the cost in the initial state is different from zero represents the optimal end time.

For a stochastic problem on the other hand, it is not trivial to find the optimal end time in a similar manner. It could be possible to rather describe this as a probability $p(y(k) = y(0)$ for the initial state being reached. However, this is not straightforward. Therefore, we will only run the algorithm a few iterations past what is expected to be the optimal stopping time t_f. Then the system can be simulated for different values of $T \leq N$ to find the value that gives the best result.

Another possibility could be to check the current estimated cost from the initial state at each iteration. Initially this will be equal to the maximum cost, but eventually it will decrease. Perhaps it is possible to set a limit such that when the cost is below this threshold, we can end the algorithm and assume that this N is "good enough" by some measure.

Initialization of Cost Matrix

For $t < N$, all elements in V are set to a large value. It is however not obvious how to set the end conditions in the cost matrix. Nevertheless, it is clear that the last stage should have a zero cost in the desired end state and larger costs in the other elements. The simplest way to do this is to give all other elements the same cost. This corresponds to initializing the last stage as

$$V(\xi(N), N) = \begin{cases} 0 & x(N) = x_f \\ C & \text{else} \end{cases} \tag{6.4}$$

where there is no cost when in the desired end state, and C is set to a number larger than the maximum possible end time based on the given parameters. This is also the method used for the deterministic system in my project. It does however have some drawbacks when the system under consideration is stochastic.

Another possibility is to penalise the elements based on how far away they are from the desired end state.

$$V(\xi(N), N) = \begin{cases} 0 & x(N) = x_f \\ c(x_1, x_2, x_f) & \text{else} \end{cases} \tag{6.5}$$

where c is a penalty function. This can be quadratic, linear or a combination of both. The method is of course more complicated, but can have certain advantages. If for instance it turns out that the cart is too far away from the end position to ever reach it within the remaining time horizon, it might just give up if the cost is equal at all other states. Here one could argue that the cart should try to get as close as possible to the desired end state even if it can not reach it exactly. This problem might be avoided by the use of a penalising function. In the calculations carried out here, we will use a penalty function where only the neighbouring elements has a reduced cost.

Handling Values Outside of the State Grid

When calculating the sum in the expected cost (5.17), you will at some points hit values outside of the given grid for $y_2(t+1)$ and $\theta(t+1)$. Since the cart cannot have a negative velocity, $y_2(t+1) < 0$ will be punished by a large cost. This is also the case for $\hat{\theta}(t+1) < 0$, since this will make the problem infeasible. Values larger than the maximum chosen in the algorithm can be allowed, but some care has to be taken. An increase in $\hat{\theta}$ will naturally result in a decrease in the expected cost, since this will mean that the cart needs less time on breaking to reach a zero velocity. Extrapolation in the direction of increasing $\hat{\theta}$ may therefore result in an artificially low expected cost. This can for instant be avoided by using a larger grid for $\hat{\theta}$, or choosing a different extrapolation method.

6.4 Parallelization

Parallelization involves simultaneous execution of parts of the code. Dynamic programming is well-suited for parallelization, since some of the calculations can be carried out independently of each other. This can help reduce the time spent on these calculations. How much the running time can be reduced depends on how many parallel executions that are available and how much delay the transfer of information between the processes introduce, among others. It will not be possible to achieve a reduction to fifty percent of original run time by using two parallel processes due to this aforementioned delay. There are however a lot of time to save if there are many calculations to be done or if they require much computational power each.

It is not uncommon to use parallelization in combination with dynamic programming, due to the large number of calculations that needs to be conducted even for a reasonably small problem. Maidens et al. (2016) show that the execution speed of their dynamic programming algorithm is increased through parallelization, but also that some care needs to be taken in the coordination of the parallel processes.

6.4.1 Parallel for-Loops

In MATLAB parallelization can be achieved with the *Parallel Computing Toolbox* and the *parfor*-loop (MathWorks, c). It works like the standard for-loop, except that the different iterations can be done in parallel. This is achieved through the use of so called *workers* in a parallel pool. They receive data from the client, do most of the calculations and send the results back to the client. The workers evaluate the iterations independently and in no particular order. If nothing else is specified, all available workers will be used.

It is not allowed to use nested parfor-loops and it is not possible to calculate the outer loop in parallel, since the iterations are not independent. If using an inner loop, multiple parfor-loops will be created. We therefore use the second outermost loop for parallelization. The main difference compared to the original implementation presented as pseudo code above, is that the second for-loop is replaced by a parfor statement. This does however introduce some difficulties. For instance you have to iterate over integers, such that the iterations cannot be done in steps Δ that are not integer. Also, the index variable in a nested for-loop cannot be used as part of an expression. It can only be used in plain form (e.g. Array(i,j) and not Array(i,j+1)). This is also true for the for loops that are not parallel as long as an one of the loops are parallel. Therefore, some reformulations might be required.

To get even better performance, it is possible to use the Cloud with MATLAB (MathWorks, b).

6.5 Experiments

In order to evaluate the theory in practice, multiple experiments are conducted. Here we will briefly present the tests that are done on each of the problems and the motivation behind them. First, an overview will be given, and then each problem will be presented briefly.

6.5.1 Overview

After choosing parameters and running the corresponding DP algorithms, we will receive the optimal control inputs $u^*(x(t), t)$ dependent on state and time. This dual control law can then be used for testing.

Common Goals for the Experiments

Experiments are conducted with some variations, on both the examples introduced in Chapter 4 and 5. The exact form of the experiments varies with the different examples however, since they have different properties that we want to highlight. There are nevertheless some common, overall goals for the numerical experiments. These are as follows:

1. Evaluate different discretization accuracies and computation times.
2. Investigate if parallel for loops can speed up the computation time.
3. Evaluate the behaviour of the dual controller up against the CE controller.

The first item is accomplished by running the DP algorithms with different grids and investigating the resulting control matrix, or simulating the systems. Timing of both serial and parallel execution of the code is done in MATLAB, whereas comparison to the CE controller is conducted through simulation. Both single simulations to easier observe the difference in the control input and states, and multiple simulations to obtain a mean performance of the different controllers are performed.

Simulation

To simulate the systems with the dual control laws, at each time step a lookup in the table is performed to extract the optimal control to apply to the system. In these simulations, cubic interpolation is used. For the integrator system, the control law will be independent of time t. For the cart system it will however depend on time. The following parameters have to be chosen prior to simulation

- $\hat{\theta}(0)$ - Initial estimate of θ
- $P(0)$ - Initial variance of θ

- $x(0)$ - Initial state value
- N - Time horizon (Not necessarily the same as the one used in the DP solution, but in the minimum-time problem it has to be equal or less.)

6.5.2 Integrator with One Unknown Parameter

The first problem to be studied is the integrator with one unknown parameter. The main motivation for this problem, besides investigating the items stated above is to:

1. Reproduce the dual controller from Åström and Helmersson (1986) and recreate Figures 2 and 3 presented in the article.
2. Compare the dual controller found for the system with original variables to the dual controller for the normalized system.

By showing that it is possible to obtain similar results as the one produced in the article, the implementation done in the thesis can in many ways be verified. It also makes it possible to compare the runtime then and now.

The DP algorithm is executed with different grid sizes from 16 × 16 to 128 × 128. The number of iterations is also varied. The system is further simulated with the dual control law found by the DP algorithm for $T = 31$. Moreover, a Monte Carlo simulation is used to compare the performance of the dual controller to a CE controller and a cautious controller. The CE control used is

$$u_{ce}(t) = -\frac{x(t)}{\hat{\theta}(t)} \tag{6.6}$$

while the cautious control is

$$u_{cautious}(t) = -\frac{\hat{\theta}(t)x(t)}{\hat{\theta}^2(t) + P(t)} \tag{6.7}$$

Only Gaussian noise and parameter uncertainty are considered. For this problem it is not as interesting to look at faults in the noise assumptions, since it does not impact the problem in the same way as it does for the minimum-time problem. All calculations of the expected cost to go presented in this subsection are done with $n = 52$ Hermite points in the interval $[-2.5348, 2.5348]$. This interval was chosen after close observation of the expected cost to go calculation.

6.5.3 Integrator with Two Unknown Parameters

The main motivation for the problem with two unknown parameters is to

1. Investigate the typical behaviour of the dual controller through simulation

2. See how this problem differs from the one with one unknown parameter

The DP algorithm is run three iterations on a $16 \times 16 \times 16$ grid in x, θ_1 and θ_2. $\Delta P = 0.25$, $P_{ii,max} = 2$, $P_{ij,max} = 1$, $\Delta u = 0.2$ and $n = 14$ Hermite points are used in the calculation of the control law. The control law used in the simulation is the one found by the DP algorithm for $T = 3$. The simulation is done with initial values $x(0) = 0$, $P(0) = [1\ 0; 0\ 1]$, $\hat{\theta}(0) = [0.5\ 0.5]^T$, while a variance $R = 1$ was chosen for the noise.

6.5.4 Cart System without Noise

The motivation behind the minimum-time problem with no noise is to

1. Show how the dual controller handles an unknown breaking coefficient.

The DP algorithms are run with $n = 5$ Hermite (or Legendre) points, and for all simulations we have

- Desired end position $x_1(N) = 1$
- Desired end velocity $x_2(N) = 0$
- Initial position $x_1(0) = 0$
- Initial velocity $x_2(0) = 0$

When the noise v is set to zero, the only uncertainty in the system is the breaking coefficient. The cart system is simulated with calculated dual control laws obtained with two different methods. One where the a priori uncertainty of the parameter is assumed to be Gaussian, and the parameter estimation from (3.5) is used along with a Gauss-Hermite quadrature for approximation of the integral. The second where the a priori uncertainty of the parameter is assumed to be uniform, and the parameter estimation from (3.15) and (3.16) is used together with a Gauss-Legendre quadrature for approximation of the integral. The grid used in the DP algorithms for $x_1 \times x_2 \times \hat{\theta} \times P$ are $12 \times 12 \times 17 \times 5$.

The simulations are done with a few selected initial values for $\hat{\theta}(0)$ to see how the dual controller behaves in the different situations. Additionally, a comparison with a CE controller is conducted.

6.5.5 Cart System with Noise

For the minimum-time problem with noise, the testing environment is mostly the same as for the system without noise, except that v is different from zero. The main motivation for this problem is to

1. See how the dual controller handles additional noise in the system.
2. See how different uncertainty descriptions can affect the solution.

The DP algorithm is run with an expected noise component that is normally distributed with zero mean and variance $R = 0.01$. The a priori uncertainty of the parameter is also assumed to be Gaussian, and the parameter estimation from (3.5) is used along with a Gauss-Hermite quadrature for approximation of the integral. To evaluate the resulting dual control laws, the system is simulated multiple times to find the mean deviation from the desired end state. The noise applied in the simulations will first be drawn from a Gaussian distribution and then from a uniform distribution.

The possibility of improving the performance of the dual controller by using the nonlinear parameter estimation, as described in Subsection 3.2.2, is also investigated.

Chapter 7

Results and Discussion

In this chapter, the results will be presented and discussed. We will look at the solutions obtained from the DP algorithms presented in Section 6.1 and show their performance in the experiments introduced in Section 6.5.

All parallel calculations are done in MATLAB with four workers. This is maximum for the quad-core computer. For the integrator problem, the interpolation method used was *cubic* interpolation. This was chosen because it gave the best results for the given resolution in the state variables. The interpolation method used for the cart problem was *linear* interpolation. This was especially to avoid negative values in the cost function when there were no noise in the system (this can happen if you use cubic interpolation). Also, for the minimum-time problem, the two interpolation methods produced very similar results.

7.1 Integrator with One Unknown Parameter

In this section the results for the integrator system with one unknown parameter will be presented and discussed. First, it will be shown that the dual control laws obtained for the integrator system with one unknown parameter are similar to what Åström and Helmersson (1986) found, and some properties of the control law will be discussed. Later, this will be used to compare with the results from the system with the original variables. In the end the dual control law will be used in simulations and its performance will be evaluated relative to a CE and cautious controller.

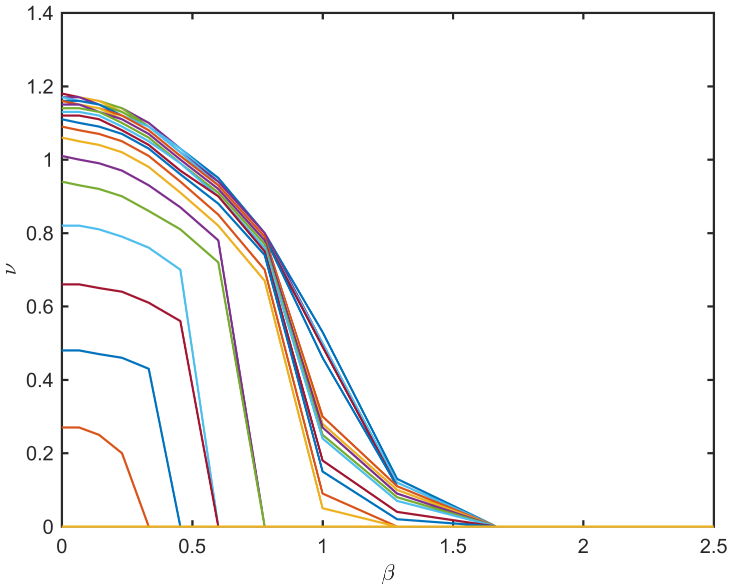


Figure 7.1: Integrator: Dual control laws $\nu(\eta = 0, \beta)$ for different time horizons T.

7.1.1 The Dual Control Law

The DP algorithm in normalized variables was first executed with a grid size of 16×16 in η and β to produce Figure 7.1. This shows a result similar to Figure 3 in Åström and Helmersson (1986), except that here all values of T between 1 and 31 are used. The figure illustrates the different probing zones for the different values of T. As can be seen here, the zone increases with the time horizon. For $T = 1$ there is no probing, for $T = 2$ there is probing for β in the interval $[0, 0.3]$ and so on. For a CE controller the control input will be zero everywhere when $\eta = 0$. The figure also clearly shows that the dual control law is discontinuous. For large T you can notice that the tail differs a little from Figure 3 in Åström and Helmersson (1986). This is due to the resolution of the grid. In Figure 7.2 the same result is shown as a surface plot. Here it is easy to see how the solution is also nonlinear in time. For small T the input changes a lot in only one step. When T increases however, you can see that there is little difference. As Åström and Helmersson (1986) pointed out, the difference in the control policies for $T = 100$ and $T = 31$ is very small. They found that $|\nu_{100} - \nu_{30}| < 0.012$ for all η and β. This is also consistent with the results from the experiments conducted in this thesis.

As mentioned in Section 6.2, the number of Hermite points used is important due to discontinuities. If the number of points is not sufficient, this can lead to a zigzag pattern in $\nu(\eta, \beta)$, especially for large T. The results shown here are found with $n = 52$ Hermite points in the interval $[-2.5348, 2.5348]$. The optimal number of

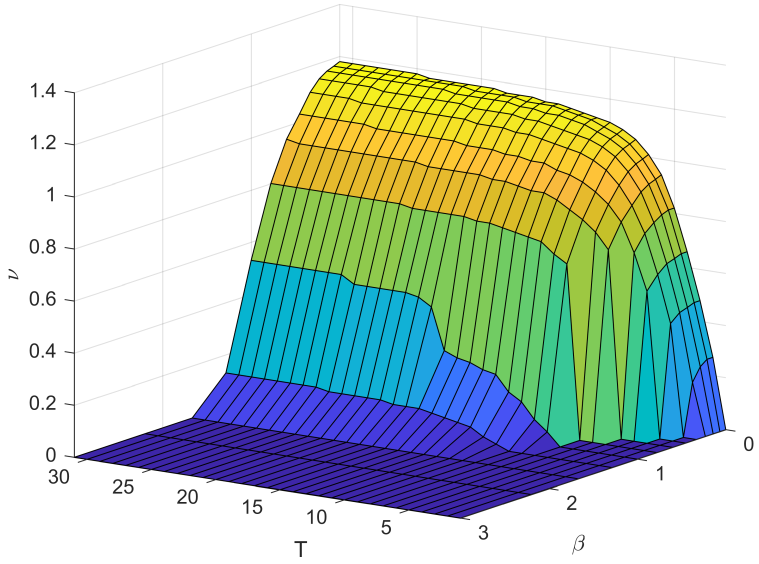


Figure 7.2: Integrator: Surface plot of dual control laws $\nu(\eta = 0, \beta)$ for different time horizons T.

points is not found, but the ones used in these calculations seem to be sufficient for the given grid. Alternatively, you could divide the integration area into multiple intervals as was done by Åström and Helmersson (1986). They used Simpson's algorithm to approximate the integral.

Another difficulty first observed by Åström and Helmersson (1986), is that when calculating the integral in the expected cost to go, you use a set of points in the state space and interpolate for the values between these. The resulting expected cost with the given control ν is calculated as a weighted sum of discrete points. When the control is varied, the set of points used is varied and consequently the interpolation area can also change. This can lead to artificial local minima. To minimize the impact of this, it may help to increase the resolution of the state grids. Regardless, it is important to be aware of these difficulties for the solution to be of a certain quality. Thorough investigation of both the problem and solution is necessary to get a complete understanding of all the elements that affect the problem.

Figure 7.3 is similar to Figure 2 in Åström and Helmersson (1986). To produce this figure, the normalized variables η and β are represented by

$$\eta' = \frac{\eta}{1+\eta} \tag{7.1}$$

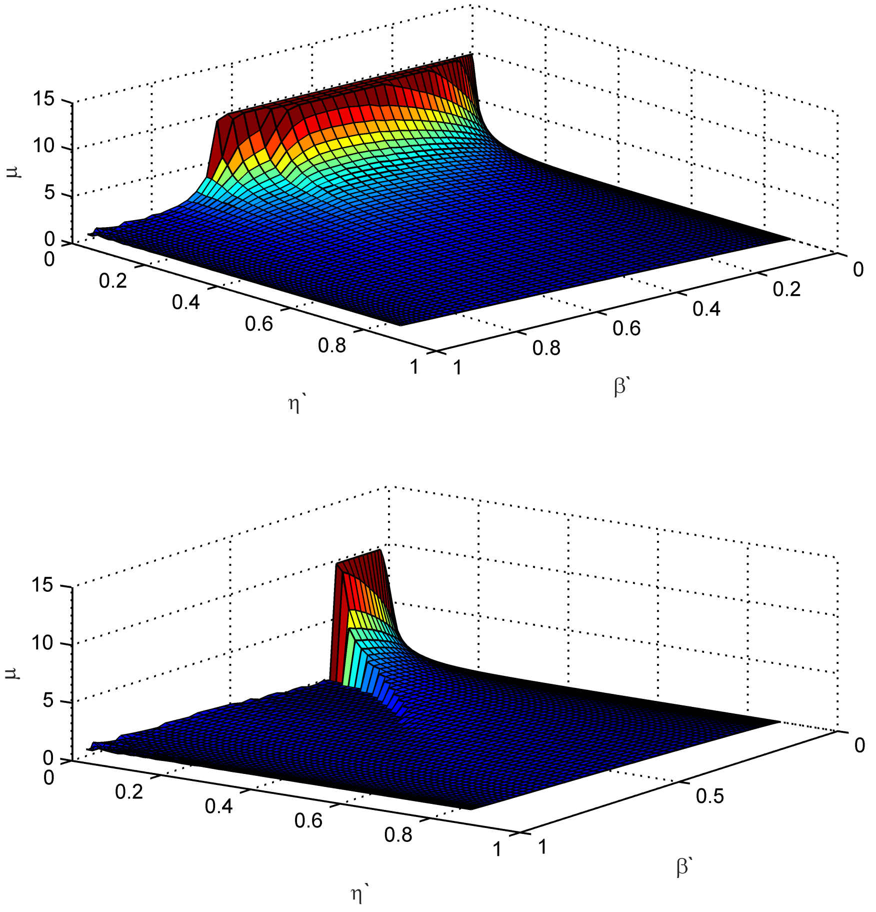
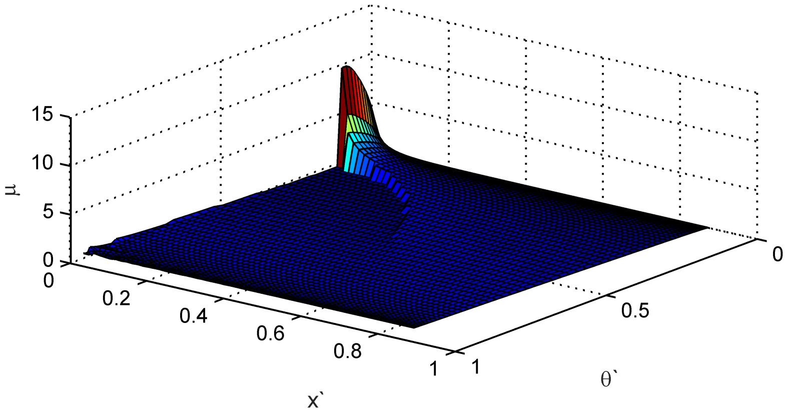


Figure 7.3: Integrator: Dual control laws for $T = 31$ (upper) and $T = 3$ (lower) found with normalized variables.



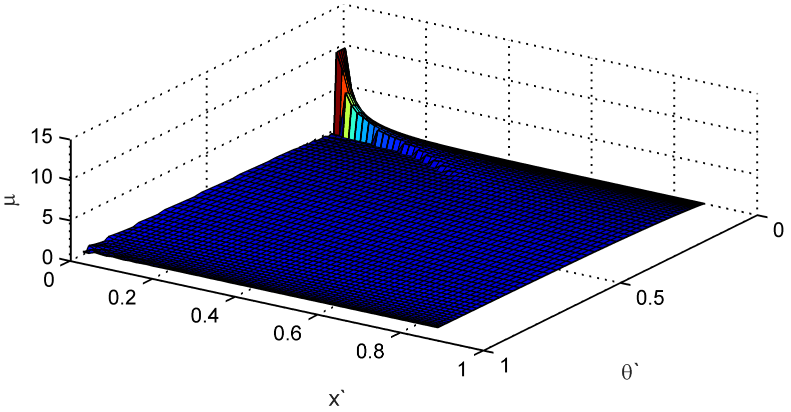


Figure 7.4: Integrator: Dual control laws for $T = 3$ with $P = 1$ (upper) and $P = 0.25$ (lower) found with original variables.

$$\beta' = \frac{\beta^2}{1+\beta^2} \tag{7.2}$$

while the control is scaled as

$$\mu = \frac{\nu\beta}{\eta} \tag{7.3}$$

Figure 7.4 shows the result obtained with the original variables for two different values of P and $T = 3$. The variables x and θ are represented by

$$x' = \frac{x}{1+x} \tag{7.4}$$

$$\theta' = \frac{\hat{\theta}^2}{1+\hat{\theta}^2} \tag{7.5}$$

while the control is still the same. In original variables it is equal to

$$\mu = \frac{u\hat{\theta}}{x} \tag{7.6}$$

It can be noted that for $P = 1$ and $R = 1$, the two representations are identical. The results shown in Figures 7.3 and 7.4 are found for a 64×64 grid in η and β or x and $\hat{\theta}$, respectively. Also, $\Delta\nu = \Delta u = 0.002$ and $\Delta P = 0.25$. The resolutions of the grids were increased to remove numerical errors.

The reason Åström and Helmersson (1986) chose this representation was that CE control is the plane $\mu = 1$ and cautious control is the plane $\mu = \frac{\beta^2}{1+\beta^2}$. This makes it easy to compare the dual controller to the two other controllers. It can be observed from Figure 7.3 that the dual controller approaches the CE controller when the parameter uncertainty represented by β is large. When the control error η is large however, the dual controller agrees with a cautious controller.

If you compare the result for the normalized variables at $T = 3$ in Figure 7.3 with the result for the original variables with $P = 1$ in Figure 7.4, you see that they are not identical. A reason for this can be the coarse grid used for $\Delta P = 0.25$. Despite this, you see that they have a lot in common. It can also be seen that for $P = 0.25$, the volume of the spike is less than for $P = 1$. When the variance is smaller, there will be less probing.

We have seen that it is possible to achieve similar results both with and without using normalized variables. For a system without normalization, the DP algorithm will produce a control law that will also depend on P. By using the control policies found for $P = 1$, you can produce almost identical results. The solution is however also dependent on the grid used for P, which therefore has to have a certain resolution for the results to be adequate. This is apparent especially for higher T. Both representations have some advantages. Clearly the greatest advantage with the normalization is the independence of P, and consequently the shorter runtime for the DP algorithm. In addition, it highlights the importance of the ratio of parameter estimate and variance. On the other hand it can be informative to study the solution for different values of P, while keeping $\hat{\theta}$ constant.

7.1.2 Simulation

To demonstrate the performance of the dual controller found, simulations are performed. First, a typical example of a single simulation will be given. Later, the results from a Monte Carlo approach that compares the controller with a CE controller and a cautious controller will be shown.

Prior to the simulation, the DP algorithm was run on a 16×16 grid. The control law used in the simulation is from $T = 31$. The behaviour of the controller will depend on which T the control law used in the simulation is calculated from. This is especially the case for small T, as was shown in Figure 7.2. The control law used in the simulations shown here is obtained with the DP algorithm for the system with normalized variables. The same result would however be obtained by using the dual control law for the system in original variables.

The system was simulated with the following set of variables: $\hat{\theta}(0) = 1$, $P(0) = 4$, $x(0) = 0$, $R = 1$, $N = 25$. The result from the simulation can be seen in Figure 7.5. This shows the optimal control input and state simulated over the given time horizon. You can see from the figure that the dual control is different from the CE control at the beginning when the uncertainty is large, but very similar after about five steps. This is approximately the same time as the uncertainty is approaching zero.

Notice also that the control input is different from zero at time $t = 0$, even though the state is zero (as was shown in Figure 7.1). In addition, it can be seen that the dual controller not only excites the system to learn, but that it can be more cautious than a CE controller if the uncertainty related to the parameter is large and there are large control errors.

Monte Carlo Simulations

The system was simulated multiple times with the same initial values $\hat{\theta}(0)$, $P(0)$ and $x(0)$, and θ drawn from a Gaussian distribution with mean $\hat{\theta}$ and variance P. Tables 7.1 and 7.2 show the mean cost for 10 000 simulations with a time horizon $N = 25$. Table 7.1 shows how the mean cost varies for different values of $P(0)$ when $\hat{\theta}(0) = 2$ is constant. Table 7.2 on the other hand, shows how the mean cost varies for different values of $\hat{\theta}(0)$ when $P(0) = 1$ is constant.

Table 7.1: Integrator: Comparison of dual, CE and cautious controller. Mean cost over 10 000 simulations for $\hat{\theta}(0) = 2$.

$P(0)$	1	4	7	10
Dual	29	37	42	47
Cautious	30	48	53	58
CE	278	10^7	10^3	10^5

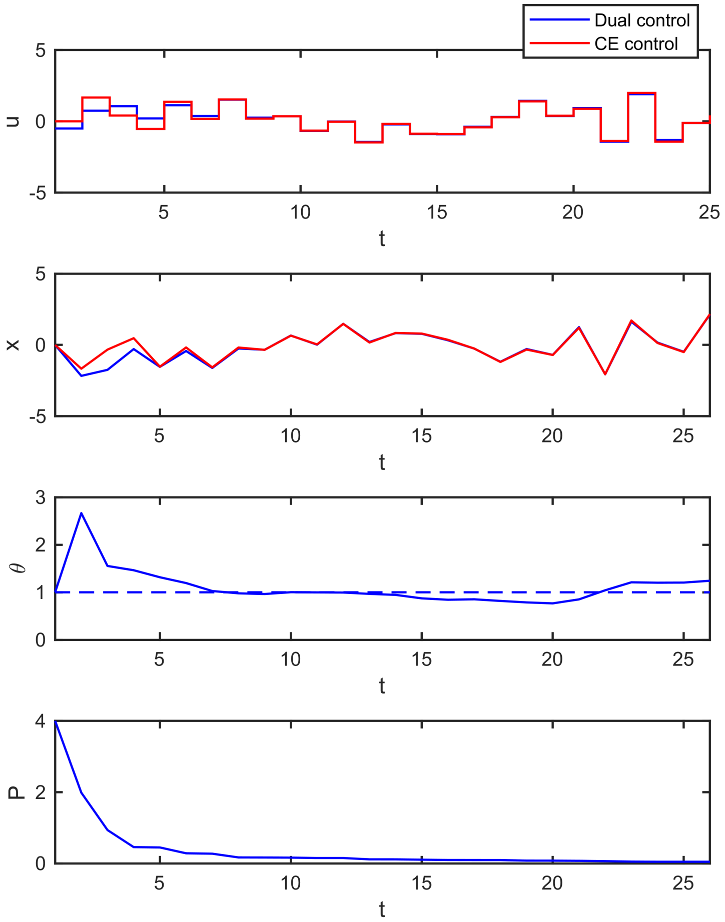


Figure 7.5: Integrator sim 1: System with one unknown parameter simulated with the dual control law.

Table 7.2: Integrator: Comparison of dual, CE and cautious controller. Mean cost over 10 000 simulations for $P(0) = 1$.

$\hat{\theta}(0)$	-4	-3	-1	1	3	4	7	10
Dual	26	26	32	32	26	26	25	25
Cautious	26	26	46	46	26	26	25	25
CE	26	30	10^3	10^3	42	26	25	25

Table 7.1 and 7.2 clearly show that the mean cost is large with the CE controller compared to the two other controllers. It should also be noted that there are large variations in the mean cost with this CE controller when the initial parameter estimate is small relative to the initial variance. Therefore, the cost shown for the CE controller will vary if the simulation is executed multiple times. The high costs are due to estimates $\hat{\theta}(t) \approx 0$. This leads to the CE controller applying a very high control input to the system. Therefore, for small values of the parameter relative to the variance, this CE controller is not the best choice. This is not a problem with the dual or cautious controller. For a large parameter estimate $|\hat{\theta}(0)| = 4$ and a low relative variance $P(0) = 1$, the three controllers have a very similar behaviour. It can also be noted that the cost is symmetric in $\hat{\theta}(0)$.

The clear advantage of the dual controller compared to the other two controllers is apparent when the variance is large relative to the parameter estimate. This is due to the experimentation conducted by the dual controller. There do however exist more sophisticated heuristic adaptive controllers than the ones shown here, that may have a better performance.

7.2 Integrator with Two Unknown Parameters

In this section, a simulation with the dual control law obtained for the integrator system with two unknown parameters will be presented. Since this problem requires significantly larger computation time than the system with one unknown parameter, the DP algorithm was only executed $N = 3$ iterations.

The control law used in the simulation is from $T = 3$. The simulation was done with initial values $x(0) = 0$, $P(0) = [1\ 0; 0\ 1]$, $\hat{\theta}(0) = [0.5\ 0.5]^T$, while the true parameters were $\theta = [1\ 1.5]^T$. The noise has variance $R = 1$. Results from a typical simulation can be found in Figure 7.6.

In general, the dual controller appears to be a lot more aggressive than the CE controller for this problem. If the parameter uncertainty for one parameter is set to zero however, the behaviour is exactly as the one found for the system with only one unknown parameter. This can be said to verify parts of the dual controller shown here. A reason for the controller using what might be an excessive amount of control input, is that the uncertainty does not decrease enough. Moreover, when there are two unknown parameters, there is an extra term corresponding to the covariance that is different from zero.

Since the extension to two unknown parameters means introducing three new hyperstates, the DP algorithm has an exponential increase in runtime. Therefore, the grids used for this solution is coarser than for the system with one unknown parameter. Furthermore, the number of Hermite points is much smaller. Since the result is found only for $T = 3$, this might be sufficient. At least for a T of this magnitude, it did not introduce any error to the system with one unknown parameter.

7.3 Runtime for the DP Algorithms for the Integrator System

This section will present the runtime for the DP algorithms and see how this varies for the different versions of the integrator problem that we have looked at, and for different sizes of the state grid.

Table 7.3 shows the runtime for one iteration of the DP algorithm for the integrator system with original variables. Runtime for the systems with both one and two unknown parameters are given. It can be seen that the reduction in runtime obtained by parallel computations for the system with two unknown parameters is much larger than for the system with one unknown parameter. By using parallelization for this system, it is possible to achieve a runtime of almost one third of the serial program. The reason for this is that there is a very large number of calculations done for each worker, compared to the system with only one unknown parameter, resulting from the three extra for-loops.

Table 7.3: Integrator: Runtime of DP algorithm for original variables with one and two unknown parameters, respectively. Grids used are $16 \times 16 \times 9$ and $16 \times 16 \times 16 \times 9 \times 9 \times 5$. $\Delta u = 0.2$ and $|u_{max}| = 5$.

	Grid size	16×16
One unknown	Parallel	1.0 s
	Serial	1.0 s
Two unknowns	Parallel	3380 s
	Serial	9331 s

Table 7.4 shows the result for different sizes of the grid for ν and β or x and $\hat{\theta}$ from 16×16 to 128×128. This is also for a single iteration of the algorithm.

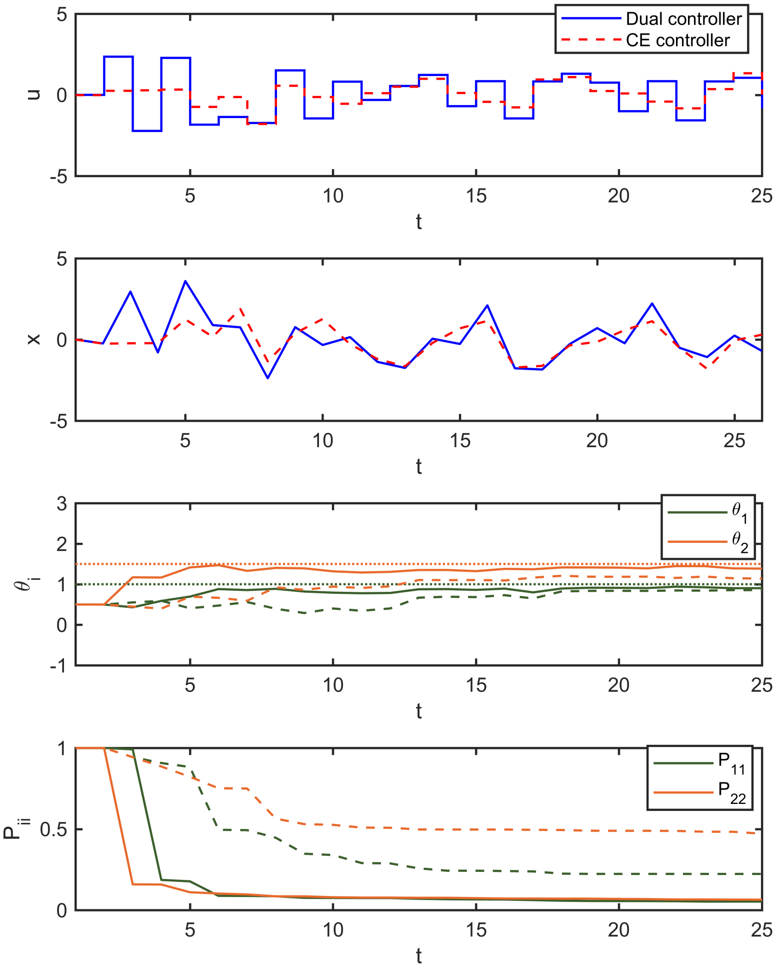


Figure 7.6: Integrator sim 2: System with two unknown parameters simulated with the dual control law. (Dotted lines are true parameter values.)

Table 7.4: Integrator: Runtime of DP Algorithms for different grid sizes. The result is for a system with one unknown parameter. $\Delta\nu = \Delta u = 0.01$, $|u_{max}| = 10$ and 5 points is used to represent P for the original variables.

	Grid size	16×16	32×32	64×64	128×128
Normalized variables	Parallel	3 s	11.5 s	44 s	173.5 s
	Serial	4.5 s	17 s	67 s	274.5 s
Original variables	Parallel	17.5 s	71.5 s	275 s	1127 s

Difference in runtime with and without normalized variables for one iteration is naturally large, due to the extra variable in the system of original variables. The grid for the extra variable also influences the runtime. Despite the grid for P being relatively coarse in these calculations, the runtime for the system with the original variables is much larger.

One iteration of the algorithm for a 64×64 grid in η and β for the normalized system took just above 40 seconds. In comparison Åström (1983) documented that their program required 180 CPU-hours for solving the problem for $T = 30$ on their DEC VAX 11/780. This means that one iteration of the Bellman equation took 6 hours, which is about five hundred times as much as for our program. Clearly, it has been a major development in the hardware since then.

The parallelization in the DP algorithms is done in η. This means that the number of iterations for the workers increase with the grid size. From both Tables 7.3 and 7.4, we can see that the runtime is reduced with the parallel for loop. However, the runtime is still more than half of the runtime compared to the program without parallelization, even though there are four workers. A reason for this is that some matrices are required as a whole in the inner loops. MATLAB then sends the entire matrix to each worker, resulting in high data communication overhead. Despite of this, there is definitely possible to reduce runtime by parallelization of the code. There may also be possible to do this in a more sophisticated manner. Here we have only used the built in function in MATLAB. It might be possible to do even better with a custom made function for our system.

7.4 Cart System without Noise

In this section the results from the experiments with the dual controller for the cart system without noise will be presented and discussed.

As mentioned in Section 6.3, it is not trivial to find the optimal stopping time for the DP algorithm. Experimentation is therefore used to choose an end time t_f. Table 7.5 illustrates the deviation from the desired end state when simulating the cart system with the dual control law from two different values of the initial parameter estimate, and with different end times. The deviation is defined as

$$\epsilon_x := |x_1(N) - x_{1f}| \tag{7.7}$$

$$\epsilon_v := |x_2(N) - x_{2f}| \tag{7.8}$$

where x_{1f} and x_{2f} are the desired end position and end velocity, respectively.

The result shown here indicates that the optimal end time at least should be $t_f \geq 2.1$. Since there are little or no improvement by simulating longer than this, $t_f = 2.1$ is chosen in the simulations shown here.

Table 7.5: Minimum-time: Deviation from desired end state for different values of t_f and $\hat{\theta}(0)$, without noise. The result shown is for $P(0) = 1$ and $\theta = 1$

	$\hat{\theta}(0) = 0.25$	$\hat{\theta}(0) = 2$
$t_f = 2.0$	$\epsilon_x = 0.08$, $\epsilon_v = 0.16$	$\epsilon_x = 0.37$, $\epsilon_v = 0.93$
$t_f = 2.1$	$\epsilon_x = 0.02$, $\epsilon_v = 0.02$	$\epsilon_x = 0.02$, $\epsilon_v = 0.03$
$t_f = 2.2$	$\epsilon_x = 0.01$, $\epsilon_v = 0.03$	$\epsilon_x = 0.02$, $\epsilon_v = 0.03$
$t_f = 2.3$	$\epsilon_x = 0.02$, $\epsilon_v = 0.03$	$\epsilon_x = 0.02$, $\epsilon_v = 0.03$

The dual control laws obtained for the system with a uniform and Gaussian a priori distribution for θ was used in the simulation. Both these approaches did, as expected, produce the same result in the simulations. Figure 7.7 shows the result from a simulation with initial estimate $\hat{\theta}(0) = 2$ and covariance $P(0) = 1$, and Figure 7.8 shows the result from a simulation of the system with initial estimate $\hat{\theta}(0) = 0.25$ and covariance $P(0) = 1$. Since there is no noise in the system, a correct parameter estimate is found in both simulations within only one time step. The deviation from the desired end state is, as shown in Table 7.5, $\epsilon_x = 0.02$ and $\epsilon_v = 0.03$ in the first simulation and $\epsilon_x = 0.02$ and $\epsilon_v = 0.02$ in the second. It should be noted however that due to the discretization used, there will be an error of the same magnitude even for the deterministic system. A grid with a higher resolution will decrease this error, but also increase runtime.

The discrete space approximation is also the main reason for the oscillations in the input signals. This is especially apparent in Figure 7.7. The optimal control law that the DP algorithm chooses, and which is shown here, is just one out of multiple equally good solutions. Even though increasing the resolution of the grid used for position x_1 could decrease the deviation from the desired end position, it also strengthens the need for a proper initialization of the cost matrix. If $\Delta x_1 = 0.01$ is used and $x_1(N) = 1.01$ has the same cost as $x_1(N) = 0$ for instance, this will result in the cart having trouble reaching the final destination.

As can be seen from the figures, the dual controller introduces a nonzero input signal u_2 prior to the estimated switching times. For $\hat{\theta}(0) = 0.25$, this happens already after one step. Since the cart has to have a nonzero velocity before breaking, it can not break at $t = 0$. For $\hat{\theta}(0) = 2$ on the other hand, the breaking does not appear until $t = 0.9$. This suggests that when the initial guess of the estimate $\hat{\theta}(0)$ is small relative to the initial variance $P(0)$, the cart will break earlier and thereby decrease the uncertainty related to the parameter faster. This is also intuitive, since a less sensitive break is less likely to result in a large cost. When the initial estimate is large however, the breaking will not begin until later, since the probability that this may lead to a very small (or negative) velocity is too high. Negative velocity

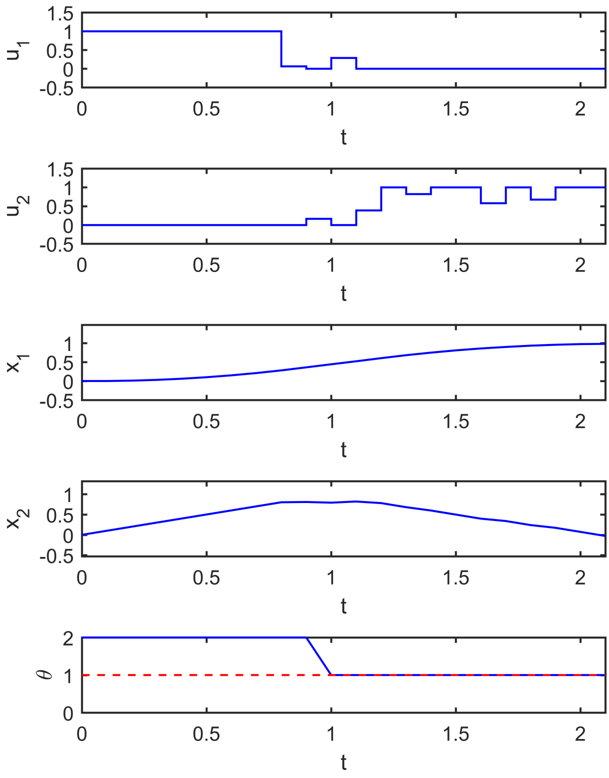


Figure 7.7: Cart sim 1: System without noise simulated with dual control law. Initial estimate $\hat{\theta}(0) = 2$ and variance $P(0) = 1$.

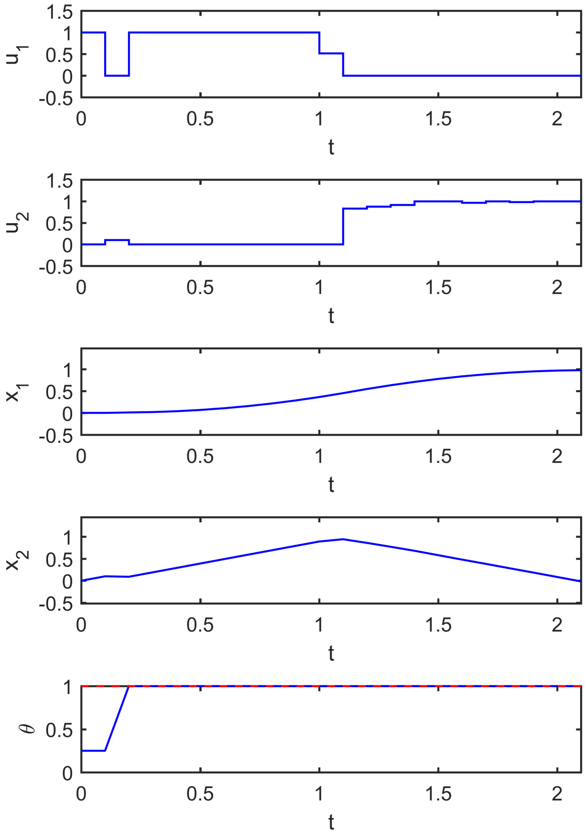


Figure 7.8: Cart sim 2: System without noise simulated with dual control law. Initial estimate $\hat{\theta}(0) = 0.25$ and variance $P(0) = 1$.

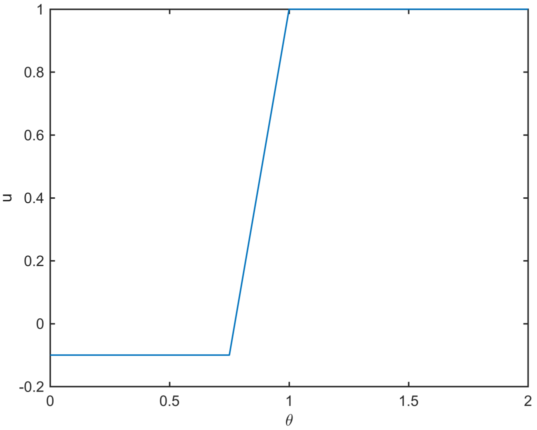


Figure 7.9: Cart: Dual control law $u^*(y_1 = 0, y_2 = 0.1, \hat{\theta}, P = 1)$ at $t = 0.1$ for $t_f = 2.1$

is impossible in practice, and is therefore represented as a high cost in the DP algorithm.

As Figure 7.9 shows, there is a nonlinear relationship between the estimate and the dual control law. At the second time step, the dual controller will add a perturbation signal for $\hat{\theta} \leq 0.75$. These values for the parameter estimate also correspond to expected end times $t_f > 2.1$ calculated from (5.8) for a deterministic system. The best course of action will also depend on how much time you want to use. It can be possible to achieve a smaller deviation from the end state by using longer time.

A CE controller would not have shown the same behaviour as the dual controller that has been presented here. It would rather have started breaking after the optimal switching time, as shown in Figure 5.2. In this example, the deviation from the desired end state at time $t = 2.1$ is $\epsilon_x = 0.39$ and $\epsilon_v = 0.3$. Hence, the dual controller clearly has some advantages over the simple CE controller for these types of optimal control problems. It should also be noticed that the optimal end time found with dual control can be larger than with a CE controller, but that the end conditions should be better. Therefore one cannot necessarily only compare the optimal end times found.

7.5 Cart System with Noise

This section will investigate how the dual controller can handle both Gaussian and uniformly distributed disturbances in the system.

The result shown in Table 7.6 is obtained with the DP algorithm for a system that is assumed to have an unknown parameter and noise that are normally distributed. The inclusion of normally distributed noise is properly handled by the dual controller. The mean deviation from the end state is not much larger for this system than the system without noise, and the behaviour is otherwise very similar to what was shown in Figure 7.7 and 7.8 for the two different initial parameter estimates. It can however be seen from the results that for this particular system, applying noise from a different probability distribution can have a big impact. For $t_f = 2.1$, the mean error in position is six times as large for the system that has been applied noise drawn from a uniform distribution.

The numbers in the parentheses show the resulting mean deviations if the nonlinear parameter estimation is used in the simulations with the uniformly distributed noise. This indicates that ϵ_x can be reduced with the nonlinear parameter estimation, even though the reduction is minimal. It can be noted that the mean deviation from desired end velocity, ϵ_v is not necessarily as informative as ϵ_x. Since the initial state is $x(0) = [0\ 0]^T$, the cart will have $\epsilon_v = 0$ at time $t = 0$.

Table 7.6: Minimum-time: Mean deviation from desired end state over 5000 simulations, for different values of t_f and probability distributions for v. $\hat{\theta}(0) = 0.25$, $P(0) = 1$ and $\theta = 1$.

Distribution of $v(t)$	Gaussian	Uniform
$t_f = 2.0$	$\epsilon_x = 0.41$, $\epsilon_v = 0.32$	$\epsilon_x = 0.47(0.43)$, $\epsilon_v = 0.26(0.34)$
$t_f = 2.1$	$\epsilon_x = 0.09$, $\epsilon_v = 0.02$	$\epsilon_x = 0.36(0.29)$, $\epsilon_v = 0.13(0.14)$
$t_f = 2.2$	$\epsilon_x = 0.04$, $\epsilon_v = 0.02$	$\epsilon_x = 0.20(0.16)$, $\epsilon_v = 0.05(0.05)$
$t_f = 2.3$	$\epsilon_x = 0.04$, $\epsilon_v = 0.02$	$\epsilon_x = 0.15(0.11)$, $\epsilon_v = 0.04(0.04)$

In Subsection 3.3.1 it was suggested that it might be possible to derive a DP algorithm that could better handle disturbances drawn from a uniform probability distribution by improving the parameter estimation method. As described there, the assumption that the noise and parameter are uniformly distributed introduces some difficulties due to the resulting distribution of the sum of the two uniformly distributed variables not being uniform. It was therefore, based on simulations, proposed that it might be sufficient to still use a Gauss-Hermite quadrature to approximate the expected cost to go. This method with the nonlinear parameter estimation and the Gauss-Hermite quadrature did however not produce an adequate result in the calculations done here. It could be possible to pursue the method further by finding weights based on the probability distribution function of the sum directly, but I am not sure that it will be the best use of time. After all, the difference in mean variance for the two parameter estimation methods found in the example in Subsection 3.2.2 was less than 0.01 when the noise was in the interval $[-1,\ 1]$. For this minimum-time problem, the interval is even smaller and therefore it will most likely not be much to gain by pursuing this path for the problems considered here. If however the variance had been very high, it might be more to gain as was shown in the previously mentioned example.

It can be that there were other problems related to for instance initialization of the cost matrix that could have improved the results for the method with nonlinear parameter estimation and the Gauss-Hermite quadrature. The results indicate that much of the reason for the increase in the deviation from the desired end state is due to the controller giving up. Therefore, it might also be possible to achieve better results with the dual controller derived with the Gaussian assumption when the noise is drawn from a uniform distribution, by giving the end conditions more consideration.

7.6 Runtime for the DP Algorithm for the Cart System

In this section we will present the runtime for the DP algorithm for the minimum-time problem and see how this varies for different sizes of the state grid. Table 7.7 shows the runtime for one iteration of the DP algorithm with and without parallelization for the given grids.

Table 7.7: Minimum-time: Runtime of DP Algorithm. $\Delta u = 0.1$, $|u_{max}| = 1$ and grid is given by $x_1 \times x_2 \times \hat{\theta} \times P$.

Grid size	$12 \times 12 \times 17 \times 5$	$111 \times 12 \times 17 \times 5$
Parallel	10 s	69 s
Serial	12.5 s	113 s

As one can expect, the runtime for the problem with an increased resolution in x_1 is many times that of the coarser grid. The runtime shown in the table is for the program where parallelization is done in the state x_1. Exchanging the two states x_1 and x_2 for the grid where the resolution of x_1 is increased, such that the parallelization is done for x_2, results in a longer runtime of about five seconds for each iteration of the algorithm.

7.7 Common Observations

Solving adaptive optimal control problems with dynamic programming is challenging, and the many different factors affecting the problem make it difficult to analyse the solutions obtained. These factors range from discretization of the continuous space and how to handle values outside of the grid and in between the discrete values, to parameter estimation and numerical approximation of the integral in the Bellman equation. There are a lot of choices to be made, and it is important to remember that these choices can have not only an individual effect on the dual controller, but also the interplay between them has to be considered. Moreover, every problem is different and what works for one problem does not necessarily work for the next.

Small changes in the implementation can have a great impact on the solution. Runtime depends heavily on choices done in the implementation, since there are a lot of computations to be done. Most of the runtime for the DP algorithms is however spent on the interpolation. The programs were not written with the main focus on runtime, and it is therefore most likely room for improvement. It has nevertheless been given some consideration, and I believe that what is found here is a good example of what is possible to achieve with the current computational power.

The two main problems presented in this thesis are in many ways very different, but the main features of a dual controller are present for both the problems. Although the dual controllers designed in this thesis performed well, the method is unfortunately still not scalable to larger systems. Therefore, it is interesting to explore alternative methods that can provide the same dual properties. Wittenmark (2002) suggested designing a cautious controller and adding a perturbation signal to the system to increase learning. The additional signal can be white noise for instance. This was also shown by La et al. (2016) in their example with a tractor passing a corner. They found that measurement noise improved the parameter estimates by acting as an excitation signal. The major drawback with this method is naturally that it is not possible to apply this signal in any systematic way. Heirung et al. (2015) on the other hand, derived a more systematic approach by adding a term representing the uncertainty to the objective function. Their controller converged to a CE controller when the uncertainty was decreased and thereby avoided unnecessary loss.

As we have seen here, dual control adds a lot of complexity to the problem, so it is not necessarily the best solution in general. There are however certain situations where it can be worth the extra effort. Åström and Wittenmark (2013) mention especially the situations where you have a very short time horizon and it is important to get a good estimate of the unknown parameters immediately and when the parameter is varying rapidly and can change sign. Even though only constant parameters are considered in this thesis, Åström and Wittenmark (2013) showed that the same methods can be used for a system with varying parameters. As we have demonstrated here, the minimum-time problem with conditions on the end state is also a typical problem where dual properties of the controller can be desirable.

Chapter 8

Conclusion and Future Work

This thesis has given an overview of adaptive optimal control problems and how to solve them using dynamic programming. Multiple problems have been formulated and solved numerically. It has been shown that it is possible to reproduce the solution to the dual control problem first solved by Åström and Helmersson (1986), and how the problem can be extended to include two unknown parameters. Also, dual control has proved to give an elegant solution to a minimum-time problem with an unknown breaking coefficient. Different parameter estimation methods have been derived and the effect of introducing noise to the control problem has been investigated. Experimentation with the dual control laws has illustrated how the optimal solutions to adaptive control problems manage to be both cautious and to work actively to reduce the uncertainty related to the model parameters. It has been demonstrated how this distinguishes the dual controller from some ordinary, heuristic adaptive controllers.

Calculations with different discretization accuracies indicate that the runtime depends heavily on the grids used. It is shown that one iteration of the DP algorithms can be done within a matter of seconds on a fairly large grid for the problems with one unknown parameter. The extended problem with two unknown parameters illustrates how the curse of dimensionality affects the problem with a drastic increase in runtime. Despite this, it is shown that it is possible to solve this problem with dual control too, and that the resulting control law possesses the same properties as for the system with one unknown parameter. Results from calculations for all the problems also clearly indicate that runtime for the DP algorithms can be reduced by running the calculations in parallel.

For the future, it can be useful to improve the algorithms made in the thesis, to investigate further the possibility of reduced runtime and memory usage. Especially for the problem with multiple unknown parameters, where the runtime is the longest. If the runtime for this problem can be reduced, it will be easier to run more experiments that can contribute to an even deeper insight. For the minimum-time problem, it can be advantageous to look closer into the derivation of a proper stopping criteria and a penalty function for the initialization of the cost matrix. Possible improvements could contribute to a more robust dual controller for this problem, and perhaps better performance in the presence of uniformly distributed noise. Additionally, it would be interesting to compare the results from this approach with DP to other suboptimal controllers. Furthermore, it can be a lot to gain on using the new insight obtained about AOCP and DP to investigate approximate methods for active excitation.

Appendix A

Statistics

A.1 Uniform Distribution

The uniform probability distribution is described by an interval $[a, b]$, where the random variable can be found anywhere with equal probability. The expected value of a uniformly distributed random variable X is

$$\mu = E[X] = \frac{a+b}{2} \tag{A.1}$$

while the variance is

$$\sigma^2 = Var(X) = E[X^2] - E[X]^2 = \frac{(b-a)^2}{12} \tag{A.2}$$

It has the probability density function

$$f(X = x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b, \\ 0 & \text{else} \end{cases} \tag{A.3}$$

with a and b upper and lower limits of the interval, respectively. In terms of mean and variance this can be expressed as

$$f(X = x) = \begin{cases} \frac{1}{2\sigma\sqrt{3}} & -\sigma\sqrt{3} \leq x - \mu \leq \sigma\sqrt{3}, \\ 0 & \text{else} \end{cases} \tag{A.4}$$

This suggests that $a = \mu - \sigma\sqrt{3}$ and $b = \mu + \sigma\sqrt{3}$.

The expected value of a function $m(X)$, where X is a uniformly distributed random variable can be found from

$$E[m(x)] = \int_{-\sigma\sqrt{3}+\mu}^{\sigma\sqrt{3}+\mu} f(x)m(x)dx = \int_{-\sigma\sqrt{3}+\mu}^{\sigma\sqrt{3}+\mu} \frac{1}{2\sigma\sqrt{3}} m(x)dx \tag{A.5}$$

A.2 Normal Distribution

The normal probability distribution is bell shaped, with the largest probability centred at the expected value of the random variable X

$$\mu = E[X] = \int_{-\infty}^{\infty} x f(x) dx \tag{A.6}$$

and with variance

$$\sigma^2 = E[X^2] - E[X]^2 \tag{A.7}$$

and probability density function

$$f(X = x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{A.8}$$

The expectation of a function $m(X)$ that depends on the random variable X, is given by the following expression

$$E[m(x)] = \int_{-\infty}^{\infty} m(x) \frac{1}{\sqrt{2\sigma_x^2\pi}} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} dx \tag{A.9}$$

Bibliography

Åström, K. J. (1983). Theory and applications of adaptive controla survey. *Automatica*, 19(5):471–486.

Åström, K. J. and Helmersson, A. (1986). Dual control of an integrator with unknown gain. *Computers & Mathematics with Applications*, 12(6):653–662.

Åström, K. J. and Wittenmark, B. (2013). *Adaptive control.* Courier Corporation.

Bar-Shalom, Y. and Tse, E. (1974). Dual effect, certainty equivalence, and separation in stochastic control. *IEEE Transactions on Automatic Control*, 19(5):494–500.

Bellman, R. (1954). The theory of dynamic programming. Technical report, DTIC Document.

Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific Belmont, MA, 3rd edition.

Bertsekas, D. P. (2011). *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific Belmont, MA, 3rd edition.

Brown, R. G. and Hwang, P. Y. (2012). *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions.* Wiley, 4 edition.

Diehl, M. and ESAT-SCD, K. (2011). Numerical optimal control draft.

Feldbaum, A. (1960). Dual control theory. *Automn. Remote Control*, 21.

Foss, B. and Heirung, T. A. N. (2013). Merging optimization and control. *Norwegian University of Science and Technology.*

Heirung, T. A. N. (2016). Dual control: optimal, adaptive decision-making under uncertainty.

Heirung, T. A. N., Foss, B., and Ydstie, B. E. (2015). Mpc-based dual control with online experiment design. *Journal of Process Control*, 32:64–76.

Heirung, T. A. N., Ydstie, B. E., and Foss, B. (2017). Dual adaptive model predictive control. *Automatica*, 80:340–348.

Ioannou, P. A. and Sun, J. (2012). *Robust adaptive control.* Courier Corporation.

Kirk, D. E. (1967). An introduction to dynamic programming. *IEEE Transactions on Education*, 4(10):212–219.

Kirk, D. E. (2012). *Optimal control theory: an introduction.* Courier Corporation.

La, H., Potschka, A., Schlöder, J., and Bock, H. (2016). Dual control and information gain in controlling uncertain processes. *IFAC-PapersOnLine*, 49(7):139–144.

Lee, J. M. and Lee, J. H. (2009). An approximate dynamic programming based approach to dual adaptive control. *Journal of process control*, 19(5):859–864.

Maidens, J., Packard, A., and Arcak, M. (2016). Parallel dynamic programming for optimal experiment design in nonlinear systems. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 2894–2899. IEEE.

MathWorks. Interpolating gridded data. [Online; accessed 11-November-2016].

MathWorks. Parallel computing on the cloud with matlab. [Online; accessed 17-January-2017].

MathWorks. Parallel for-loops (parfor). [Online; accessed 17-January-2017].

Powell, W. B. (2011). *Approximate Dynamic Programming: Solving the curses of dimensionality.* John Wiley & Sons, 2nd edition.

Servi, L. and Ho, Y. (1981). Recursive estimation in the presence of uniformly distributed measurement noise. *IEEE Transactions on Automatic Control*, 26(2):563–565.

Shapiro, A., Tekaya, W., Soares, M. P., and da Costa, J. P. (2013). Worst-case-expectation approach to optimization under uncertainty. *Operations Research*, 61(6):1435–1449.

Thompson, A. M. and Cluett, W. R. (2005). Stochastic iterative dynamic programming: a monte carlo approach to dual control. *Automatica*, 41(5):767–778.

Wittenmark, B. (2002). Adaptive dual control. *Control Systems, Robotics and Automation, Encyclopedia of Life Support Systems (EOLSS), Developed under the auspices of the UNESCO.*

Zarowski, C. J. (2004). *An introduction to numerical analysis for electrical and computer engineers.* John Wiley & Sons.