Optimal Control under Quantised Measurements

A Particle Filter and Reduced Horizon Approach

Martin Vikström Morin

LUND UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6031
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2017

Abstract

This thesis covers the optimal control of stochastic systems with coarsely quantised measurements. A particle filter approach is used both for the estimation and control problem. Three main families of particle filters are examined for state estimation, standard SIR filters, SIR filters with generalised sampling and auxiliary filters. A couple of different proposal distributions and weight functions were examined for the generalised SIR and auxiliary filter respectively. The choice of proposal distribution had the greatest impact on performance but the unrivalled best filter was achieved with a combination of generalised sampling and the auxiliary particle filter. For the problem of control the particle filter was used for cost-to-go evaluation by forward simulation in time. Simplifications of the full dynamic programming problem were done by reducing the time horizon resulting in M-measurement feedback policies and a new M-measurement cost feedback policy. One-measurement feedback and M-measurement cost feedback was examined for $M \leq 4$ and although probing behaviour was observed none of the examined controllers managed to outperform a certainty equivalent controller.

Acknowledgements

I would like to thank my supervisor Anton Cervin and the rest of the Department of Automatic Control for allowing to do my thesis work there. I asked for freedom to explore and it was given to me in plenty. I would also like to thank Erik Ottosson, Markus Greiff, Joakim Guth and Anders Engström who have shared room or worked on their theses in parallel with me. Being exposed to a wide range of problems is always stimulating and listening to their problems as well as being listened to was of great help for breaking creative stalemates.

Extending beyond the work done in this thesis I would like to give thanks to the people at my former workplace Anoto. I would like thank Anders Sjögren for giving me the opportunity to work there and the thrust you had for me and my work. To all my coworkers, especially Håkan Eriksson, Frederik Bjorne, Alexander Israelsson and Kilian Gülich which I've worked closely with, thank you for stimulating and fun conversations. An extra, special thanks to Kilian Gülich for recommending me to Anoto in the first place. My time there is of great importance to me and if it wasn't for him, it wouldn't have happened.

On a more personal note I would like to thank the last year's lunch colleagues, Bella Gleisner, Elin Törnqvist, Lovisa Nilsson and Christina Rönngren to name a few that haven't already been mentioned. Thank you for the food, the ocean view and for keeping me sane. The same goes for all people, new and old, that have been a part of my life these last 6 year, it's been a pleasure.

Contents

1

Introduction

This thesis will deal with the problem of control of stochastic system under the added uncertainty of coarsely quantized measurements. The basic problem of control of stochastic systems is well studied and for simple systems several useful results exist such as Kalman filters and LQG control. However, the introduction of quantised measurement invalidates these results and introduces new effects that need to be accounted for, the main effect being that the state uncertainty now depends on the control input of the system.

To understand this effect, consider a quantised measurement. That measurement will bound the signal to lie within a certain interval and, without any other information about the system, all possible signals within that interval are equally likely. Now, steer the system hard against one of the interval boundaries and take a new measurement. Based on the knowledge of the past control, the probability of the state to be close to that aimed for boundary is now much higher and likewise much lower for it to be close to the other boundaries. With this information, even if the new measurement is the same as the old, the probability distribution of the state is no longer uniform and has shifted and morphed towards the boundary. The choice of control input therefore clearly affects the state probability distribution and this effect will have to be considered when designing a controller.

The reason for having coarse measurement might be purely practical, stemming from limitations in hardware, but can also be an intentional design choice. By dividing the range of possible measurements in fewer intervals, less information is needed to encode the data but also with larger distances between the boundaries the measurement is expected to change less frequently. This reduces the number of times a measurement has to be communicated to the controller and all this combined means that less information needs to be passed over the communication channel. For system where this communication itself is expensive, for example when it's done over a large shared network, a coarse quantisation approach might therefore

be a good choice. Because of this, to be able to improve the control performance in these low informations settings is of great interest and the goal of this thesis.

In trying to achieve better performance, the approach chosen here consists of two parts, the first being improving the state estimation. This is the problem of identifying the probability distribution of the state based on all available knowledge. A method known as a particle filter will be used and the goal will be to design a robust and high performing filter for systems with quantised measurements. The second part is the problem of optimal control, based on the given estimate, in other words, how should the control be chosen? This is done by simulating the system forward in time and from that decide the optimal control. Only methods that fully utilise the special information set provided by the particle filter will be considered but due to the expense of forward simulation, a reduced horizon approach will be used. How far into the future the system will be simulated will be reduced and the system performance for the remaining time until the end will be replaced by some approximation. The goal of this second part will be to design and develop a couple of these methods and evaluate them based on how their different control behaviours affect performance.

The layout of the thesis follows the same approach with it being sectioned into two, largely separate, parts. The first part is dedicated towards the estimation problem and the second is covering the optimal control. Both parts have the same layout with the first chapter introducing the necessary background and theory, the second dealing with the design and development of the methods used and the third dealing with the evaluation of said methods. Before that however, this introductory chapter will be ended by giving short general overviews of the two problem areas as well as defining a system for benchmarking purposes.

1.1 The Estimation Problem

The problem of estimation has its basis in a model of the system that captures the stochastic nature of the process. This model is then used to generate methods for translating a series of measurements into a probability distribution of the state. A general state space model has the following form

$$\begin{aligned} \boldsymbol{x}_{k+1} &= f(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\nu}_k) \\ \boldsymbol{z}_k &= h(\boldsymbol{x}_k, \boldsymbol{e}_k) \end{aligned} \tag{1.1}$$

where $\boldsymbol{x}_k$ is the stochastic state at time step k, $\boldsymbol{u}_k$ is the control signal at time step k, $\boldsymbol{z}_k$ is the available measurement, $\boldsymbol{\nu}_k$ and $\boldsymbol{e}_k$ are independent white noise sequences and f and h is the dynamics and measurement functions respectively. In the case where no measurement noise is present, $\boldsymbol{e}_k = \boldsymbol{0}$, and the measurement function h is

invertible the problem is said to be fully observed. This is because given a measurement, z_k, the state can be determined without uncertainty $x_k = h^{-1}(z_k)$. When this is not the case and either measurement noise is present or h is not invertible the system is said to be partially observed and the estimation problem is then much harder. If all functions are linear the solution is the well known Kalman filter but for non-linear systems there is no general solution. A popular approach for handling non-linear systems is the extended Kalman filter which linearises the functions around a an estimated trajectory, resulting in a local linear approximation for which a regular Kalman filter can be applied.

The systems considered in this thesis can all be written on the state space form above. The quantised measurements translate to a h which is a staircase function, meaning it is constant apart from discontinuities at the interval boundaries. The function h is clearly not invertible, making the system partially observed, and it is also not linear, disqualifying the use of a standard Kalman Filter. However, due to the fact that h is a stair case and thereby having derivative zero almost everywhere, the extended Kalman filter can not be used either. The linearisation of h would simply be constant, resulting in the linearised system not having any measurement information in it at all. An estimation methods that doesn't rely on a linearisation of h is therefore required.

A popular, fully non-linear estimation method of late is the so called particle filter. The reason for its popularity is its simplicity and ability to handle almost arbitrary problems. It is a sampled based approach that represents the state distribution as a collection of random samples, also known as particles. After initialisation, these particles are simply simulated forward in time according to the given system model and the resulting distribution of particles gives an approximate probability distribution at each time-step. The basic concept isn't much more difficult than that although for a proper implementation, a few more details, see Chapter 2, are needed. This simplicity and lack of linearisation makes the particle filter well suited for the problem of quantised measurement studied here.

1.2 Optimal Stochastic Control

The basis of all control is the problem of choosing a control signal to achieve one's goal. In optimal control that goal is formalised in the form of a cost function, a function that gives large values for undesired process behaviour and small values for the desired ones. The optimal control problem then becomes how to choose control signal such to minimise the cost function. In the realm of stochastic system the only difference is that the cost function needs to be a function of stochastic

variables. A fairly general and popular form of cost function is

$$J = \mathrm{E}\left[\sum_{k=0}^{N} \mathscr{L}_k(\boldsymbol{x}_k, \boldsymbol{u}_k)\right] \tag{1.2}$$

where E is the expectation operator and $\mathscr{L}_k$ some function representing the immediate cost at time-step k. There is of course even more general forms of cost functions but this form will cover a vast amount of other problems and can be solved using the well known concept of dynamic programming [Bellman, 1957].

It is possible to make statements about the optimal control based on the effect the control has on the system, in this case mainly focusing on the effect on the state uncertainty. If the uncertainty of the state, the state variance, is independent of the control signal, the separation principle is said to hold. This means that the resulting optimal control will not depend on the state variance at all, it will only depend on the state estimate given by the distribution mean. The separation principle gets its name from this fact, since the optimal control problem then can be separated into two parts, estimation and control.

As mentioned earlier, for a system with quantised measurement the control will affect the shape and therefore the variance of the state distribution. Such systems are said to have dual effect. Dual effect means that the state variance depends on the choice of control and as a result will the separation principle not apply for systems with dual effect. An optimal controller will for such a system have to balance actions that improve the estimate, by reducing the state variance, against actions that steer the system toward the desired state. This is called dual control because of this need to balance the effect the control has on both the state variance and point estimate.

The controller resulting from the dynamic programming will be a dual controller, it will implicitly take the dual effect into consideration. However, due to the difficulties of solving dynamic programming problems, a number of more or less heuristic approaches have been introduced throughout the years with the goal of introducing probing behaviour to the control. Probing is a control action that temporary steers the system away from the desired state in order to improve the estimate and reduce the variance in order to achieve a long term gain. The goal of this thesis is to design methods of dual control that utilise the full probability distribution estimate from the particle filter. All methods will be based on dynamic programming so probing behaviour will be included implicitly. However, due to the aforementioned computational difficulty, simplifications of the full dynamic programming problem are needed.

1.3 Process Definition and Discretisation

For benchmark purposes, a case study will be conducted and the examined process of that study will here be defined. Notation will be established and key properties of the system will be presented for later use in the thesis. Readers unfamiliar with the concepts here are referred to other introductory literature on the subject, for instance [Glad and Ljung, 2000].

The examined process will be the model of a DC servo given in Figure 1.1. It includes two additive white noise terms and models load disturbances as integrated white noise on the input.

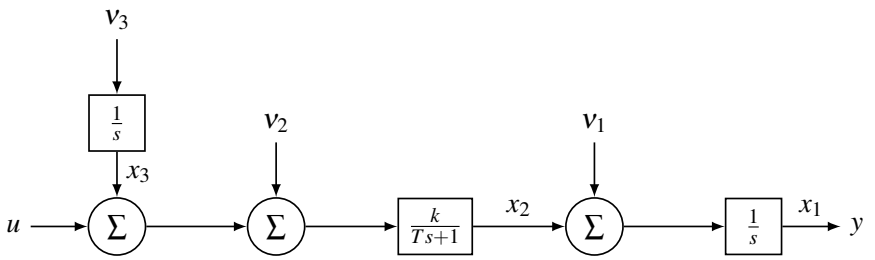

Figure 1.1 DC servo model used for case study.

The model can be summarized in an ordinary linear time invariant state-space model

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u + \boldsymbol{N}\boldsymbol{\nu} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} \end{aligned} \tag{1.3}$$

with the following matrix and vector definitions. For notational convenience we set $\tau = \frac{1}{T}$ and $\mu = \frac{k}{T}$.

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \boldsymbol{\nu} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\tau & \mu \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 \\ \mu \\ 0 \end{bmatrix},$$

$$\boldsymbol{N} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{1.4}$$

The white noise input, $\boldsymbol{\nu}$, has power spectral density $\boldsymbol{R}$ and zero mean.

The system given by (1.3) is observable but not controllable. The third state, d, is not controllable which should be intuitively clear since it represents an uncontrolled

disturbance. This will cause some problem in the future when designing controllers and for that reason we split the system into two parts: one modelling the motor response and the other modelling the load disturbance.

$$\begin{aligned} \dot{\boldsymbol{x}}^{\boldsymbol{m}} &= \boldsymbol{A}^{\boldsymbol{m}}\boldsymbol{x}^{\boldsymbol{m}} + \boldsymbol{B}^{\boldsymbol{m}}u + \boldsymbol{B}^{\boldsymbol{m}}\boldsymbol{x}^{\boldsymbol{d}} + \boldsymbol{N}^{\boldsymbol{m}}\boldsymbol{\nu}^{\boldsymbol{m}} \\ \dot{\boldsymbol{x}}^{\boldsymbol{d}} &= \boldsymbol{N}^{\boldsymbol{d}}\boldsymbol{\nu}^{\boldsymbol{d}} \\ \boldsymbol{y} &= \boldsymbol{C}^{\boldsymbol{m}}\boldsymbol{x}^{\boldsymbol{m}} \end{aligned} \tag{1.5}$$

Here $\boldsymbol{x}^{\boldsymbol{m}} = [x_1\ x_2]^T$, $\boldsymbol{\nu}^{\boldsymbol{m}} = [\nu_1\ \nu_2]^T$, $\boldsymbol{x}^{\boldsymbol{d}} = [x_3]$ and $\boldsymbol{\nu}^{\boldsymbol{d}} = [\nu_3]$. The matrices are of dimensions such that the multiplications are defined and they satisfy:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}^{\boldsymbol{m}} & \boldsymbol{B}^{\boldsymbol{m}} \\ \boldsymbol{0} & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}^{\boldsymbol{m}} \\ 0 \end{bmatrix}, \quad \boldsymbol{N} = \begin{bmatrix} \boldsymbol{N}^{\boldsymbol{m}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{N}^{\boldsymbol{d}} \end{bmatrix}, \tag{1.6}$$

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}^{\boldsymbol{m}} & 0 \end{bmatrix}$$

where $\boldsymbol{0}$ is the zero matrix of corresponding size. The system given by $(\boldsymbol{A}^{\boldsymbol{m}}, \boldsymbol{B}^{\boldsymbol{m}}, \boldsymbol{C}^{\boldsymbol{m}})$ is now both controllable and observable and the disturbance $\boldsymbol{\nu}^{\boldsymbol{d}}$ can simply be seen as an input to this reduced system.

The performance metric for the continuous case is on the form

$$V = \mathrm{E}\left[\int_0^{t_f} \boldsymbol{x}^T\boldsymbol{Q}^x\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{Q}^u\boldsymbol{u}dt\right] \tag{1.7}$$

where t_f is the length of time the system should be controlled for and $\boldsymbol{Q}^x$ and $\boldsymbol{Q}^u$ are chosen weighting matrices that are positive semi-definite and positive definite respectively. Note that because of the uncontrollability of the third state, x_3, no weight can be put on it. An equivalent cost can therefore be formed by only considering the $\boldsymbol{x}^{\boldsymbol{m}}$ states.

$$V = \mathrm{E}\left[\int_0^{t_f} (\boldsymbol{x}^{\boldsymbol{m}})^T\boldsymbol{Q}^{xm}\boldsymbol{x}^{\boldsymbol{m}} + \boldsymbol{u}^T\boldsymbol{Q}^u\boldsymbol{u}dt\right] \tag{1.8}$$

No distinction will be made between these two forms and for simplicity $\boldsymbol{x}$ and $\boldsymbol{Q}^x$ will be used instead of $\boldsymbol{x}^{\boldsymbol{m}}$ and $\boldsymbol{Q}^{xm}$ as long as no confusion arises. Also and without concerns for details, we will let t_f be infinity and it will be for this infinite case or, when not applicable, extremely long time horizons the focus of this report will lie.

Discretisation

Since all controllers and observers discussed in this thesis will be of discrete-time type, the continuous model presented above will be discretised. The details of this

will be left to the appendix and other suitable literature on the subject, for example [Åström and Wittenmark, 2011]. Only the end results will be presented here to establish the notation and some key properties. No weight will be put on potential discretisation problems such as aliasing and inter-sample behaviour. The sample time will be assumed small enough for this not to be a problem.

Let t_k denote the time at sampling instance k such that $t_k + h_k = t_{k+1}$, h_k is here the sample time at time step k. The state at time t_k can then written as $\boldsymbol{x_k}$ and analogously the control signal and output can be written as u_k and y_k respectively. In the future, variables subscripted with k will denote the discrete time variants. When need arises other variable names than k might be used but it will then be made clear from context that it's a time index.

By integrating between t_k and t_{k+1} with a zero-order hold on the control signal the following discretisation of (1.3) can be achieved, see Appendix A.1 for details.

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k u_k + \boldsymbol{\nu}_k \\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k \end{aligned} \tag{1.9}$$

$\boldsymbol{\nu}_k = [\nu_k^1, \nu_k^2, \nu_k^3]$ is here a white noise sequence with power spectrum $\widehat{\boldsymbol{R}}_k$. $\widehat{\boldsymbol{R}}_k$ will be positive definite if we restrict $\boldsymbol{N}$ and $\boldsymbol{R}$ to have full rank, see Appendix A.2. These restrictions are natural since they ensure that the noise inputs are independent. For details on the numerical calculation of these system matrices see Appendix A.4.

It's clear from Appendix A.1 that the system matrices only depend on h_k, meaning that if periodic sampling is deployed the time dependent system above will be time independent. When that's the case the subscripts will be dropped from the system matrices as such:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{\Gamma} u_k + \boldsymbol{\nu}_k \\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k \end{aligned} \tag{1.10}$$

In this case the white noise power spectrum is also constant and will be denoted $\widehat{\boldsymbol{R}}$.

The controllability problem remains for the discretised problem but fortunately, similar to the continuous system, the system matrices can be divided into a model and load disturbance part, see Appendix A.5. From that, one can also see that both $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi^m}$ are upper triangular.

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi^m} & \boldsymbol{\Gamma^m} \\ \boldsymbol{0} & 1 \end{bmatrix}, \quad \boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\Gamma^m} \\ 0 \end{bmatrix} \tag{1.11}$$

Resulting in:

$$\begin{aligned} \boldsymbol{x}_{k+1}^{\boldsymbol{m}} &= \boldsymbol{\Phi}_k^{\boldsymbol{m}} \boldsymbol{x}_k^{\boldsymbol{m}} + \boldsymbol{\Gamma}_k^{\boldsymbol{m}} u_k + \boldsymbol{\Gamma}_k^{\boldsymbol{m}} \boldsymbol{x}_k^{\boldsymbol{d}} + \boldsymbol{\nu}_k^{\boldsymbol{m}} \\ \boldsymbol{x}_{k+1}^{\boldsymbol{d}} &= \boldsymbol{x}_k^{\boldsymbol{d}} + \boldsymbol{\nu}_k^{\boldsymbol{d}} \\ \boldsymbol{y}_k &= \boldsymbol{C}^{\boldsymbol{m}} \boldsymbol{x}_k^{\boldsymbol{m}} \end{aligned} \tag{1.12}$$

The cost function can be discretised according to Appendix A.3, resulting in

$$J = \mathrm{E}\left[\sum_{k=0}^{N} \boldsymbol{x}_k^T \boldsymbol{Q}_k^x \boldsymbol{x}_k + 2\boldsymbol{x}_k^T \boldsymbol{Q}_k^{xu} \boldsymbol{u}_k + \boldsymbol{u}_k^T \boldsymbol{Q}_k^u \boldsymbol{u}_k\right] \tag{1.13}$$

Just like in the continuous case we will let N be infinity without to much concern for details and it's implied that the uncontrollable state is ignored in the cost function.

Discretised Measurement

The system presented here, both the continuous and discrete, are linear and therefore of little interest in this thesis. They will therefore be complemented with a non-linear quantiser as a measurement function. The quantiser will be on the form:

$$\boldsymbol{z} = round\left(\frac{\boldsymbol{y}}{\Delta q}\right)\Delta q \tag{1.14}$$

In other words, the elements of $\boldsymbol{y}$ are rounded to nearest multiple of the quantizer interval, Δq. The resulting quantised systems then becomes

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u + \boldsymbol{N}\boldsymbol{\nu} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} \\ \boldsymbol{z} &= round\left(\frac{\boldsymbol{y}}{\Delta q}\right)\Delta q \end{aligned} \tag{1.15}$$

in the continuous case and

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k u_k + \boldsymbol{\nu}_k \\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k \\ \boldsymbol{z}_k &= round\left(\frac{\boldsymbol{y}_k}{\Delta q}\right)\Delta q \end{aligned} \tag{1.16}$$

in the discrete.

Simulation

When doing performance comparisons the above model was simulated using Simulink. A model was made of the continuous process above which was used for all subsequent experiments. Different discrete estimators and controllers were

Table 1.1 Default Process Parameters

$k = 1$ $T = 1$	$h_k = 0.05$ $\Delta q = 20$	$\boldsymbol{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\boldsymbol{Q}^x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \boldsymbol{Q}^u = 0.1$

applied in open and closed loop situations and the relevant data was gathered for comparison. More information on the test cases and performance metrics used will be given alongside the corresponding presentation and discussion of the results. If nothing else is stated, the process parameters in Table 1.1 will be used.

Part I

Estimation

2

Bayesian Inference and Particle Filters

This chapter will cover the theory behind the filters used to identify the system state. The goal is to develop theory that allows for identification of the complete probability distribution of the system state and computational methods for approximating it. This stands in contrast to traditional methods where the goal is to find an optimal estimator of some sort, a filter that delivers a state estimation that is optimal according to some criterion such as maximum a posterior probability, MAP, or minimal mean square error, MMSE. There isn't necessarily a clear line between these methods since the theory presented also serves as a basis for optimal estimator design. However, such topics will not be a subject of this chapter and it will solely focus on the complete probability distribution.

The base for this will be the theory of Bayesian inference which first will be presented in short. This is the theory that allows for the probabilities to be propagated through the system and identified at the other end. Only in rare cases have these equations an analytical solution, the famous Kalman filter being one. Because of this, the rest of the chapter will be spent developing theory that allows for approximate solutions to be calculated, the theory of sequential Monte Carlo (SMC) methods, also known as particle filters (PF). A rather complete introduction of SMC methods can be found in [Doucet et al., 2001] while most of the work presented here are based on foundational work of [Gordon et al., 1993] and the general particle filter introduction in [Schön, 2010]. A further development of the particle filter, the auxiliary particle filter, first introduced by [Pitt and Shephard, 1999] and independently discovered during the development of this thesis, will also be presented. No original contributions to the theory will be made other than presenting the above methods in a common framework of intermediate distribution reweighting, the goal of this reweighting is to improve numerical performance of the methods. In total four PF:s will be presented, each an extension of the one before, and later in Chap-

ter 4, will their performance be compared when they are applied on the process presented in Figure 1.1.

2.1 Bayesian Inference for Discrete Dynamic Systems

A brief introduction to Bayesian inference will here be given with the goal of providing a recursive update formulation for the state probability for a discrete time system. In other words: Given the probability distribution of $\boldsymbol{x}_k$, what will the probability distribution of $\boldsymbol{x}_{k+1}$ be?

Bayesian Inference and Prediction

Bayesian inference is in words the method to determine the distribution of an examined variable, given some known prior distribution and observations. For a random variable, Θ and some measurements X that depend on Θ, Bayes' rule can be used to formulate posterior distribution of Θ. Here Θ is also allowed to depend on a third parameter ρ.

$$p(\Theta|X,\rho) = \frac{p(X|\Theta,\rho)p(\Theta|\rho)}{p(X|\rho)} = \frac{p(X|\Theta)p(\Theta|\rho)}{p(X|\rho)} \propto p(X|\Theta)p(\Theta|\rho) \tag{2.1}$$

Θ is here the *examined variable*, the one information wanted for. X are *observations* whose outcome depends on Θ. $p(\Theta|\rho)$ is then the *prior distribution*, i.e. the distributions before any observations X were made. $p(X|\Theta)$ is the *likelihood* of the variable Θ giving the observed measurement X. $p(X|\rho)$ is the *marginal likelihood*, the total likelihood of the outcome X, regardless of Θ.

These distributions can also be used to make predictions of future observations. Let $\widetilde{X}$ be some random variable dependent on Θ. A *prior predictive distribution* can then be formed based on none of the current observation, only the prior distribution

$$p(\widetilde{X}|\rho) = \int p(\widetilde{X}|\Theta)p(\Theta|\rho)d\Theta \tag{2.2}$$

A better *posterior prediction distribution* can be formed by including all current observations.

$$p(\widetilde{X}|X,\rho) = \int p(\widetilde{X}|\Theta)p(\Theta|X,\rho)d\Theta \tag{2.3}$$

State Probability Update

To put this into context and to derive the recursive probability update, consider the fairly general non-linear discrete time system given on the state-space form:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= f(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{\nu}_k \\ \boldsymbol{z}_k &= h(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{e}_k \end{aligned} \tag{2.4}$$

An equivalent stochastic formulation is:

$$\begin{aligned} \boldsymbol{x}_{k+1} &\sim p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k,\boldsymbol{u}_k) = p_{\boldsymbol{\nu}_k}(\boldsymbol{x}_{k+1} - f(\boldsymbol{x}_k,\boldsymbol{u}_k)) \\ \boldsymbol{z}_k &\sim p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k) = p_{\boldsymbol{e}_k}(\boldsymbol{z}_k - h(\boldsymbol{x}_k,\boldsymbol{u}_k)) \end{aligned} \tag{2.5}$$

With this the parameters for time-step k can now start to be identified.

The problem is to infer the distribution of $\boldsymbol{x}_k$ given all previous knowledge, making it the examined variable. The observation should of course be $\boldsymbol{z}_k$ since it depends directly on $\boldsymbol{x}_k$. However, $\boldsymbol{x}_k$ also depends on the history of the system so it also needs to be included some way. In the Bayesian inference equations, $\boldsymbol{\Theta}$ were allowed to depend on one parameter, $\boldsymbol{\rho}$. In this case let's gather all prior knowledge of the system, i.e. control signal and measurements, in a parameter, $D_k = \{\boldsymbol{z}_k, \boldsymbol{u}_k, \boldsymbol{z}_{k-1}, \boldsymbol{u}_{k-1} \dots\}$, and use D_{k-1} as that parameter.

By correctly identifying all pieces of (2.1) and noting that $\boldsymbol{z}_k$ also depends on the known $\boldsymbol{u}_k$

$$p(\boldsymbol{x}_k|D_k) = p(\boldsymbol{x}_k|\boldsymbol{z}_k,\boldsymbol{u}_k,D_{k-1}) = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k)p(\boldsymbol{x}_k|D_{k-1})}{p(\boldsymbol{z}_k|\boldsymbol{u}_k,D_{k-1})} \tag{2.6}$$

It can now be seen that the *posterior distribution* is the sought after $p(\boldsymbol{x}_k|D_k)$ and the distribution of the measurement from (2.5), has become the *likelihood*. It can also be seen that the *prior distribution*, $p(\boldsymbol{x}_k|D_{k-1})$, is the *posterior predictive distribution* of the state $\boldsymbol{x}_k$ at time-step $k-1$.

$$\begin{aligned} p(\boldsymbol{x}_k|D_{k-1}) &= \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},D_{k-1})p(\boldsymbol{x}_{k-1}|D_{k-1})d\boldsymbol{x}_{k-1} \\ &= \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1})p(\boldsymbol{x}_{k-1}|D_{k-1})d\boldsymbol{x}_{k-1} \end{aligned} \tag{2.7}$$

Left is only the *marginal likelihood*, $p(\boldsymbol{z}_k|\boldsymbol{u}_k,D_{k-1})$, i.e. the likelihood that $\boldsymbol{z}_k$ is measured at time-step k given past measurement and current control signal.

By assuming $p(\boldsymbol{x}_{k-1}|D_{k-1})$ is known, it can now be summarised into the complete recursive update formulation.

$$\begin{aligned} p(\boldsymbol{x}_k|D_{k-1}) &= \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1})p(\boldsymbol{x}_{k-1}|D_{k-1})d\boldsymbol{x}_{k-1} \\ p(\boldsymbol{x}_k|D_k) &= \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k)p(\boldsymbol{x}_k|D_{k-1})}{p(\boldsymbol{z}_k|\boldsymbol{u}_k,D_{k-1})} \\ p(\boldsymbol{z}_k|\boldsymbol{u}_k,D_{k-1}) &= \int p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k)p(\boldsymbol{x}_k|D_{k-1})d\boldsymbol{x}_k \end{aligned} \tag{2.8}$$

These are the equations that serve as the basis for all the following work in this chapter. They, together with the stochastic formulation of the dynamic system (2.5),

govern how the state distribution propagates through time. But, like said before, only in special cases does these equations have an analytical solution. The case where f and h are linear functions and $\boldsymbol{\nu}_k$ and $\boldsymbol{e}_k$ are Gaussian white-noise results in the famous Kalman-Filter, see for example [Schön, 2010]. When this isn't the case, numerical approximations needs to be used, this is the focus of the following sections.

2.2 Sampling/Importance Resampling

In order to derive suitable numerical methods to deal with (2.8) a couple of key concepts will here be established. They will cover ways to represent and sample from difficult distributions. The problem of how to numerically represent the distributions of (2.8) will be handled by sampling/importance resampling (SIR), or as it is also known, weighted bootstrap sampling. It is a Monte Carlo method based on importance sampling, as the name implies. An introduction to the concepts covered here can also be found in [Smith and Gelfand, 1992].

Perfect Monte Carlo Sampling

A naive approach for a numerical representation of some probability density function (PDF) is to simply evaluate the PDF at some predetermined grid points. A problem with this approach is the selection of the grid points, where is higher resolution needed and where can the grid spacing be larger? In the case of a dynamic system these locations are not necessary fixed in time, making this problem even harder. Monte Carlo sampling is one way around this problem.

Assume some PDF, p, from which samples can be drawn is given. Monte Carlo sampling is then to numerically approximate p by sampling it N times and forming the approximation as

$$\widehat{p}(x) = \sum_{i=1}^{N} \frac{1}{N} \delta(x - x^i), \quad x^i \sim p(x) \tag{2.9}$$

Superscript i denotes the index of the sample and should not be confused with an exponent of an exponential. The reason for a superscript being used instead of a less ambiguous subscript will be apparent when samples are drawn from the dynamic model's states.

It can be shown that $\widehat{p}$ approximates p in the sense that expected values such as $\mathrm{E}_p[g(x)] \approx \mathrm{E}_{\widehat{p}}[g(x)]$ become better with bigger N [Doucet et al., 2001]. This is what in the future will be meant by $\widehat{p}$ approximates p. An intuitive feeling for this can be

obtained by expanding the expected value.

$$\begin{aligned} \mathrm{E}_p[g(x)] &= \int g(x)p(x)dx \\ &\approx \int g(x)\widehat{p}(x)dx = \sum_{i=1}^{N}\frac{1}{N}\int g(x)\delta(x-x^i)dx = \sum_{i=1}^{N}\frac{1}{N}g(x^i) \end{aligned} \tag{2.10}$$

Approximations of this type clearly replace the integral with a equally weighted sum. It is the location of the sample points that governs what part of the function g that will be weighted higher, i.e. since x^i are sampled from p there will be more x^i located in the region where p is large, putting more weight on those values of g.

Importance Sampling

In practice it can be hard to sample from the PDF p directly and the concept of Monte Carlo sampling can then be expanded further. By drawing samples from some other *proposal distribution* p_π whose support contains the support of p, a weighted approximation can be formed as

$$\widehat{p}(x) = \sum_{i=1}^{N}\omega^i\delta(x-x^i), \quad x^i \sim p_\pi(x) \tag{2.11}$$

The constraints on the support of the proposal distribution should come naturally since otherwise there will be parts of p, x^i never will be able to represent. For future reference, let $\widehat{p}$ mean an approximation of p on this form.

The problem is now one of determining the weights ω^i, this is what importance sampling is for. The idea is easily demonstrated by forming the integral of the expected value that is approximated and inserting p_π into it.

$$\begin{aligned} \mathrm{E}_p[g(x)] &= \int g(x)p(x)dx = \int g(x)\frac{p_\pi(x)}{p_\pi(x)}p(x)dx \\ &= \int g(x)\frac{p(x)}{p_\pi(x)}p_\pi(x)dx = \int g(x)w(x)p_\pi(x)dx \end{aligned} \tag{2.12}$$

Where $w(x) = \frac{p(x)}{p_\pi(x)}$. By inserting a equally weighted Monte Carlo sampled approximation of p_π

$$\mathrm{E}_p[g(x)] \approx \int g(x)w(x)\sum_{i=1}^{N}\frac{1}{N}\delta(x-x^i)dx = \sum_{i=1}^{N}\frac{w(x^i)}{N}g(x^i), \quad x^i \sim p_\pi(x) \tag{2.13}$$

one can then easily identify $\omega^i = \frac{w(x^i)}{N}$, resulting in

$$\widehat{p}(x) = \sum_{i=1}^{N}\frac{w(x^i)}{N}\delta(x-x^i), \quad x^i \sim p_\pi(x) \tag{2.14}$$

The reason for the support constrains can now also be seen in a different way. Without that constraint, the insertion of $w(x)$ would result in division by zero. Although this is the only real constraint on p_π, it can be shown that the better p_π follows p, the better the approximation will be [Schön, 2010].

By following very similar procedures even further relaxations can be made. First assume that not only is it not possible to sample from p, it can not be fully evaluated either, it is only known to some proportional constant $p(x) \propto f(x)$. By normalising f and inserting an approximation of p_π the expected value becomes

$$\begin{aligned} \mathrm{E}_p[g(x)] &= \frac{\int g(x)f(x)dx}{\int f(x)dx} = \frac{\int g(x)\frac{f(x)dx}{p_\pi(x)}p_\pi(x)}{\int \frac{f(x)}{p_\pi(x)}p_\pi(x)dx} \\ &= \frac{\int g(x)w(x)p_\pi(x)dx}{\int w(x)p_\pi(x)dx} \approx \frac{\sum_{i=1}^N \frac{1}{N}\int g(x)w(x)\delta(x-x^i)dx}{\sum_{i=1}^N \frac{1}{N}\int w(x)\delta(x-x^i)dx} \\ &= \frac{\sum_{i=1}^N \frac{1}{N}g(x^i)w(x^i)}{\sum_{i=1}^N \frac{1}{N}w(x^i)} = \sum_{i=1}^N \frac{w(x^i)}{\sum_{i=1}^N w(x^i)}g(x^i), \quad x^i \sim p_\pi(x) \end{aligned} \tag{2.15}$$

where w now instead is $w(x) = \frac{f(x)}{p_\pi(x)}$. It is clear that the approximation remains the same except p is replaced with f and the weights need to be normalised such that $\sum_{i=1}^N \omega^i = 1$.

$$\widehat{p}(x) = \sum_{i=1}^N \omega^i \delta(x-x^i), \quad \omega^i = \frac{w(x^i)}{\sum_{i=1}^N w(x^i)}, \quad x^i \sim p_\pi(x) \tag{2.16}$$

Up until now, it has been assumed that an equally weighted approximation of p_π is used but nothing restricting the use of a weighted one with weights $\widetilde{\omega}^i$ and samples x^i taken from yet another PDF p_Π.

$$\begin{aligned} \mathrm{E}_p[g(x)] &= \frac{\int g(x)w(x)p_\pi(x)dx}{\int w(x)p_\pi(x)dx} \approx \frac{\sum_{i=1}^N \widetilde{\omega}^i \int g(x)w(x)\delta(x-x^i)dx}{\sum_{i=1}^N \widetilde{\omega}^i \int w(x)\delta(x-x^i)dx} \\ &= \frac{\sum_{i=1}^N \widetilde{\omega}^i g(x^i)w(x^i)}{\sum_{i=1}^N \widetilde{\omega}^i w(x^i)} = \sum_{i=1}^N \frac{\widetilde{\omega}^i w(x^i)}{\sum_{i=1}^N \widetilde{\omega}^i w(x^i)}g(x^i), \quad x^i \sim p_\Pi(x) \\ &\Longrightarrow \\ \widehat{p}(x) &= \sum_{i=1}^N \omega^i \delta(x-x^i), \quad \omega^i = \frac{\widetilde{\omega}^i w(x^i)}{\sum_{i=1}^N \widetilde{\omega}^i w(x^i)}, \quad x^i \sim p_\Pi(x) \end{aligned} \tag{2.17}$$

This is the most general form and will be used to express the future state distributions expressed in the past distributions.

Algorithm 2.1: Sampling/Importance Resampling (SIR)

Given some approximation of the proposal distribution p_π with samples x^i and weights $\widetilde{\omega}^i$, approximate $p(x) \propto f(x)$

Calculate weights: $$\widehat{\omega}^i = \frac{f(x^i)}{p_\pi(x)}\widetilde{\omega}^i$$

Normalize weights: $$\omega^i = \frac{\widehat{\omega}^i}{\sum_{i=1}^N \widehat{\omega}^i}$$

Resample:

Draw N samples, $\widetilde{x}^i$ from $P(x = x^i) = \omega^i$

Form approximation: $\widehat{p}(x) = \sum_{i=1}^{N} \frac{1}{N}\delta(x - \widetilde{x}^i)$

Sampling/Importance Resampling

Importance sampling is not without problem though, especially when the filter is used recursively on itself, using the approximation $\widehat{p}$ as p_π to get an approximation of some other PDF and redoing this over and over. In such cases the weights might degenerate where most of the weights goes towards zero, effectively loosing resolution of the approximation. It is therefore very inefficient to have samples with very small weights and for this reason SIR is used. It is a method that equalizes the weights of the approximation.

The SIR procedure starts with a normal importance sampling step but then performs a resampling of the generated approximation. During the resampling N new samples are simply drawn from the approximation $\widehat{p}$ and assigned equal weights $\frac{1}{N}$. This is a simple Monte Carlo sampling to approximate $\widehat{p}$ and the result will then also approximate the original p, since $\widehat{p}$ is a discrete PDF and sampling from it is done by simply drawing x such that $P(x = x^i) = \omega^i$. The procedure is summarised in Algorithm 2.1. Once again it is worth noting that the closer to p the distribution from where x^i are drawn from, the better the algorithm will perform.

How to draw samples from the discrete distribution in the resampling step is something that haven't yet been touch upon. There are a number of different strategies for doing so but it has been shown, [Douc and Cappé, 2005], that in respect to PF performance, the resampling strategy has little impact. Because of that, only one method will be presented here and it is the systematic sampling presented in Algorithm 2.2. It is stated for completeness and the interested reader is referred elsewhere. It is chosen mainly for simplicity and speed, not necessarily for its accuracy in producing truly random independent samples. In the future, it will only be

Algorithm 2.2: Systematic Sampling

Given a discrete distribution $P(x = x_i) = \omega_i,\ 1 \leq i \leq N$.

Sample uniform distribution: $\widetilde{u} \sim \mathscr{U}(0,1)$

Form the ordered numbers: $u_i = \dfrac{(i-1)+\widetilde{u}}{N},\ 1 \leq i \leq N$

Form accumulative sum: $s_i = \sum_{k=1}^{i} \omega_k,\ s_0 = 0$

Form sample counts: $n_i = \text{number of } u_i \in (s_{i-1}, s_i]$

The new samples can now be generated by taking n_i copies of x_i.

this algorithm that is used when new samples are drawn in Algorithm 2.1. For that reason the details of the resample step will no longer be specified and only referred to as *resampling*.

2.3 SIR Particle Filter

With the concepts previously introduced, the first and simplest particle filter can be introduced, the SIR particle filter (SIR-PF). Recall the stochastic dynamic system (2.5) and the recursive probability update (2.8), both restated here for ease of reference.

Dynamic Model:

$$\begin{aligned} \boldsymbol{x}_{k+1} &\sim p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k, \boldsymbol{u}_k) = p_{\boldsymbol{\nu}_k}(\boldsymbol{x}_{k+1} - f(\boldsymbol{x}_k, \boldsymbol{u}_k)) \\ \boldsymbol{z}_k &\sim p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{u}_k) = p_{\boldsymbol{e}_k}(\boldsymbol{z}_k - h(\boldsymbol{x}_k, \boldsymbol{u}_k)) \end{aligned} \tag{2.18}$$

Recursive Probability Update:

$$p(\boldsymbol{x}_k|D_{k-1}) = \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) p(\boldsymbol{x}_{k-1}|D_{k-1}) d\boldsymbol{x}_{k-1} \qquad \textit{Prediction}$$

$$p(\boldsymbol{x}_k|D_k) \propto p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{u}_k) p(\boldsymbol{x}_k|D_{k-1}) \qquad \textit{Measurement Update}$$

With the base assumptions that $p_{\boldsymbol{e}_k}$ can be evaluated and that $p_{\boldsymbol{\nu}_k}$ and $p_{\boldsymbol{x}_0} = p(\boldsymbol{x}_0|D_0)$ can be both evaluated and sampled from, start by creating an approximation of $p(\boldsymbol{x}_0|D_0) \approx \widehat{p}(\boldsymbol{x}_0|D_0)$. This approximation will recursively be propagated through the recursive update functions, utilising the SIR algorithm in the *measurement update*, in order to form our estimate at time-step k. To derive the details of

this procedure, assume $\widehat{p}(\boldsymbol{x}_{k-1}|D_{k-1})$ is known and insert it into the *prediction* step of the recursive update.

$$\begin{aligned} p(\boldsymbol{x}_k|D_{k-1}) &\approx \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1})\widehat{p}(\boldsymbol{x}_{k-1}|D_{k-1})d\boldsymbol{x}_{k-1} \\ &= \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1})\sum_{i=1}^{N}\omega_{k-1}^i\delta(\boldsymbol{x}_{k-1}-\boldsymbol{x}_{k-1}^i)d\boldsymbol{x}_{k-1} \\ &= \sum_{i=1}^{N}\omega_{k-1}^i p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i,\boldsymbol{u}_{k-1}) \end{aligned} \tag{2.19}$$

The predictive distribution can clearly be approximated as a weighted sum of N other distributions. The easiest way to take N samples from such a sum is to simply take one sample from each term and and an approximation can then be formed as

$$\widehat{p}(\boldsymbol{x}_k|D_{k-1}) = \sum_{i=1}^{N}\omega_{k-1}^i\delta(\boldsymbol{x}_k-\boldsymbol{x}_k^i), \quad \boldsymbol{x}_k^i \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i,\boldsymbol{u}_{k-1}) \tag{2.20}$$

These samples can easily be taken given the assumption the $p_{\boldsymbol{\nu}_k}$ and the dynamic model, (2.18) as

$$\boldsymbol{x}_k^i = f(\boldsymbol{x}_{k-1}^i,\boldsymbol{u}_{k-1}) + \boldsymbol{\nu}_{k-1}^i, \quad \boldsymbol{\nu}_{k-1}^i \sim p_{\boldsymbol{\nu}_{k-1}} \tag{2.21}$$

The name "particle filter" should here be apparent since the samples $\boldsymbol{x}_{k-1}$ act as particles being propagated in the system. In the future, the terms sample and particle will be used interchangeably to represent $\boldsymbol{x}_k$.

By using $\widehat{p}(\boldsymbol{x}_k|D_{k-1})$ as the basis for the SIR algorithm and identifying $w(\boldsymbol{x}_k)$ as

$$w(\boldsymbol{x}_k) = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k)p(\boldsymbol{x}_k|D_{k-1})}{p(\boldsymbol{x}_k|D_{k-1})} = p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k) \tag{2.22}$$

an approximation for $\widehat{p}(\boldsymbol{x}_k|D_k)$ can be obtained since by assumption, $p_{\boldsymbol{e}_k}$ can be evaluated and thereby so can $p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k)$.

These steps can now be summarized in the approximative recursive update functions. Given $\widehat{p}(\boldsymbol{x}_{k-1}|D_{k-1}) = \sum_{i=1}^{N}\omega_{k-1}^i\delta(\boldsymbol{x}_{k-1}-\boldsymbol{x}_{k-1}^i)$

$$\begin{aligned} \widehat{p}(\boldsymbol{x}_k|D_{k-1}) &= \sum_{i=1}^{N}\omega_{k-1}^i\delta(\boldsymbol{x}_k-\boldsymbol{x}_k^i), & \boldsymbol{x}_k^i &\sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i,\boldsymbol{u}_{k-1}) \\ \widehat{p}(\boldsymbol{x}_k|D_k) &= \sum_{i=1}^{N}\omega_k^i\delta(\boldsymbol{x}_k-\boldsymbol{x}_k^i), & \omega_k^i &= \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k^i,\boldsymbol{u}_k)\omega_{k-1}^i}{\sum_{i=1}^{N}p(\boldsymbol{z}_k|\boldsymbol{x}_k^i,\boldsymbol{u}_k)\omega_{k-1}^i} \\ \widehat{p}(\boldsymbol{x}_k|D_k) &\leftarrow \text{Resample } \widehat{p}(\boldsymbol{x}_k|D_k) \end{aligned} \tag{2.23}$$

Compared to the exact update functions, (2.8), the integral of the prediction is replaced by sampling from the proposal distribution while the multiplication of continuous distributions in the measurement update is replaced by re-weighting of the sample weights. The resample step lacks analogue since it was introduced purely to provide better numerical behaviour. It is therefore not necessarily crucial that it is performed every time-step and there are PF variants where the degeneracy of the weights are measured and resampling is only performed when it has reached some predefined limit. For simplicity these strategies will not be employed in this work but are still recommended.

Because of the nature of the resampling step, information is discarded when it is performed. This is intentional but it means that when the approximate distribution should be used for calculation of some expectation value or similar, it is the distribution before resampling that should be used for the best results.

2.4 Auxiliary Particle Filter

The auxiliary particle filter (A-PF) [Pitt and Shephard, 1999] is a variant of the particle filter, designed to improve the resampling step so that it truly is the least important information that is discarded. It accomplishes this by moving the resampling to the next time-step and introduces an extra re-weighting step. This re-weighting is done based on the most recent measurements, thereby using all the information available before discarding any samples. This re-weighting can be seen as introducing a new auxiliary random variable between the prediction and measurement update step, hence the name. To illustrate this, the exact recursive update functions, (2.8), are once again turned to.

Let $q(\boldsymbol{x}_k, \boldsymbol{x}_{k-1})$ be some bounded, non-negative function. Inserting this into the prediction step of the recursive update gives

$$\begin{aligned} p_{\boldsymbol{\chi}_k}(\boldsymbol{x}_k) &\propto \int \frac{p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1})}{q(\boldsymbol{x}_k, \boldsymbol{x}_{k-1})} q(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1}|D_{k-1}) d\boldsymbol{x}_{k-1} \\ p(\boldsymbol{x}_k|D_k) &\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{u}_k) p_{\boldsymbol{\chi}_k}(\boldsymbol{x}_k) \end{aligned} \tag{2.24}$$

In the continuous case no real gain is made but in the sampled case q can be utilised to shape the prediction step to yield better performance. For that reason is it a good idea to base the choice of q on the currently available measurements, $q(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|D_k)$, as well but for convenience it will be left out for now. Two different approaches of how to utilise this q will be presented: the first being the A-PF which focuses on improving the resampling while the second is the general PF which focuses on improving the predictive sampling. For the A-PF it is required that q

is independent of $\boldsymbol{x}_k$ which makes it possible to introduce the auxiliary variable $\boldsymbol{\chi}_{k-1}|D_{k-1}$.

$$\begin{aligned} p(\boldsymbol{\chi}_{k-1}) &\propto q(\boldsymbol{\chi}_{k-1}) p_{\boldsymbol{x}_{k-1}|D_{k-1}}(\boldsymbol{\chi}_{k-1}) \\ p_{\boldsymbol{\chi}_k}(\boldsymbol{x}_k) &\propto \int \frac{p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1})}{q(\boldsymbol{\chi}_{k-1})} p(\boldsymbol{\chi}_{k-1}) d\boldsymbol{\chi}_{k-1} \\ p(\boldsymbol{x}_k|D_k) &\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{u}_k) p_{\boldsymbol{\chi}_k}(\boldsymbol{x}_k) \end{aligned} \tag{2.25}$$

q can clearly be used to almost arbitrary control the distribution of $\boldsymbol{\chi}_{k-1}|D_{k-1}$. When converting these equations to the sampled PF this extra variable requires a SIR step, effectively allowing us to remove the resample step of $\widehat{p}(\boldsymbol{x}_k|D_k)$ resulting in the following update functions.

Given $\widehat{p}(\boldsymbol{x}_{k-1}|D_{k-1}) = \sum_{i=1}^N \omega_{k-1}^i \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^i)$

$$\begin{aligned} &\widehat{p}(\boldsymbol{\chi}_{k-1}) = \sum_{i=1}^N \widetilde{\omega}_{k-1}^i \delta(\boldsymbol{\chi}_{k-1} - \boldsymbol{x}_{k-1}^i), && \widetilde{\omega}_{k-1}^i = \frac{q(\boldsymbol{x}_{k-1}^i)\omega_{k-1}^i}{\sum_{i=1}^N q(\boldsymbol{x}_{k-1}^i)\omega_{k-1}^i} \\ &\widehat{p}(\boldsymbol{\chi}_{k-1}) \leftarrow \text{Resample } \widehat{p}(\boldsymbol{\chi}_{k-1}) \\ &\widehat{p}_{\boldsymbol{\chi}_k}(\boldsymbol{x}_k) = \sum_{i=1}^N \frac{\widetilde{\omega}_{k-1}^i}{q(\boldsymbol{x}_{k-1}^i)} \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^i), && \boldsymbol{x}_k^i \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i, \boldsymbol{u}_{k-1}) \\ &\widehat{p}(\boldsymbol{x}_k|D_k) = \sum_{i=1}^N \omega_k^i \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^i), && \omega_k^i = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k^i, \boldsymbol{u}_k)\widetilde{\omega}_{k-1}^i / q(\boldsymbol{x}_{k-1}^i)}{\sum_{i=1}^N p(\boldsymbol{z}_k|\boldsymbol{x}_k^i, \boldsymbol{u}_k)\widetilde{\omega}_{k-1}^i / q(\boldsymbol{x}_{k-1}^i)} \end{aligned} \tag{2.26}$$

Here the normalisation of the weights for $\widehat{p}_{\boldsymbol{\chi}_k}(\boldsymbol{x}_k)$ have been incorporated into the normalisation of ω_k^i. The same could be made for $\widetilde{\omega}_{k-1}^i$ if a resampling routine that could handle non-normalised weights was used.

From these equations, first note that if not for the resample-step, the introduction of q would have no effect, making it clear how to interpret the function q. The effect it has is that it changes the weights before the resampling, allowing for control of what samples to prioritise. It can in theory be chosen arbitrary but a careless choice can ruin the filter while a good choice can discard outliers and prioritise the most valuable particles.

The key to a good choice of q is, as earlier mentioned, to condition q on the all available measurements $q(\boldsymbol{x}_{k-1}|D_k)$. This is the main advantage against the SIR filter presented before: it only uses the information up until k when resampling $\widehat{p}(\boldsymbol{x}_k|D_k)$, the A-PF uses the information in the next time step, D_{k+1} as well. Note that if q is held constant the basic SIR filter is received. The q proposed for this

thesis will be

$$\begin{aligned} q(\boldsymbol{x}_{k-1}) &= q(\boldsymbol{x}_{k-1}|D_k) = p(\boldsymbol{z}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_k,\boldsymbol{u}_{k-1}) \\ &= \int p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1})d\boldsymbol{x}_k \end{aligned} \tag{2.27}$$

With this q, weight will be added to the particles likely to give the current measurement while unlikely particles are more likely to be discarded during resampling. The integral in question is always, at least in theory, able to be computed given that the system model is known and on the form (2.18). If for some reason exact evaluation isn't possible an approximative evaluation also has the potential to improve performance since all valid choices of q are equivalent in the continuous case. An example of how the choice of q affects performance can be found in Chapter 4.

2.5 General Particle Filter

Previously, the restriction on $q(\boldsymbol{x}_k,\boldsymbol{x}_{k-1})$ was quite relaxed but now assume q is a PDF of $\boldsymbol{x}_k$ conditioned on $\boldsymbol{x}_{k-1}$ from which samples can be drawn. q will have a similar function as for the A-PF but instead of gaining control of the weights before resampling, control over the proposal distribution from where the predictive samples are drawn is achieved.

Assume once again that a sampled approximation $\widehat{p}(\boldsymbol{x}_{k-1}|D_{k-1})$ exists and let it propagate through the prediction step of (2.24).

$$\begin{aligned} p_{\boldsymbol{\chi}_k}(\boldsymbol{x}_k) &\propto \int \frac{p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1})}{q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})} q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1}|D_{k-1})d\boldsymbol{x}_{k-1} \\ &\approx \sum_{i=1}^{N} \frac{p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i,\boldsymbol{u}_{k-1})\omega_{k-1}^i}{q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i)} q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i) \end{aligned} \tag{2.28}$$

Once again the result is a sum of distributions and one new sample will be taken from each term. This time however, importance sampling will be employed for each term. One sample will be drawn from each $q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i)$ and given a weight, resulting in

$$p_{\boldsymbol{\chi}_k}(\boldsymbol{x}_k) \propto \sum_{i=1}^{N} \frac{p(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i,\boldsymbol{u}_{k-1})\omega_{k-1}^i}{q(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i)} \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^i), \quad \boldsymbol{x}_k^i \sim q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i) \tag{2.29}$$

With this and given $\widehat{p}(\boldsymbol{x}_{k-1}|D_{k-1}) = \sum_{i=1}^{N} \omega_{k-1}^i \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^i)$, the sampled form

of (2.24) becomes

$$\begin{aligned}
\widehat{p}_{\boldsymbol{\chi}_k}(\boldsymbol{x}_k) &= \sum_{i=1}^{N} \widetilde{\omega}_k^i \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^i), \qquad & \boldsymbol{x}_k^i &\sim q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i) \\
& & \widetilde{\omega}_k^i &= \frac{p(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i, \boldsymbol{u}_{k-1})\omega_{k-1}^i}{q(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i)} \\
\widehat{p}(\boldsymbol{x}_k|D_k) &= \sum_{i=1}^{N} \omega_k^i \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^i), \qquad & \omega_k^i &= \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k^i, \boldsymbol{u}_k)\widetilde{\omega}_k^i}{\sum_{i=1}^{N} p(\boldsymbol{z}_k|\boldsymbol{x}_k^i, \boldsymbol{u}_k)\widetilde{\omega}_k^i} \\
\widehat{p}(\boldsymbol{x}_k|D_k) &\leftarrow \text{Resample } \widehat{p}(\boldsymbol{x}_k|D_k)
\end{aligned} \tag{2.30}$$

A filter on this form, will be called a general particle filter (G-PF) [Schön, 2010] since it relaxes the demand of having to sample from $p_{\boldsymbol{\nu}_k}$ and instead allows for an arbitrary proposal distribution q. Just like for the A-PF, the performance depends on the choice of q and it is a good idea to condition q on D_k, $q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, D_k)$. Like previously stated, the performance of the importance sampling depends on how similar the proposal is to the posterior. A good choice of q would then be to sample directly from the posterior conditioned on the previous particle

$$q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, D_k) \propto p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{u}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, u_{k-1}) \tag{2.31}$$

Just as for the A-PF, an approximation can be used when exact sampling and evaluation isn't possible. Due to the normalisation of the weights it is also enough to only know q up to a proportionality constant.

2.6 General Auxiliary Particle Filter

The two approaches presented before are not mutually exclusive and can be deployed at the same time creating a general auxiliary particle filter (GA-PF). Given two different weight functions, $q_1(\boldsymbol{x}_{k-1})$ and $q_2(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$, where q_2 is a PDF from which samples can be taken, the recursive update becomes

$$\begin{aligned}
p(\boldsymbol{\chi}_{k-1}) &\propto q_1(\boldsymbol{\chi}_{k-1})p(\boldsymbol{x}_{k-1}|D_{k-1}) \\
p_{\boldsymbol{\chi}_k}(\boldsymbol{x}_k) &\propto \int \frac{p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1})}{q_1(\boldsymbol{\chi}_{k-1})q_2(\boldsymbol{x}_k|\boldsymbol{\chi}_{k-1})} q_2(\boldsymbol{x}_k|\boldsymbol{\chi}_{k-1})p(\boldsymbol{\chi}_{k-1})d\boldsymbol{\chi}_{k-1} \\
p(\boldsymbol{x}_k|D_k) &\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{u}_k)p_{\boldsymbol{\chi}_k}(\boldsymbol{x}_k)
\end{aligned} \tag{2.32}$$

Given $\widehat{p}(\boldsymbol{x}_{k-1}|D_{k-1}) = \sum_{i=1}^{N} \omega_{k-1}^{i} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{i})$ the sampled form becomes

$$
\begin{aligned}
&\widehat{p}(\boldsymbol{\chi}_{k-1}) = \sum_{i=1}^{N} \widetilde{\omega}_{k-1}^{i} \delta(\boldsymbol{\chi}_{k-1} - \boldsymbol{x}_{k-1}^{i}), && \widetilde{\omega}_{k-1}^{i} = \frac{q_1(\boldsymbol{x}_{k-1}^{i})\omega_{k-1}^{i}}{\sum_{i=1}^{N} q_1(\boldsymbol{x}_{k-1}^{i})\omega_{k-1}^{i}} \\
&\widehat{p}(\boldsymbol{\chi}_{k-1}) \leftarrow \text{Resample } \widehat{p}(\boldsymbol{\chi}_{k-1}) \\
&\widehat{p}_{\boldsymbol{\chi}_k}(\boldsymbol{x}_k) = \sum_{i=1}^{N} \widetilde{\omega}_{k}^{i} \delta(\boldsymbol{x}_k - \boldsymbol{x}_{k}^{i}), && \boldsymbol{x}_{k}^{i} \sim q_2(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{i}) \\
& && \widetilde{\omega}_{k}^{i} = \frac{p(\boldsymbol{x}_{k}^{i}|\boldsymbol{x}_{k-1}^{i}, \boldsymbol{u}_{k-1})\widetilde{\omega}_{k-1}^{i}}{q_1(\boldsymbol{x}_{k-1}^{i})q_2(\boldsymbol{x}_{k}^{i}|\boldsymbol{x}_{k-1}^{i})} \\
&\widehat{p}(\boldsymbol{x}_k|D_k) = \sum_{i=1}^{N} \omega_{k}^{i} \delta(\boldsymbol{x}_k - \boldsymbol{x}_{k}^{i}), && \omega_{k}^{i} = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_{k}^{i}, \boldsymbol{u}_k)\widetilde{\omega}_{k}^{i}}{\sum_{i=1}^{N} p(\boldsymbol{z}_k|\boldsymbol{x}_{k}^{i}, \boldsymbol{u}_k)\widetilde{\omega}_{k}^{i}}
\end{aligned}
\tag{2.33}
$$

The same arguments regarding the choice of q_1 and q_2 can be used here as when they were used separately so the proposed function and PDF remain the same.

3

Filter Design

The aim of this chapter is to go over the details specific for the process studied in the thesis regarding the filters used. All necessary details will be provided for an implementation of the filters discussed although no code or details regarding the implementation will be provided. This is also where process specific theory is developed in order to support the performance discussion in Chapter 4.

A standard Kalman filter will be designed in which the quantiser will be approximated as additive white noise. The purpose of this is to provide a reference for the other filters as well as examine the need for such a heavy-handed approach as particle filters. Following that, the details regarding the four different particle filters presented in the previous chapter will be given. The necessary weight functions will be defined, both the exact forms, when possible, and some coarse approximations used for comparison.

3.1 Kalman Filter - Additive Noise Approximation

Like previously mentioned, in the case of linear dynamics and Gaussian noise the state distributions are completely characterised by the mean, $\hat{\boldsymbol{x}}_{k|k}$, and covariance, $\mathbf{E}_{k|k}$, and the recursive update functions can be distilled down to the famous Kalman filter. The details are left to [Schön, 2010] or any introductory text in the subject and the end result is presented here. Given a linear Gaussian system

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k u_k + \boldsymbol{\nu}_k \quad & \boldsymbol{\nu}_k &\sim \mathcal{N}(\mathbf{0}, \widehat{\boldsymbol{R}}_k) \\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k & & \\ \boldsymbol{z}_k &= \boldsymbol{y}_k + \boldsymbol{e}_k & \boldsymbol{e}_k &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_k) \end{aligned} \tag{3.1}$$

The recursive update of $\hat{\boldsymbol{x}}_{k|k}$ and $\mathbf{E}_{k|k}$ are then given by

$$\begin{aligned}
&\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{\Phi}_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{\Gamma}_{k-1}u_{k-1} \\
&\mathbf{E}_{k|k-1} = \boldsymbol{\Phi}_{k-1}\mathbf{E}_{k-1|k-1}\boldsymbol{\Phi}_{k-1}^T + \widehat{\boldsymbol{R}}_{k-1} \qquad \textit{Prediction} \\
&\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1}) \\
&\mathbf{E}_{k|k} = \mathbf{E}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{C}\mathbf{E}_{k|k-1} \qquad \textit{Measurement Update}
\end{aligned} \tag{3.2}$$

where

$$\boldsymbol{K}_k = \mathbf{E}_{k|k-1}\boldsymbol{C}^T(\boldsymbol{C}\mathbf{E}_{k|k-1}\boldsymbol{C}^T + \boldsymbol{\Sigma}_k)^{-1} \tag{3.3}$$

The Kalman filter is exact if $\boldsymbol{\nu}_k$ and $\boldsymbol{e}_k$ are Gaussian but it but it is also, in the MMSE sense, the optimal linear estimator in cases where they are random variables of other zero mean distributions with covariance $\widehat{\boldsymbol{R}}_k$ and $\boldsymbol{\Sigma}_k$ respectively.

In order to use the Kalman Filter for the studied process the quantised measurement function needs to be approximated with some form of additive white noise model.

$$\boldsymbol{z}_k = round\left(\frac{\boldsymbol{y}_k}{\Delta q}\right)\Delta q \approx \boldsymbol{y}_k + \boldsymbol{e}_k \tag{3.4}$$

For this approximation to be exact, $\boldsymbol{e}_k$ needs to correlated with all the previous states and measurements since in the real, quantized measurement are highly dependent on the history of the system. This would make it to no longer be white noise and identifying this dependence was the entire problem in first place. A simpler approximation is therefore needed and the one used in this thesis is a worst case scenario.

Given no prior information, a quantized measurement, $\boldsymbol{z}$, of $\boldsymbol{y}$ results in a uniform distribution of $\boldsymbol{y}$. The only information available is what interval $\boldsymbol{y}$ lies in. A suitable choice of additive noise would then be a uniform distribution with variance $\boldsymbol{\Sigma}_k = \{\frac{\Delta q^2}{12}\delta_{i,j}\}$ centred around $\boldsymbol{z}_k$, i.e. having zero mean. This is the approximation referred to when in the future the Kalman filter or the additive noise approximation is mentioned.

Dynamic Measurement Approximation The previous approximation is constant and doesn't depend on the history of the system. A reasonable assumption would be that better performance could be achieved if the mean and variance of $\boldsymbol{e}_k$ would dynamically change to better reflect the belief in $\boldsymbol{y}_k$. Although this will not be further explored in the thesis, a method for achieving this will here be presented that in preliminary testing has performed well, bridging the gap between the static additive noise model and the particle filter and in some cases matching the performance of the particle filter.

The idea is to first predict $\boldsymbol{e}_k$ by approximating it with a uniform distribution with the mean and variance being the same and proportional, respectively, to the predicted mean and variance of $\boldsymbol{y}_k$. The resulting uniform distribution can then simply

be truncated to lie completely in the quantisation interval of the current measurement. The resulting filter can be summarized as follows.

$$
\begin{aligned}
\hat{\boldsymbol{x}}_{k|k-1} &= \boldsymbol{\Phi}_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{\Gamma}_{k-1}u_{k-1} \\
\mathbf{E}_{k|k-1} &= \boldsymbol{\Phi}_{k-1}\mathbf{E}_{k-1|k-1}\boldsymbol{\Phi}_{k-1}^T + \widehat{\boldsymbol{R}}_{k-1} \\
y_i^u &= \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1} + k_\delta \frac{\sqrt{12}}{2}\sqrt{(\boldsymbol{C}\mathbf{E}_{k|k-1}\boldsymbol{C}^T)_{i,i}} \\
y_i^l &= \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1} - k_\delta \frac{\sqrt{12}}{2}\sqrt{(\boldsymbol{C}\mathbf{E}_{k|k-1}\boldsymbol{C}^T)_{i,i}}
\end{aligned}
\qquad \textit{Prediction}
$$

$$
\begin{aligned}
q_i^u &= \max\left(\min\left(y_i^u, (\boldsymbol{z}_k^u)_i\right), (\boldsymbol{z}_k^l)_i\right) \\
q_i^l &= \min\left(\max\left(y_i^l, (\boldsymbol{z}_k^l)_i\right), (\boldsymbol{z}_k^u)_i\right) \\
\boldsymbol{\mu}_k &= \left\{\frac{q_i^u + q_i^l}{2}\right\}, \quad \boldsymbol{\Sigma}_k = \left\{\frac{(q_i^u - q_i^l)^2}{12}\delta_{i,j}\right\}
\end{aligned}
\qquad \textit{Truncation of Measurement Distribution}
$$

$$
\begin{aligned}
\boldsymbol{K}_k &= \mathbf{E}_{k|k-1}\boldsymbol{C}^T(\boldsymbol{C}\mathbf{E}_{k|k-1}\boldsymbol{C}^T + \boldsymbol{\Sigma}_k)^{-1} \\
\hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{\mu}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1}) \\
\mathbf{E}_{k|k} &= \mathbf{E}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{C}\mathbf{E}_{k|k-1}
\end{aligned}
\qquad \textit{Measurement Update}
$$

$\boldsymbol{z}_k^u$ and $\boldsymbol{z}_k^l$ are here vectors containing the upper and lower boundaries of the quantisation intervals corresponding to the measurement $\boldsymbol{z}_k$. k_δ is a parameter controlling the width of the approximative uniform distribution. The scaling with $\frac{\sqrt{12}}{2}$ is there to relate the width of the uniform approximation of $\boldsymbol{e}_k$ to a uniform approximation of $\boldsymbol{y}_k$. Preliminary tests pointed to a good choice being $k_\delta = 4$ but further examination is needed.

The method is easy to implement and can be converted to a "send-on-delta" event-based control scheme but is heuristic in nature and fails to accommodate for correlation between the elements of $\boldsymbol{y}_k$. To expand the concept of uniform approximation to include such effects, efficient methods for calculation of first and second moment of area for a polygon are needed. With the availability of such methods a uniform approximation can be made of $\boldsymbol{x}_{k|k-1}$ resulting in a polygon. The effect of the measurement on the polygon will then simply be a truncation, resulting in a slightly different polygon. The measurement update step can be then simply be replace by the calculation of the first and second moments of this resulting polygon.

3.2 SIR Particle Filter

Consider the discretised system defined in (1.16) and compare to the stochastic formulation of the general non-linear system used in the derivation of the particle filters, (2.5). Firstly, in order to use the SIR filter the measurement likeliness, $p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k)$, needs to be able to be evaluated. Looking at (1.16) it is seen that due to the quantizer, $\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k$ is a discrete variable. All derivations for the particle filters were done for continuous variables but this is no real problem. Bayes' rule holds when mixing discrete and continuous variable if one exchanges the PDF to the discrete variables' probability mass function (PMF), see Appendix B.1, thereby allowing the whole derivation to take place with mixed variables. Because of this and with some disregard for mathematical notation no distinction will be made between the PDF and PMF unless specific need arises.

For the measurement, this means that $p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k)$ can be defined to be a function that is one for the quantised measurement of $\boldsymbol{x}_k$ and zero otherwise. The predictive distribution can be handled in exactly the same way as discussed in the derivation of the PF found in (2.21). Summed up, the needed functions to implement a SIR filter are

$$\begin{aligned} p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k,\boldsymbol{u}_k) &= p_{\boldsymbol{\nu}_k}(\boldsymbol{x}_{k+1}-\boldsymbol{\Phi}_k\boldsymbol{x}_k-\boldsymbol{\Gamma}_k u_k) \\ p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k) &= \begin{cases} 1 & \text{if } -\frac{\Delta q}{2} \leq \boldsymbol{C}_k\boldsymbol{x}_k-\boldsymbol{z}_k < \frac{\Delta q}{2} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{3.5}$$

The evaluation of $p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k)$ is straight forward and in order to use the SIR filter samples need to be drawn from $p_{\boldsymbol{\nu}_k}$. Since $\boldsymbol{\nu}_k$ is a zero mean Gaussian multivariate random variable with covariance $\widehat{\boldsymbol{R}}_k$ this is easily done given MATLAB or some other statistics toolbox. Since $\widehat{\boldsymbol{R}}_k$ is positive definite it will have a lower triangular Cholesky decomposition.

$$\widehat{\boldsymbol{R}}_k = \widehat{\boldsymbol{N}}_k\widehat{\boldsymbol{N}}_k^T \tag{3.6}$$

Let $\overline{\boldsymbol{\nu}}$ be a vector containing independent unit Gaussian noise components, $\boldsymbol{\nu}_k$ can then be written as

$$\boldsymbol{\nu}_k = \widehat{\boldsymbol{N}}_k\overline{\boldsymbol{\nu}} \tag{3.7}$$

Samples from $\boldsymbol{\nu}_k$ can then easily be drawn simply by drawing samples from the one dimensional unit Gaussian distribution. With this the implementation of the basic SIR filter is straightforward.

3.3 Auxiliary Particle Filter

In addition to the functions needed for the SIR filter, the A-PF also needs to at least approximately evaluate the q proposed in (2.27)

$$q(\boldsymbol{x}_{k-1}|D_k) = \int p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1})d\boldsymbol{x}_k \tag{3.8}$$

To solve this the attention is turned to $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1})$. It can be written as

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1}) &= p_{\boldsymbol{\nu}_{k-1}}(\boldsymbol{x}_k - \boldsymbol{\Phi}_{k-1}\boldsymbol{x}_{k-1} - \boldsymbol{\Gamma}_{k-1}u_{k-1}) \\ &= \frac{1}{\det \widehat{\boldsymbol{N}}_{k-1}} p_{\overline{\boldsymbol{\nu}}}\big(\widehat{\boldsymbol{N}}_{k-1}^{-1}(\boldsymbol{x}_k - \boldsymbol{\Phi}_{k-1}\boldsymbol{x}_{k-1} - \boldsymbol{\Gamma}_{k-1}u_{k-1})\big) \end{aligned} \tag{3.9}$$

The inversion of $\widehat{\boldsymbol{N}}_k$ can always be made since $\widehat{\boldsymbol{R}}_k$ is positive definite and therefore have full rank and $\widehat{\boldsymbol{N}}_k$ must therefore also have full rank.

In order to simplify some expressions, lets introduce the notation

$$\boldsymbol{n}_k = \begin{bmatrix} n_k^1 \\ n_k^2 \\ n_k^3 \end{bmatrix} = \widehat{\boldsymbol{N}}_{k-1}^{-1}(\boldsymbol{x}_k - \boldsymbol{d}_k), \quad \boldsymbol{d}_k = \begin{bmatrix} d_k^1 \\ d_k^2 \\ d_k^3 \end{bmatrix} = \boldsymbol{\Phi}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{\Gamma}_{k-1}u_{k-1} \tag{3.10}$$

The components of $\overline{\boldsymbol{\nu}}$ are independent so $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1})$ can be written as

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1}) = \frac{1}{\det \widehat{\boldsymbol{N}}_{k-1}} p_{\overline{\nu}}(n_k^1) p_{\overline{\nu}}(n_k^2) p_{\overline{\nu}}(n_k^3) \tag{3.11}$$

where $p_{\overline{\nu}}$ is the one dimensional unit Gaussian distribution. Now, express $p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k)$ in $\boldsymbol{n}_k$ as well. For this purpose, define a window function as

$$W_b^a(x) = \begin{cases} 1 & \text{if } a \le x < b \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

which leads to

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k) = W_{\boldsymbol{z}_k-\Delta q/2}^{\boldsymbol{z}_k+\Delta q/2}\big(\boldsymbol{C}(\widehat{\boldsymbol{N}}_{k-1}\boldsymbol{n}_k + \boldsymbol{d}_k)\big) \tag{3.13}$$

Since $\boldsymbol{C} = [1\ 0\ 0]$ and $\boldsymbol{N}_{k-1}$ is lower triangular this simplifies to

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k) = W_{\Delta_k^l}^{\Delta_k^u}(n_k^1) \tag{3.14}$$

where

$$\Delta_k^u = \frac{1}{\widehat{\boldsymbol{N}}_{k-1}^{(1,1)}}\Big(\boldsymbol{z}_k - d_k^1 + \frac{\Delta q}{2}\Big), \quad \Delta_k^l = \frac{1}{\widehat{\boldsymbol{N}}_{k-1}^{(1,1)}}\Big(\boldsymbol{z}_k - d_k^1 - \frac{\Delta q}{2}\Big) \tag{3.15}$$

$\widehat{\boldsymbol{N}}_{k-1}^{(1,1)}$ denotes element 1,1 of $\widehat{\boldsymbol{N}}_{k-1}$. Putting all this together into (3.8) results in

$$\begin{aligned} q(\boldsymbol{x}_{k-1}|D_k) &= \int p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1})d\boldsymbol{x}_k \\ &= \int W_{\Delta_k^l}^{\Delta_k^u}(n_k^1)\frac{1}{\det\widehat{\boldsymbol{N}}_{k-1}}p_{\overline{\mathrm{v}}}(n_k^1)p_{\overline{\mathrm{v}}}(n_k^2)p_{\overline{\mathrm{v}}}(n_k^3))\det\widehat{\boldsymbol{N}}_{k-1}d\boldsymbol{n}_k \\ &= \int W_{\Delta_k^l}^{\Delta_k^u}(n_k^1)p_{\overline{\mathrm{v}}}(n_k^1)dn_k^1 \\ &= \int_{\Delta_k^l}^{\Delta_k^u}(n_k^1)p_{\overline{\mathrm{v}}}(n_k^1)dn_k^1 = \frac{1}{2}\left(\mathrm{erf}\left(\frac{\Delta_k^u}{\sqrt{2}}\right)-\mathrm{erf}\left(\frac{\Delta_k^l}{\sqrt{2}}\right)\right) \end{aligned} \tag{3.16}$$

where "erf" is the error function

$$\mathrm{erf}(x) = \frac{1}{\sqrt{\pi}}\int_{-x}^{x} e^{-t^2}dt \tag{3.17}$$

With this expression the A-PF can easily be implemented given a function for evaluation of the error function. However, due to the expensive evaluation of the error function the performance when using a coarse approximation of the error function will also be evaluated. The approximation used in these tests is

$$\widetilde{\mathrm{erf}}(x) = \begin{cases} -1+(4x^2+1)^{-1} & \text{if } x<-0.5 \\ x & \text{if } -0.5\le x\le 0.5 \\ 1-(4x^2+1)^{-1} & \text{if } x>0.5 \end{cases} \tag{3.18}$$

It was chosen by simple visual inspection in order to capture the overall features of the error function. In no way is it trying to provide an optimal estimation of either the error function or $q(\boldsymbol{x}_{k-1}|D_k)$. The purpose is simply to examine the effect of imperfect evaluation of the proposed weight function. The resulting Gaussian PDF and cumulative distribution function (CDF) can be seen in Figure 3.1.

The coarseness of the approximation is clear and the different regions of the error function approximation can easily be seen. The region $-0.5 < x/\sqrt{2} < 0.5$ is approximated with a uniform distribution and consequently loses all detail, while the tails are overemphasised since they fail to go towards zero fast enough.

3.4 General Particle Filter

For the G-PF the proposed sample function, (2.31), is

$$q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},D_{k-1}) = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k,\boldsymbol{u}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_{k-1})}{p(\boldsymbol{z}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_k,\boldsymbol{u}_{k-1})} \tag{3.19}$$

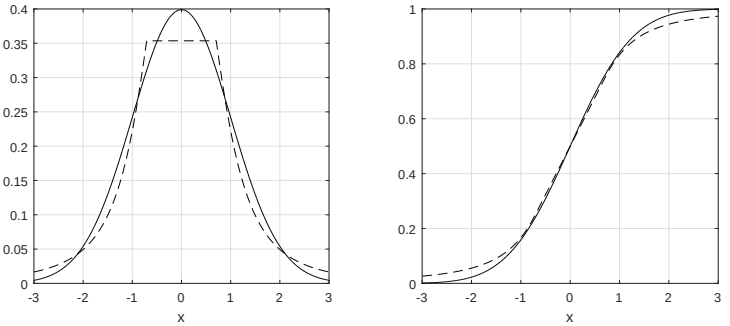

Figure 3.1 **Left:** PDF of unit normal distribution (Solid) and approximation using (3.18) (Dashed). **Right:** CDF of unit normal distribution (Solid) and approximation using (3.18) (Dashed).

First note that the denominator already have been calculated, it is the weight function used for the A-PF. Inserting this expression into the expressions for the weights in G-PF (2.30) results in

$$\begin{aligned} \widetilde{\omega}_k^i = \frac{p(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i,\boldsymbol{u}_{k-1})\omega_{k-1}^i}{q(\boldsymbol{x}_k^i|\boldsymbol{x}_{k-1}^i,D_{k-1})}, \quad \omega_k^i = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k^i,\boldsymbol{u}_k)\widetilde{\omega}_k^i}{\sum_{i=1}^N p(\boldsymbol{z}_k|\boldsymbol{x}_k^i,\boldsymbol{u}_k)\widetilde{\omega}_k^i} \\ \implies \omega_k^i = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_k,\boldsymbol{u}_{k-1})\omega_{k-1}^i}{\sum_{i=1}^N p(\boldsymbol{z}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_k,\boldsymbol{u}_{k-1})\omega_{k-1}^i} \end{aligned} \tag{3.20}$$

where

$$p(\boldsymbol{z}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_k,\boldsymbol{u}_{k-1}) = \frac{1}{2}\left(\operatorname{erf}\left(\frac{\Delta_k^u}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{\Delta_k^l}{\sqrt{2}}\right)\right) \tag{3.21}$$

To sample from q, the work done for the A-PF is reused to express q in the independent variables n_k^i. Using (3.11) and (3.14) gives

$$q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},D_{k-1}) = \frac{W_{\Delta_k^l}^{\Delta_k^u}(n_k^1)p_{\overline{\mathrm{v}}}(n_k^1)p_{\overline{\mathrm{v}}}(n_k^2)p_{\overline{\mathrm{v}}}(n_k^3)}{\det \widehat{\boldsymbol{N}}_{k-1}^{-1} p(\boldsymbol{z}_k|\boldsymbol{x}_{k-1},\boldsymbol{u}_k,\boldsymbol{u}_{k-1})} \tag{3.22}$$

From this it can be seen that the distributions for n_k^2 and n_k^3 are untouched and remain unit Gaussian. n_k^1 is the only affected variable and it is clear that it becomes a truncated unit Gaussian distribution with bounds Δ_k^u and Δ_k^l. In order to sample from q, samples can be drawn from the truncated and the regular unit Gaussian and then transformed back to the dependent variable $\boldsymbol{x}_k$ via (3.10).

For accurate sampling from the truncated Gaussian, [Botev, 2015] was used. It contains a function *trandn* which is a MATLAB implementation based on the results of

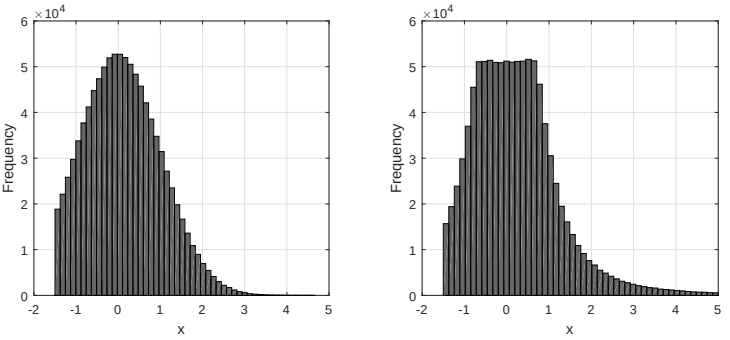

Figure 3.2 Comparison of sampling methods for a unit Gaussian distribution truncated at -1.5 and 5. Histogram over 1000000 samples and with 50 bins. **Left:** *trand* function from [Botev, 2015]. **Right:** Inverse sampling using the method proposed in Appendix B.2.

[Botev, 2017]. For comparison, the error function approximation (3.18) was used as the basis for a simple inverse transform sampling method. For details on this approximative sampling, see Appendix B.2. To take the idea of approximate sampling to the extreme $q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, D_{k-1})$ is chosen to be a uniform distribution on the entire quantisation interval given by $\boldsymbol{y}_k$. Note that the cancellation of $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, u_{k-1})$ in the calculation of the particle weights doesn't occur when these approximations are used and the full expressions are needed.

A comparison of [Botev, 2015] and the approximate sampling can be seen in Figure 3.2. Comparing with Figure 3.1, the approximate sampling manages to capture the shape of the approximate PDF, thereby suffering from the same shortcomings, while [Botev, 2015] manages much better to capture the true PDF.

3.5 General Auxiliary Particle Filter

Since the GA-PF is just a combination of the A-PF and the G-PF, nothing new is required for the GA-PF. The results from the previous two sections can be combined without any modification.

4

Open-Loop Analysis

The filters presented in Chapter 3 are here evaluated on the system system presented in Chapter 1. Keeping the control signal at zero, the system is simulated in open loop and the filters are applied to the quantised measurement. For the use in some of the performance metrics presented below, the true state of the system is recorded. At first the Kalman filter is compared to the G-PF at a range of different sample frequencies and quantisation intervals. The result of when the system deviates most from the simple linear Gaussian system is used as a basis for the comparisons of the different PF-types.

During all the simulations performed, the random seeds used for the process noise terms are kept constant to keep the comparisons fair. How good the state can be estimated is highly dependent on the trajectory of the system since the lower bound on the state-variance is highly dependent on the relation between the state and the quantisation intervals boundaries [Karlsson and Gustafsson, 2005]. The trajectory of the continuous system, resulting from the chosen random seeds, can be seen in Figure 4.1.

4.1 Performance Metrics

Mean Square Error

In general, a minimal mean square error (MMSE) estimator is given by the mean of the probability distribution [Blom et al., 2005]. This means that the Kalman filter is the MMSE estimator of a linear Gaussian system since it is the exact analytical propagation of the state mean and covariance of such a system. For a more general system, taking the mean of the distribution produced by a particle filter would produce an MMSE estimate since a particle filter produces an approximation of the state distribution. This means that the mean square error (MSE) estimate produced by the distribution mean is a good performance metric for how well the filter approximates the state distribution.

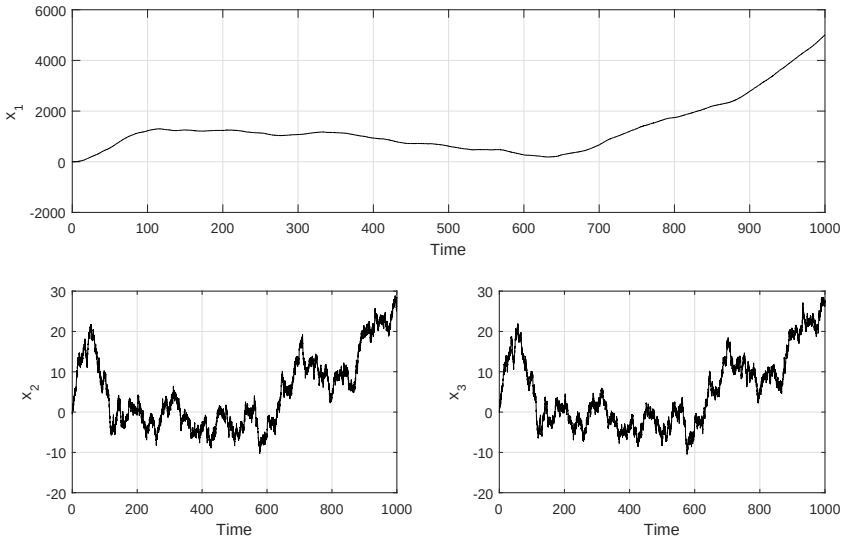

Figure 4.1 The system trajectory used during the open loop filter comparison.

The MSE is given by

$$MSE = \frac{1}{N_t} \sum_{k=1}^{N_t} (\widehat{\boldsymbol{x}}_k - \boldsymbol{x}_k^*)^T (\widehat{\boldsymbol{x}}_k - \boldsymbol{x}_k^*) \tag{4.1}$$

Where $\widehat{\boldsymbol{x}}_k$ is the state estimate given by the approximative distribution's mean, $\boldsymbol{x}_k^*$ is the true state mean and N_t is the number of time steps in the trajectory.

Effective Sample Size N_{eff}

Another way of measuring the quality of an approximation given by a PF is to look at the inter-sample dependency of the approximation. The Monte Carlo sampling methods described above utilise independently drawn samples to produce the estimates. However, mainly due to the necessary resampling step, this independence is lost in the particle filter and the quality degrades. How much it degrades depends just on how many samples are discarded when resampling and which samples happens to be favoured in prediction. To judge this inter-sample dependency the effective sample size measurement from [Carpenter et al., 1999] will be used.

The notion of effective sample size is based on the fact that the variance of the

sample mean μ of N_s independently drawn samples is

$$\text{Var}[\mu] = \frac{\sigma^2}{N_s} \tag{4.2}$$

Where σ^2 is the variance of the sampled distribution. If the samples are dependent this will no longer hold and the effective samples size is then defined as the N_{eff} that satisfies

$$\text{Var}[\mu] = \frac{\sigma^2}{N_{\text{eff}}} \tag{4.3}$$

To use this concept in a particle filter setting both $\text{Var}[\mu]$ and σ^2 need to be approximated. This is achieved by running the same simulation M times and for each of the M simulations, at each time-step calculate the sample mean and sample variance as

$$\mu_k = \sum_{i=1}^{N} \omega^i \boldsymbol{x}_k^i, \quad \sigma_k^2 = \sum_{i=1}^{N} \omega^i \boldsymbol{x}_k^i \boldsymbol{x}_k^{iT} - \mu_k \mu_k^T \tag{4.4}$$

σ^2 can now be approximated for the different components of $\boldsymbol{x}_k$ by taking the mean of σ_k^2 over all the M simulations while $\text{Var}[\mu]$ is approximated by taking the sample variance of μ_k over all simulations. By inserting these estimates in (4.3) an effective sample size at each time step can be calculated for the states. Note that the effective sample size does not need to be an integer but to make the sample size analogue clear all N_{eff} results presented will be rounded to the nearest integer.

Sample Diversity N_{div}

A similar but more simplistic concept than N_{eff} is the notion of sample diversity. Since the resampling step effectively copies other samples after discarding the low weight samples, a simple way measuring the inter-sample dependency is to count the number of unique samples, N_{div}, after each resampling. This is how sample diversity will be defined and has the benefit of not needing an average over multiple simulations to be formed.

4.2 Results

Additive Noise Approximation

In order to evaluate the validity of the additive noise approximation, the Kalman filter is compared to the G-PF using 10000 particles. The *MSE* of the two filters is calculated for a range of different quantisation intervals and sampling intervals with the results to be found in Figure 4.2. Since the G-PF approximates the optimal solution it should be clear that the additive noise approximation is close to optimal, except for when the quantisation becomes coarse and fast sampling is used.

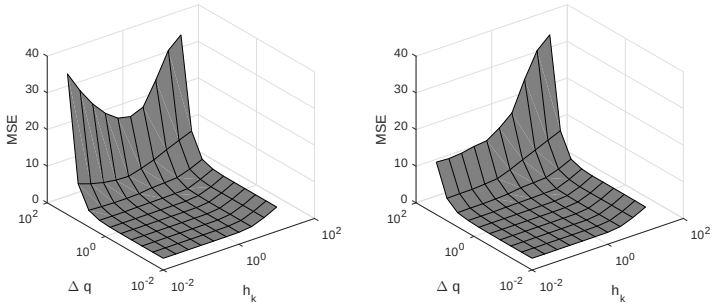

Figure 4.2 **Left:** *MSE* for Kalman filter. **Right:** *MSE* for particle filter.

The relative *MSE* of the KF compared to the G-PF found in Figure 4.3 together with the quantisation interval relative σ. σ is taken to be the standard deviation of the first component of ν_k, ν_k^1 and is chosen to represent the discrete process noise. Due to the process parameters, Table 1.1, the other components of ν_k should be of similar size.

From (A.5) and Figure 4.3 it is clear that when the sampling frequency is large, the process noise also becomes large while measurement variance of course only depends on the quantisation interval. How good the approximation is then really depends on the relative size of the measurement noise compared to the process noise. As long as the process noise dominates the additive noise approximation the Kalman filter performs well but as soon as the measurement noise becomes large, the inability of the approximation to capture the non-linearities of the process becomes apparent.

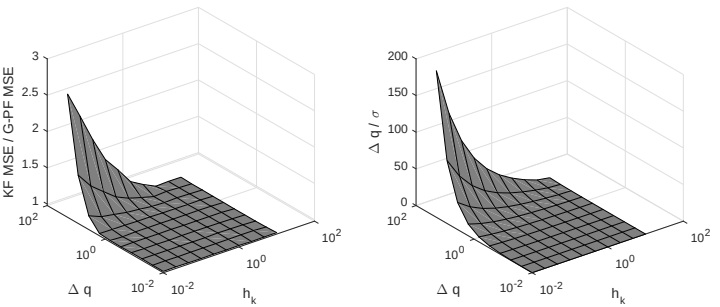

Figure 4.3 **Left:** *MSE* for particle filter relative *MSE* for Kalman filter. **Right:** Quantisation interval relative to standard deviation, σ, of the process noise ν_k^1.

The point of when the measurement noise starts to dominate is around $\Delta q > 1$ and $h_k < 1$ and becomes really overshadowing at $\Delta q > 10, h_k < 0.1$. Since the purpose of this thesis is to examine these effects the default parameters presented in Table 1.1 were chosen as $\Delta q = 20$ and $h_k = 0.05$. The G-PF *MSE* is there around 55% that of the Kalman filter. Even more extreme choices could have been made but were rejected in order to keep simulation times down and keeping some precision in the control of the system.

Particle Filter Comparisons

To benchmark the particle filters, M simulations were made for each filter and for three different number of particles used in the filters. For each simulation the *MSE* was calculated as well as the minimum sample diversity N_{div}. After the M simulations the effective sample size was calculated for each state and time-step. Due to that no real difference between the states was found, N_{eff} for each time-step was chosen as the mean of the states' effective sample size. The time mean and minimum of N_{eff} were then calculated as well as the overall N_{div} minimum, the mean N_{div} was left out due to it showing no significant difference for the different filters. The mean and standard deviation of the *MSE* were also calculated, the standard deviation was estimated with the square root of the sample variance. With $M = 100$, the results of these simulations for filters using exact evaluation of the weight functions, q, can be found in Table 4.1

Looking at the *MSE*, the differences are marginal but some trends can be seen. The SIR filter is overall the worst performer, both when it comes to accuracy, mean *MSE*, and consistency, *MSE* standard deviation. The A-PF provides consistent improvements over the standard SIR but the biggest improvement comes from the filters that utilise the general sampling strategy, i.e. the G-PF and GA-PF. Due to the variance of the *MSE*, a distinction between the G-PF and GA-PF is hard to make. The accuracy and consistency gained by weighting the resampling based on the current measurement clearly is not as important when one can base the proposal distribution on it instead. Whenever possible, that is the way to go.

Overall the filters react as expected when higher particle counts are used. The accuracy increases but with some diminishing returns when the *MSE* is getting close to around 14.6. There is a lower limit of the *MSE*, dependent on the trajectory, so it is expected that the mean *MSE* will reach some lower bound. The main advantage of the higher particle counts are better consistency, with smaller variance in *MSE*, and better filter reliability with higher effective sample sizes and sample diversity.

Filter reliability is the main problem for this system. Due to the fact that the *round* function generates a measurement probability with compact support with a step drop-off, a lot of samples will be lost in the resample step. All particles that in the

Table 4.1 Performance results for different particle filters when using exact evaluation and sampling of the weight functions q. $M = 100$.

Particle Number	Particle Filter	Mean *MSE*	STD *MSE*	Mean N_{eff}	Min N_{eff}	Min N_{div}
1000	SIR	N/A	N/A	N/A	N/A	N/A
	A-PF	N/A	N/A	N/A	N/A	N/A
	G-PF	15.0464	0.3111	132	1	1
	GA-PF	15.0708	0.3745	138	1	1
5000	SIR	14.8095	0.2312	515	2	1
	A-PF	14.7448	0.1657	504	3	3
	G-PF	14.7121	0.1244	657	3	1
	GA-PF	14.7243	0.1399	719	5	5
10000	SIR	14.7503	0.1468	1008	5	1
	A-PF	14.6630	0.1071	1013	7	16
	G-PF	14.6591	0.0875	1306	8	27
	GA-PF	14.6611	0.0665	1426	7	27
15000	SIR	14.6923	0.1227	1536	5	2
	A-PF	14.6686	0.0876	1529	9	27
	G-PF	14.6422	0.0613	1956	10	52
	GA-PF	14.6375	0.0725	2162	11	46

prediction step end up outside the quantisation interval will end up with a likelihood of zero and are therefore guaranteed to be discarded. This results in information loss and a worst case were all particles are discarded, causing the filter to fail and resulting in the incompleteness of Table 4.1 for the lowest particle count. This problem becomes especially apparent at changes in measurement. When the same measurement has been received a long time the particles start to be distributed uniformly in the interval given by that measurement and when a new measurement is made, only the ones "closest to the edge" will predict a valid measurement. This behaviour can clearly be seen in Figure 4.4 which is representative for all filters. The filter keeps a rather high sample diversity apart from several negative spikes corresponding to measurement changes.

In time-intervals when the change in $\boldsymbol{y}$ is fast, such as in the beginning and end of the trajectory Figure 4.1, the spikes are closer together but in general shorter while in the more slow changing section they are farther apart but in general longer. This is due to the fact that at every new measurement, new information is received which

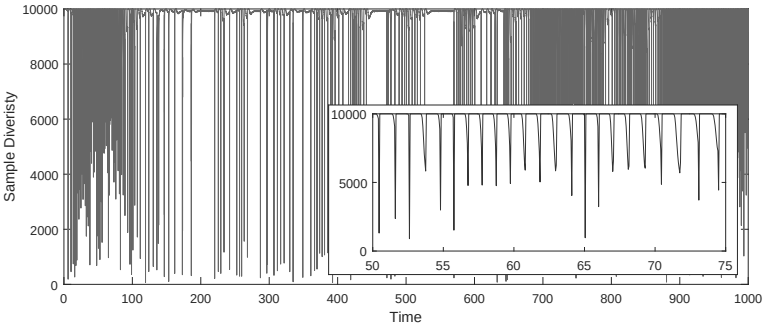

Figure 4.4 Sample diversity for G-PF using 10000 particles. **Inlay:** Magnification of time interval $t = [50, 75]$.

narrows the variance of the state distribution, allowing for an effectively higher resolution given the same number of particles. More frequent measurements then keep the variance small while infrequent measurement widens the distribution.

Although more frequent measurements give more information to work with, the worst case scenario is one of rapid changing measurements. In some cases, when the measurement rapidly toggles between two measurements, both lying in the tail of the state distributions, the sampled distribution might not be given enough time to recover after the negative spike in sample diversity. Although the number of unique samples might be large already in the next step after a spike, see Figure 4.4, they are generated from a very small set of samples resulting in them being highly correlated. This means that they no longer are an accurate representation of the state distribution and only represent a small portion of it. If the next measurement lies in the tail again, chances are that this impaired distribution fails to predict any valid particles. This recovery time after a negative sample diversity spike can clearly be seen in the effective sample size, Figure 4.5. The longer between the spikes, the higher the effective sample size manages to recover to.

This problem with the filter not recovering fast enough was the main reason for filter failure during testing. Certain trajectories drove the filter in such a way that this almost always occurred which show the importance of not only looking at the sample diversity and/or *MSE*. The GA-PF that was hard to separate solely on *MSE* became a clear winner when comparing N_{eff} Table 4.1. The step over to a generalised sampling once again provides the biggest improvement but GA-PF manages to keep a higher N_{eff} than the G-PF. It is harder to differentiate the filters in terms of minimum N_{eff} but any statements can not really be made about this, other than that it is really small. N_{eff} becomes really unreliable for small samples sizes since the measurement utilises an estimate of the distribution variance and this estimate

becomes worse and worse when the sample diversity becomes smaller. When looking at the minimum N_{div} one can also conclude that it is extremely small, a clear indicator of the process' problematic measurement function.

The obvious solution to these sample degeneracy problems is to use higher number of particles in the filters and the results support this. The minimum N_{div} scales very well with the number of particles for all filters except SIR. This is indicative of the severe shortcomings of the blind prediction, i.e. not using the current measurement. Increasing the particle count alone is not enough and the biggest improvement is seen when introducing the simplest form of conditioning on the measurement in the A-PF. This improvement is strangely not seen the same way in effective sample size. This is contributed to the effective sample size being unreliable for very small sample diversities and the improvement given by the A-PF is mainly to increase the lower bound of the sample diversity.

Another demonstration of the disconnect between N_{div} and N_{eff} can be seen in Table 4.2 which compares the different filters when using approximative evaluation of the weight functions q. From this it can immediately be seen that the uniform approximation is extremely inadequate, showing the importance of proper choice of q. However, it should be noted that the introduction of the approximate weighting greatly improves the performance of the A-PF when looking at minimum N_{div} while N_{eff} and the other metrics remain largely unaffected. The extra weight in the tails of the distribution, Figure 3.1, means a wider range of samples will be let through the resample step. These extra samples do not seem to degrade the solution compared to using exact evaluation while providing better margins when it comes to sample diversity. This is of course a balance since with constant weighting the worse performing SIR filter is received.

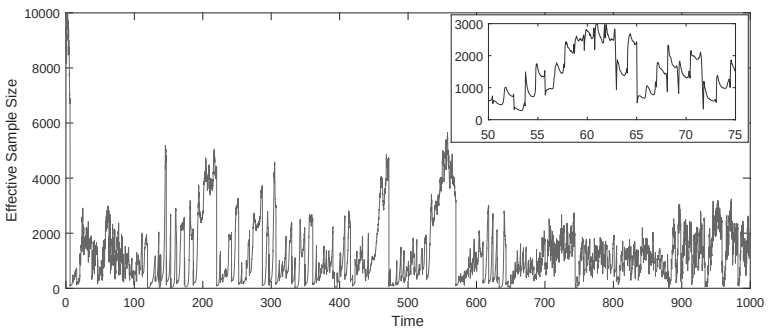

Figure 4.5 Mean effective sample size, N_{eff}, taken over the states for G-PF using 10000 particles. **Inlay:** Magnification of time interval $t = [50, 75]$.

Table 4.2 Performance results for different particle filters when using approximative evaluation and sampling of the weight functions q. Results for filters with exact evaluation and sampling are included for easy reference. The number of particles used is 10000 and $M = 100$.

Particle Filter	Approximation	Mean MSE	STD MSE	Mean N_{eff}	Min N_{eff}	Min N_{div}
SIR	None	14.7503	0.1468	1008	5	1
A-PF	None	14.6630	0.1071	1013	7	16
	Gaussian	14.6514	0.0999	1077	6	418
G-PF	None	14.6591	0.0875	1306	8	27
	Gaussian	14.6832	0.1242	568	6	7
	Uniform	25.2774	5.2317	3	0	1
GA-PF	None	14.6611	0.0665	1426	7	27
	Gaussian	14.6981	0.1262	612	5	653
	Uniform	21.6731	1.9627	7	0	69

The same advantage of the approximative resample weighting can be seen for the GA-PF as well but is in that case largely overshadowed by the losses from the approximate sampling, seen in the results for the G-PF. These performance reductions are due to the loss of similarity between the proposal and posterior distributions when the approximation is used. The G-PF with the Gaussian approximation still performs better than the basic SIR filter though. This leads to the conclusion that one should try to sample from as good a proposal distribution as possible. The results for the GA-PF when using exact sampling but approximate resample weight evaluation can be found in Table 4.3. The accuracy and consistency is retained while large improvements in minimum sample diversity are achieved, greatly increasing the reliability of the filter.

4.3 Conclusion

It has been shown that the additive noise approximation together with a Kalman filter, while it in some cases achieve good results, is inadequate for the purposes of this thesis. When the quantisation intervals become large and the need for high performance remain, high sample rate particle filters are necessary. Careful consideration has to be made in the design and evaluation of these filters. The non-linearities of the measurement function give rise to large problems with sample diversity reaching dangerously low levels if no special action is taken. Interestingly, these diversity

Table 4.3 Performance results for GA-PF when using exact sampling and approximate resample weight evaluation. $M = 100$.

Particle Number	Mean MSE	STD MSE	Mean N_{eff}	Min N_{eff}	Min N_{div}
1000	15.0219	0.2816	139	1	15
5000	14.6764	0.1145	705	4	163
10000	14.6530	0.0764	1438	7	701
15000	14.6382	0.0679	2132	12	1139

problems do not necessary manifest themselves in the accuracy and the effective sample size of the system, demonstrating the need to look at both aspects when evaluating a filter design.

Of the particle filters examined here, the clear winner was the GA-PF using exact sampling and approximate resampling weights, indicating the importance of using a good proposal distribution and the value of gaining control of the resample step. Due to this clear victory, it is this filter configuration, using 10000 particles, that will be utilised in the rest of this report. The improvements in performance when using higher particle count, although clearly evident, are deemed small enough to justify the loss in favour for shorter simulation times.

The importance of a good proposal distribution raises several interesting questions. In this case the proposal distribution could be sampled from directly but when this is not possible, how should samples be taken then? A lot of research has gone in to this question and the subsequent consequences of different sampling strategies. A good starting point for further reading in the subject is [Doucet et al., 2001] which also touches upon the concept of auxiliary particle filters. The results here naturally raise more questions regarding the A-PF. Is there an optimal choice of q? How does it relate to the proposed q used in this thesis? Other researchers have tried to answer these questions and the interested reader is advised to seek these out.

Part II

Control

5

Optimal Feedback

This chapter will cover the theory used in the solving of the optimal control problem, first presented in Chapter 1:

$$\begin{aligned} \min \quad & \mathrm{E}\left[\sum_{k=0}^{N} \mathscr{L}_k(\boldsymbol{x}_k, \boldsymbol{u}_k)\right] \\ \text{s.t.} \quad & \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\nu}_k) \\ & \boldsymbol{z}_k = h(\boldsymbol{x}_k, \boldsymbol{e}_k) \\ & \boldsymbol{x}_0 \sim p(\boldsymbol{x}_0) \end{aligned} \tag{5.1}$$

The concepts of *Dynamic Programming* [Bellman, 1957], *Dual Control* [Feldbaum, 1960], and the *Separation Principle* will form the foundation, and from it, the different feedback policies will be derived. In order to provide greater understanding of the proposed policies, the classification of different control policies into *feedback* and *closed loop* types will be touched upon briefly.

The *M-measurement Feedback* policy, first introduced by [Curry, 1970], will be presented together with the *certainty equivalent* and *open-loop optimal* [Dreyfus, 1965] feedback policies. All new control approaches presented in this thesis stem from these methods, with focus being on the special case of one-measurement feedback. This chapter will therefore serve as background for that work but also serve to briefly catalogue the prior research done before settling on the presented methods. The main innovation of the thesis is the introduction of the particle filter in the next chapter, but here a new type of feedback policy, called M-measurement Cost Feedback, will also be presented. It takes inspiration from the full M-measurement feedback policy and bridges the gap between the policies.

For simplicity and in order to isolate the effects of the quantized measurements, two restrictions will be made on the problem. Only cost functions where the immediate cost, $\mathscr{L}_k$, is quadratic and processes where the dynamics, f, are linear will be

considered when deriving the control policies:

$$\begin{aligned} \min \quad & \mathrm{E}\left[\boldsymbol{x}_N^T \boldsymbol{Q}_N^x \boldsymbol{x}_N + \sum_{k=0}^{N-1} \boldsymbol{x}_k^T \boldsymbol{Q}_k^x \boldsymbol{x}_k + 2\boldsymbol{x}_k^T \boldsymbol{Q}_k^{xu} \boldsymbol{u}_k + \boldsymbol{u}_k^T \boldsymbol{Q}_k^u \boldsymbol{u}_k\right] \\ \text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k u_k + \boldsymbol{\nu}_k \\ & \boldsymbol{z}_k = h(\boldsymbol{x}_k, \boldsymbol{e}_k) \\ & \boldsymbol{x}_0 \sim p(\boldsymbol{x}_0) \end{aligned} \tag{5.2}$$

The process noise is also further restricted to be zero-mean white noise. Note that the control cost at time-step N is ignored, since regardless of weights $\boldsymbol{Q}_N^{xu}$ and $\boldsymbol{Q}_N^u$ the optimal choice is $\boldsymbol{u}_N = \boldsymbol{0}$, since $\boldsymbol{u}_N$ can not affect the state inside the control horizon.

5.1 Dynamic Programming

The solutions of both optimisation problems (5.1) and (5.2) are far from trivial and can only in special cases be found analytically. The general approach is a recursive one, where one starts at the last time step and solves the optimal control problem for that last step. Based on that one can solve for the second to last step, and so forth. In this section these recursive equations will be formulated, and concepts like control and estimation separation and dual control will also be discussed. In order to properly formulate them and understand their consequences it's important to consider over what variables the expected value in the cost function J is taken.

The state variables $\boldsymbol{x}_k$ are stochastic so the expected value is of course taken over them, but they also depend heavily on the measurements and the control signal which can't be known beforehand. It is therefore important to consider the stochastic nature of future measurements and controls when taking the expected value. In order to clearly demonstrate this, gather all known information about the system at time-step k in $\mathscr{D}_k$ and lets decide on a control policy, π. π gives a control signal based on the available information $u_k = \pi(k, \mathscr{D}_k)$. The exact representation of $\mathscr{D}_k$ and $\pi(k, \mathscr{D}_k)$ can vary. An intuitive and easy representation of $\mathscr{D}_k$ would simply be to store the initial distribution of $\boldsymbol{x}_0$ and the subsequent sequence of measurements and controls, $\{\boldsymbol{z}_k, \boldsymbol{u}_{k-1}, \boldsymbol{z}_{k-1}, \boldsymbol{u}_{k-2} \dots\}$. Another alternative would be to store the current distribution of $\boldsymbol{x}_k$, it does not matter. What matters is that the probability of all future events can be evaluated from $\mathscr{D}_k$ and that $\mathscr{D}_k$ can be updated to $\mathscr{D}_{k+1}$ based on $\mathscr{D}_k$, the process model, the measurement $\boldsymbol{z}_{k+1}$ and the control policy π. With these notations the cost function J can for a given control policy π be rewritten to make all these hidden expectations explicit by repeated reverse use of the law of

total expectation,

$$\begin{aligned} J^{\pi} &= \mathrm{E}\left[\sum_{k=0}^{N} \mathscr{L}_k(\boldsymbol{x}_k, \boldsymbol{u}_k)\right] \\ &= \mathrm{E}_{\boldsymbol{z}_1}\left[\mathrm{E}_{\boldsymbol{z}_2}\left[\ldots \mathrm{E}_{\boldsymbol{z}_N}\left[\mathrm{E}\left[\sum_{k=0}^{N} \mathscr{L}_k(\boldsymbol{x}_k, \boldsymbol{\pi}(k, \mathscr{D}_k))\middle|\mathscr{D}_N\right]\middle|\mathscr{D}_{N-1}\right]\ldots\middle|\mathscr{D}_1\right]\middle|\mathscr{D}_0\right] \end{aligned} \tag{5.3}$$

where the innermost expectation is taken over only the state variables. By noting that $\boldsymbol{u}_k$ does not depend on the future information and separating out the terms of the sum it can be written as

$$\begin{aligned} J^{\pi} = &\mathrm{E}_{\boldsymbol{z}_0}\Big[\ \mathrm{E}\left[\mathscr{L}_0(\boldsymbol{x}_0, \boldsymbol{\pi}(0, \mathscr{D}_0))\middle|\mathscr{D}_0\right] + \mathrm{E}_{\boldsymbol{z}_1}\Big[\mathrm{E}\left[\mathscr{L}_1(\boldsymbol{x}_1, \boldsymbol{\pi}(1, \mathscr{D}_1))\middle|\mathscr{D}_1\right] \\ &+ \mathrm{E}_{\boldsymbol{z}_2}\Big[\ldots\ \mathrm{E}_{\boldsymbol{z}_N}\left[\mathrm{E}\left[\mathscr{L}_N(\boldsymbol{x}_N, \boldsymbol{\pi}(N, \mathscr{D}_N))\middle|\mathscr{D}_N\right]\middle|\mathscr{D}_{N-1}\right]\ \ldots\middle|\mathscr{D}_2\Big]\middle|\mathscr{D}_1\Big]\middle|\mathscr{D}_0\Big] \end{aligned} \tag{5.4}$$

This can be summarised in the following recursive relation:

$$\begin{aligned} J_k^{\pi}(\mathscr{D}_k) &= \mathrm{E}\left[\mathscr{L}_k(\boldsymbol{x}_k, \boldsymbol{\pi}(k, \mathscr{D}_k))\middle|\mathscr{D}_k\right] + \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[J_{k+1}^{\pi}(\mathscr{D}_{k+1})\middle|\mathscr{D}_k\right] \\ J_N^{\pi}(\mathscr{D}_N) &= \mathrm{E}\left[\mathscr{L}_N(\boldsymbol{x}_N, \boldsymbol{\pi}(N, \mathscr{D}_N))\middle|\mathscr{D}_N\right] \end{aligned} \tag{5.5}$$

with $J^{\pi} = J_0^{\pi}(\mathscr{D}_0)$. An optimal policy $\boldsymbol{\pi}^*$ is then simply a policy that minimizes $J_0^{\pi}(\mathscr{D}_0)$. The minimal cost is then obtained by minimizing each term in the recursion relation:

$$\begin{aligned} J_k^{*}(\mathscr{D}_k) &= \min_{\pi}\ \mathrm{E}\left[\mathscr{L}_k(\boldsymbol{x}_k, \boldsymbol{\pi}(k, \mathscr{D}_k))\middle|\mathscr{D}_k\right] + \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[J_{k+1}^{*}(\mathscr{D}_{k+1})\middle|\mathscr{D}_k\right] \\ J_N^{*}(\mathscr{D}_N) &= \min_{\pi}\ \mathrm{E}\left[\mathscr{L}_N(\boldsymbol{x}_N, \boldsymbol{\pi}(N, \mathscr{D}_N))\middle|\mathscr{D}_N\right] \end{aligned} \tag{5.6}$$

From this, the principle of optimality can be seen. It states that an optimal policy generating a control at time-step k needs to still be an optimal policy at the time-step $k+1$ for the information state resulting from that control. This optimal policy $\boldsymbol{\pi}^*$ is known as a *closed-loop* policy since it takes into consideration all future measurements through the expectation over $\boldsymbol{z}_{k+1}$ in $J_k^*(\mathscr{D}_k)$. Finding an analytical expression for this functional $\boldsymbol{\pi}^*$ is however in most cases not feasible. For that reason a number of different numerical and/or approximative approaches have been used. The methods presented in this thesis belong to a group called *M-measurement feedback* policies. They work by, at each time-step k, minimizing an approximation of the *cost-to-go*, $J_k^{\pi}(\mathscr{D}_k)$, formed by replacing J_{k+M}^* with some approximation that doesn't depend on future measurements. This way the control at time-step k only considers the effect it has on the M next measurements.

5.2 Dual Effect, Certainty Equivalence and Separation

To demonstrate the important concept of dual effect the general recursion relation given by (5.1) and (5.6) will be left behind in favour for the linear quadratic (LQ)

problem (5.2) and the resulting recursion relation:

$$\begin{aligned} J_k^\pi(\mathscr{D}_k) &= \mathrm{E}\left[\boldsymbol{x}_k^T \boldsymbol{Q}_k^x \boldsymbol{x}_k + 2\boldsymbol{x}_k^T \boldsymbol{Q}_k^{xu} \boldsymbol{u}_k + \boldsymbol{u}_k^T \boldsymbol{Q}_k^u \boldsymbol{u}_k \middle| \mathscr{D}_k\right] + \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[J_{k+1}^\pi(\mathscr{D}_{k+1}) \middle| \mathscr{D}_k\right] \\ J_N^\pi(\mathscr{D}_N) &= \mathrm{E}\left[\boldsymbol{x}_N^T \boldsymbol{Q}_N^x \boldsymbol{x}_N \middle| \mathscr{D}_N\right] \end{aligned} \tag{5.7}$$

where $\boldsymbol{u}_k = \boldsymbol{\pi}(k, \mathscr{D}_k)$. The dual effect, certainty equivalence and the separation principle can all be presented in more general terms [Bar-Shalom and Tse, 1974], [Tse and Bar-Shalom, 1975] but the linear quadratic case is sufficient to provide the necessary understanding. The dual effect becomes apparent when solving (5.7). This is done, as previously mentioned, by solving for the optimal control at the end-step and recursively working backwards. Since J_N^π is clearly independent of $\boldsymbol{u}_N$, we start at time-step $N-1$. For notational convenience let $s = N-1$ and $s+1 = N$. We then have

$$\begin{aligned} J_s^\pi(\mathscr{D}_s) &= \mathrm{E}\left[\boldsymbol{x}_s^T \boldsymbol{Q}_s^x \boldsymbol{x}_s + 2\boldsymbol{x}_s^T \boldsymbol{Q}_s^{xu} \boldsymbol{u}_s + \boldsymbol{u}_s^T \boldsymbol{Q}_s^u \boldsymbol{u}_s \middle| \mathscr{D}_s\right] \\ &\qquad + \mathrm{E}_{\boldsymbol{z}_{s+1}}\left[\mathrm{E}\left[\boldsymbol{x}_{s+1}^T \boldsymbol{Q}_{s+1}^x \boldsymbol{x}_{s+1} \middle| \mathscr{D}_{s+1}\right] \middle| \mathscr{D}_s\right] \\ &= \mathrm{E}\left[\boldsymbol{x}_s^T \boldsymbol{Q}_s^x \boldsymbol{x}_s + 2\boldsymbol{x}_s^T \boldsymbol{Q}_s^{xu} \boldsymbol{u}_s + \boldsymbol{u}_s^T \boldsymbol{Q}_s^u \boldsymbol{u}_s + \boldsymbol{x}_{s+1}^T \boldsymbol{Q}_{s+1}^x \boldsymbol{x}_{s+1} \middle| \mathscr{D}_s\right] \end{aligned} \tag{5.8}$$

This can be re-written to only include the state and control at time-step s by inserting $\boldsymbol{x}_{s+1} = \boldsymbol{\Phi}_s \boldsymbol{x}_s + \boldsymbol{\Gamma}_s u_s + \boldsymbol{\nu}_s$ and rearranging. Before that, note that the process noise, $\boldsymbol{\nu}_s$, is zero mean and independent of the state and control. This means that all cross-terms between the noise and state/control are zero, and the term only containing the noise can't be affected by the control and can therefore be dropped without affecting the optimal control policy. This leads to

$$\begin{aligned} J_s^\pi(\mathscr{D}_s) = \mathrm{E}\left[\boldsymbol{x}_s^T (\boldsymbol{Q}_s^x + \boldsymbol{\Phi}_s^T \boldsymbol{Q}_{s+1}^x \boldsymbol{\Phi}_s) \boldsymbol{x}_s + 2\boldsymbol{x}_s^T (\boldsymbol{Q}_s^{xu} + \boldsymbol{\Phi}_s^T \boldsymbol{Q}_{s+1}^x \boldsymbol{\Gamma}_s) \boldsymbol{u}_s \right. \\ \left. + \boldsymbol{u}_s^T (\boldsymbol{Q}_s^u + \boldsymbol{\Gamma}_s^T \boldsymbol{Q}_{s+1}^x \boldsymbol{\Gamma}_s) \boldsymbol{u}_s \middle| \mathscr{D}_s\right] \end{aligned} \tag{5.9}$$

This can be minimized with respect to $\boldsymbol{u}_s$ by completing the squares, resulting in $\boldsymbol{u}_s = -(\boldsymbol{Q}_s^u + \boldsymbol{\Gamma}_s^T \boldsymbol{Q}_{s+1}^x \boldsymbol{\Gamma}_s)^{-1} (\boldsymbol{Q}_s^{xu} + \boldsymbol{\Phi}_s^T \boldsymbol{Q}_{s+1}^x \boldsymbol{\Gamma}_s)^T \hat{\boldsymbol{x}}_{s|s}$ and

$$J_s^*(\mathscr{D}_s) = \mathrm{E}\left[\boldsymbol{x}_s^T (\boldsymbol{Q}_s^x + \boldsymbol{\Phi}_s^T \boldsymbol{Q}_{s+1}^x \boldsymbol{\Phi}_s) \boldsymbol{x}_s \middle| \mathscr{D}_s\right] - \hat{\boldsymbol{x}}_{s|s}^T \boldsymbol{C}_s \hat{\boldsymbol{x}}_{s|s} \tag{5.10}$$

where $\hat{\boldsymbol{x}}_{s|s} = \mathrm{E}[\boldsymbol{x}_s | \mathscr{D}_s]$ and

$$\boldsymbol{C}_s = (\boldsymbol{Q}_s^{xu} + \boldsymbol{\Phi}_s^T \boldsymbol{Q}_{s+1}^x \boldsymbol{\Gamma}_s)(\boldsymbol{Q}_s^u + \boldsymbol{\Gamma}_s^T \boldsymbol{Q}_{s+1}^x \boldsymbol{\Gamma}_s)^{-1} (\boldsymbol{Q}_s^{xu} + \boldsymbol{\Phi}_s^T \boldsymbol{Q}_{s+1}^x \boldsymbol{\Gamma}_s)^T \tag{5.11}$$

the last term can be rewritten as

$$\begin{aligned} \hat{\boldsymbol{x}}_{s|s}^T \boldsymbol{C}_s \hat{\boldsymbol{x}}_{s|s} &= -\mathrm{E}\left[\boldsymbol{e}_{s|s}^T \boldsymbol{C}_s \boldsymbol{e}_{s|s} \middle| \mathscr{D}_s\right] + \mathrm{E}\left[\boldsymbol{x}_s^T \boldsymbol{C}_s \boldsymbol{x}_s \middle| \mathscr{D}_s\right] \\ &= -\mathrm{tr}(\boldsymbol{C}_s \mathbf{E}_{s|s}) + \mathrm{E}\left[\boldsymbol{x}_s^T \boldsymbol{C}_s \boldsymbol{x}_s \middle| \mathscr{D}_s\right] \end{aligned} \tag{5.12}$$

where $\boldsymbol{e}_{s|s}^T = \boldsymbol{x}_s^T - \hat{\boldsymbol{x}}_{s|s}^T$ is the mean-estimate error and $\mathbf{E}_{s|s} = \mathrm{E}\left[\boldsymbol{e}_{s|s} \boldsymbol{e}_{s|s}^T \middle| \mathscr{D}_s\right]$ is the state covariance. This allows the optimal cost for step s to be written as

$$J_s^\pi(\mathscr{D}_s) = \mathrm{E}\left[\boldsymbol{x}_s^T \boldsymbol{S}_s \boldsymbol{x}_s \middle| \mathscr{D}_s\right] + \mathrm{tr}(\boldsymbol{C}_s \mathbf{E}_{s|s}) \tag{5.13}$$

and for step $s-1$ as

$$\begin{aligned} J^{\pi}_{s-1}(\mathscr{D}_{s-1}) &= \mathrm{E}\left[\boldsymbol{x}^T_{s-1}\boldsymbol{Q}^x_{s-1}\boldsymbol{x}_{s-1} + 2\boldsymbol{x}^T_{s-1}\boldsymbol{Q}^{xu}_{s-1}\boldsymbol{u}_{s-1} + \boldsymbol{u}^T_{s-1}\boldsymbol{Q}^u_{s-1}\boldsymbol{u}_{s-1}\middle|\mathscr{D}_{s-1}\right] \\ &\quad + \mathrm{E}_{\boldsymbol{z}_s}\left[\mathrm{E}\left[\boldsymbol{x}^T_s\boldsymbol{S}_s\boldsymbol{x}_s\middle|\mathscr{D}_s\right] + \mathrm{tr}(\boldsymbol{C}_s\mathbf{E}_{s|s})\middle|\mathscr{D}_{s-1}\right] \\ &= \mathrm{E}\left[\boldsymbol{x}^T_{s-1}\boldsymbol{Q}^x_{s-1}\boldsymbol{x}_{s-1} + 2\boldsymbol{x}^T_{s-1}\boldsymbol{Q}^{xu}_{s-1}\boldsymbol{u}_{s-1}\right. \\ &\quad \left. + \boldsymbol{u}^T_{s-1}\boldsymbol{Q}^u_{s-1}\boldsymbol{u}_{s-1} + \boldsymbol{x}^T_s\boldsymbol{S}_s\boldsymbol{x}_s\middle|\mathscr{D}_{s-1}\right] + \mathrm{E}_{\boldsymbol{z}_s}\left[\mathrm{tr}(\boldsymbol{C}_s\mathbf{E}_{s|s})\middle|\mathscr{D}_{s-1}\right] \end{aligned} \tag{5.14}$$

where

$$\begin{aligned} \boldsymbol{S}_s &= \boldsymbol{Q}^x_s + \boldsymbol{\Phi}^T_s\boldsymbol{Q}^x_{s+1}\boldsymbol{\Phi}_s \\ &\quad - (\boldsymbol{Q}^{xu}_s + \boldsymbol{\Phi}^T_s\boldsymbol{Q}^x_{s+1}\boldsymbol{\Gamma}_s)(\boldsymbol{Q}^u_s + \boldsymbol{\Gamma}^T_s\boldsymbol{Q}^x_{s+1}\boldsymbol{\Gamma}_s)^{-1}(\boldsymbol{Q}^{xu}_s + \boldsymbol{\Phi}^T_s\boldsymbol{Q}^x_{s+1}\boldsymbol{\Gamma}_s)^T \end{aligned} \tag{5.15}$$

From this a number of things can be seen, mainly that the control of the last step only depends on the state through $\hat{\boldsymbol{x}}_{s|s}$ and that the cost for step $s-1$ is in the same quadratic form as for step s if it wasn't for the future state covariance's contribution to the total cost. This is of importance since if the control, $\boldsymbol{u}_{s-1}$, has no effect on the state covariance, $\mathbf{E}_{s|s}$, that term can be discarded and the cost can be minimized based only on the estimate $\hat{\boldsymbol{x}}_{s-1|s-1}$, just as for step s. If the future state covariance is affected by control, the optimal control now also needs to balance the uncertainty of the state estimate against the cost associated with the estimate. When the control affects the state covariance, the optimisation problem is said to have a dual effect, because of the need to balance the uncertainty of the estimate and the estimated cost.

As previously mentioned, when no dual effect is present, the optimal control at step $s-1$ can be calculated in the same way as for s. This continues for step $s-2$, $s-3$ and so forth, resulting in the following equivalent optimal cost and control for time step k:

$$\begin{aligned} J^{CE}_k(\mathscr{D}_k) &= \mathrm{E}\left[\boldsymbol{x}^T_k\boldsymbol{S}_k\boldsymbol{x}_k\middle|\mathscr{D}_k\right] \\ \boldsymbol{u}^{CE}_k &= -(\boldsymbol{Q}^u_k + \boldsymbol{\Gamma}^T_k\boldsymbol{S}_{k+1}\boldsymbol{\Gamma}_k)^{-1}(\boldsymbol{Q}^{xu}_k + \boldsymbol{\Phi}^T_k\boldsymbol{S}_{k+1}\boldsymbol{\Gamma}_k)^T\hat{\boldsymbol{x}}_{k|k} \end{aligned} \tag{5.16}$$

where $\boldsymbol{S}_k$ is given by the following recurrence relation

$$\begin{aligned} \boldsymbol{S}_k &= \boldsymbol{Q}^x_k + \boldsymbol{\Phi}^T_k\boldsymbol{S}_{k+1}\boldsymbol{\Phi}_k \\ &\quad - (\boldsymbol{Q}^{xu}_k + \boldsymbol{\Phi}^T_k\boldsymbol{S}_{k+1}\boldsymbol{\Gamma}_k)(\boldsymbol{Q}^u_k + \boldsymbol{\Gamma}^T_k\boldsymbol{S}_{k+1}\boldsymbol{\Gamma}_k)^{-1}(\boldsymbol{Q}^{xu}_k + \boldsymbol{\Phi}^T_k\boldsymbol{S}_{k+1}\boldsymbol{\Gamma}_k)^T \\ \boldsymbol{S}_N &= \boldsymbol{Q}^x_N \end{aligned} \tag{5.17}$$

This is the discrete-time algebraic Riccati equation which should be recognised from the deterministic linear quadratic control problem. In fact, the only difference is that the state $\boldsymbol{x}_k$ is replaced by the mean state, $\hat{\boldsymbol{x}}_{k|k}$. This property is called *certainty equivalence* since the control law is the same as if certainty about the state

applied. A slightly less strict concept is that of the separation principle that applies if the optimal control policy only depends on the state through $\hat{\boldsymbol{x}}_{k|k}$, i.e. the uncertainty of the estimate does not affect how the system is controlled. The optimal stochastic control problem can then be separated into two parts, first being to accurately estimate $\hat{\boldsymbol{x}}_{k|k}$ and the other being finding the optimal control based on that estimate.

When no dual effect is present, certainty equivalence, and thereby also the separation principle, obviously applies in this case. Similar statements can even be made for systems with non-linear dynamics [Tse and Bar-Shalom, 1975] but even so, the linear system with quantized measurements studied in this report does have dual effect. However, depending on the dynamics and the size of the quantisation interval, the impact of the dual effect on the control cost might be negligible. In that case the certainty equivalent (CE) control policy presented above might be a good approximation for the optimal control.

Since the cost-to-go for step $k+1$ is approximated by $J^*_{k+1}(\mathscr{D}_{k+1}) \approx \mathrm{E}\left[\boldsymbol{x}^T_{k+1}\boldsymbol{S}_{k+1}\boldsymbol{x}_{k+1}\middle|\mathscr{D}_{k+1}\right]$ when putting the CE policy into the context of m-measurement feedback, one might be inclined to call it a 1-measurement feedback policy. However, since the chosen approximation results in a cost that is independent of the likelihood of future measurement a more accurate classification would be a zero-measurement feedback policy, or simply just a feedback policy, since it's then only using information about the current state, without consideration for the future.

5.3 Open Loop Optimal Feedback

Another (0-measurement) feedback policy is the open-loop optimal feedback (OLOF) [Dreyfus, 1965]. It assumes no further measurements will be made after the current time step and approximates the cost at step k with the open loop cost. It will be shown that in this LQ setting the resulting control policy is equivalent to the CE policy but unlike the CE policy, it doesn't completely disregard the error dynamic and will for that reason later be used in the derivation of the one-measurement and M-measurement policies.

By the assumption of no further measurements, the cost at step k for the LQ problem can be written as

$$J^{OL}_k(\mathscr{D}_k) = \mathrm{E}\left[\sum_{t=k}^{N} \boldsymbol{x}^T_t\boldsymbol{Q}^x_t\boldsymbol{x}_t + 2\boldsymbol{x}^T_t\boldsymbol{Q}^{xu}_t\boldsymbol{u}_t + \boldsymbol{u}^T_t\boldsymbol{Q}^u_t\boldsymbol{u}_t \middle| \mathscr{D}_k\right] \tag{5.18}$$

Separating the state into estimate and estimation error, $\boldsymbol{x}_t = \hat{\boldsymbol{x}}_{t|k} + \boldsymbol{e}_{t|k}$, gives

$$J_k^{OL}(\mathscr{D}_k) = \sum_{t=k}^{N} \hat{\boldsymbol{x}}_{t|k}^T \boldsymbol{Q}_t^x \hat{\boldsymbol{x}}_{t|k} + 2\hat{\boldsymbol{x}}_{t|k}^T \boldsymbol{Q}_t^{xu} \boldsymbol{u}_t + \boldsymbol{u}_t^T \boldsymbol{Q}_t^u \boldsymbol{u}_t + \mathrm{E}\left[\boldsymbol{e}_{t|k}^T \boldsymbol{Q}_t^x \boldsymbol{e}_{t|k} \middle| \mathscr{D}_k\right] \tag{5.19}$$

where $\hat{\boldsymbol{x}}_{t|k} = \mathrm{E}[\boldsymbol{x}_t|\mathscr{D}_k, \boldsymbol{U}_{k:t-1}]$ and $\boldsymbol{U}_{k:t-1} = \{\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \ldots \boldsymbol{u}_{t-1}\}$. The cross terms with $\boldsymbol{e}_{t|k}$ vanish since $\mathrm{E}[\boldsymbol{e}_{t|k}|\mathscr{D}_k, \boldsymbol{U}_{k:t-1}] = \boldsymbol{0}$. Note that, without the presence of measurement, the mean and error dynamics are given by

$$\begin{aligned} \hat{\boldsymbol{x}}_{t+1|k} &= \boldsymbol{\Phi}_t \hat{\boldsymbol{x}}_{t|k} + \boldsymbol{\Gamma}_t \boldsymbol{u}_t \\ \boldsymbol{e}_{t+1|k} &= \boldsymbol{\Phi}_t \boldsymbol{e}_{t|k} + \boldsymbol{\nu}_t \end{aligned} \tag{5.20}$$

By noting that the error is independent of the control, the cost can be divided into two sums with only the first being affected by the control.

$$\begin{aligned} J_k^{OL}(\mathscr{D}_k) = &\sum_{t=k}^{N} \hat{\boldsymbol{x}}_{t|k}^T \boldsymbol{Q}_t^x \hat{\boldsymbol{x}}_{t|k} + 2\hat{\boldsymbol{x}}_{t|k} \boldsymbol{Q}_t^{xu} \boldsymbol{u}_t + \boldsymbol{u}_t^T \boldsymbol{Q}_t^u \boldsymbol{u}_t \\ &+ \mathrm{E}\left[\sum_{t=k}^{N} \boldsymbol{e}_{t|k}^T \boldsymbol{Q}_t^x \boldsymbol{e}_{t|k} \middle| \mathscr{D}_k\right] \end{aligned} \tag{5.21}$$

The first sum can be seen as the cost for a deterministic process with the state replaced by $\hat{\boldsymbol{x}}_{t|k}$. Since the dynamics of $\hat{\boldsymbol{x}}_{t|k}$ are linear and deterministic this term can easily be minimized using standard LQR techniques, resulting in the same control as the CE policy found in (5.16) with the same Riccati equations as in (5.17). The optimal open-loop feedback cost can then be written as

$$J_k^{OL*}(\mathscr{D}_k) = \hat{\boldsymbol{x}}_{k|k}^T \boldsymbol{S}_k \hat{\boldsymbol{x}}_{k|k} + \mathrm{E}\left[\sum_{t=k}^{N} \boldsymbol{e}_{t|k}^T \boldsymbol{Q}_t^x \boldsymbol{e}_{t|k} \middle| \mathscr{D}_k\right] \tag{5.22}$$

For future use the second term will be simplified further with the goal of expressing it in only the error at time-step k. First express the future errors in $\boldsymbol{e}_{k|k}$,

$$\begin{aligned} \boldsymbol{e}_{t+1|k} &= \boldsymbol{\Phi}_t \boldsymbol{e}_{t|k} + \boldsymbol{\nu}_t \\ &= \boldsymbol{\Phi}_{t+1:k} \boldsymbol{e}_{k|k} + \sum_{i=k}^{t} \boldsymbol{\Phi}_{t+1:i+1} \boldsymbol{\nu}_i \end{aligned} \tag{5.23}$$

where $\boldsymbol{\Phi}_{t:k} = \boldsymbol{\Phi}_{t-1}\boldsymbol{\Phi}_{t-2}\ldots\boldsymbol{\Phi}_k$ with $\boldsymbol{\Phi}_{i:i} = \boldsymbol{I}$. Insert this into the second term and note

that all cross terms between $\boldsymbol{\nu}_i$ and $\boldsymbol{e}_{k|k}$ are zero since they are independent,

$$\begin{aligned}
\mathrm{E}\left[\sum_{t=k}^{N} \boldsymbol{e}_{t|k}^T \boldsymbol{Q}_t^x \boldsymbol{e}_{t|k} \middle| \mathscr{D}_k\right] &= \mathrm{E}\left[\sum_{t=k}^{N} \boldsymbol{e}_{k|k}^T \boldsymbol{\Phi}_{t:k}^T \boldsymbol{Q}_t^x \boldsymbol{\Phi}_{t:k} \boldsymbol{e}_{k|k} \middle| \mathscr{D}_k\right] \\
&\quad + \mathrm{E}\left[\sum_{t=k+1}^{N} \sum_{i=k}^{t-1} \boldsymbol{\nu}_i^T \boldsymbol{\Phi}_{t:i+1}^T \boldsymbol{Q}_t^x \boldsymbol{\Phi}_{t:i+1} \boldsymbol{\nu}_i \middle| \mathscr{D}_k\right] \\
&= \mathrm{E}\left[\boldsymbol{e}_{k|k}^T \sum_{t=k}^{N} \left(\boldsymbol{\Phi}_{t:k}^T \boldsymbol{Q}_t^x \boldsymbol{\Phi}_{t:k}\right) \boldsymbol{e}_{k|k} \middle| \mathscr{D}_k\right] + const \\
&= \mathrm{E}\left[\boldsymbol{e}_{k|k}^T \boldsymbol{F}_k \boldsymbol{e}_{k|k} \middle| \mathscr{D}_k\right] + const \\
&= \mathrm{tr}\left(\mathbf{E}_{k|k} \boldsymbol{F}_k\right) + const
\end{aligned} \tag{5.24}$$

The sum containing the noise terms is constant regardless of state and only depends on the process itself. It can therefore be ignored in all cases of minimisation. The matrix $\boldsymbol{F}_k$ can be calculated with the following relation

$$\begin{aligned}
\boldsymbol{F}_k &= \boldsymbol{\Phi}_k^T \boldsymbol{F}_{k+1} \boldsymbol{\Phi}_k + \boldsymbol{Q}_k^x \\
\boldsymbol{F}_N &= \boldsymbol{Q}_N^x
\end{aligned} \tag{5.25}$$

Ignoring the constant terms, the equivalent optimal cost is then

$$\begin{aligned}
J_k^{OL*}(\mathscr{D}_k) &= \hat{\boldsymbol{x}}_{k|k}^T \boldsymbol{S}_k \hat{\boldsymbol{x}}_{k|k} + \mathrm{E}\left[\boldsymbol{e}_{k|k}^T \boldsymbol{F}_k \boldsymbol{e}_{k|k} \middle| \mathscr{D}_k\right] \\
&= \mathrm{E}\left[\boldsymbol{x}_k^T \boldsymbol{S}_k \boldsymbol{x}_k \middle| \mathscr{D}_k\right] + \mathrm{E}\left[\boldsymbol{e}_{k|k}^T (\boldsymbol{F}_k - \boldsymbol{S}_k) \boldsymbol{e}_{k|k} \middle| \mathscr{D}_k\right] \\
&= \mathrm{E}\left[\boldsymbol{x}_k^T \boldsymbol{S}_k \boldsymbol{x}_k \middle| \mathscr{D}_k\right] + \mathrm{tr}\left(\mathbf{E}_{k|k} (\boldsymbol{F}_k - \boldsymbol{S}_k)\right)
\end{aligned} \tag{5.26}$$

5.4 Myopic/Greedy Control

Moving along to the first controller taking future measurements into consideration, these are M-measurement controllers that are greedy or short sighted (myopic). This means that instead of trying to consider the cost all the way to the end of the time horizon they only consider the cost over a much shorter horizon. In effect, this means at time-step k the cost-to-go at time-step $k+m$ is approximated as zero, where m is the horizon length considered. The assumption behind this is that the acquired cost beyond that is negligible compared to the cost acquired before that.

For large m the problem is still hard to solve so a myopic controller with horizon 3 will here be derived. It will further be restricted to the LQ problem studied in this thesis. Start by setting $J_{k+3}^{\mu}(\mathscr{D}_{k+3}) = 0$ and examine the cost-to-go for time-step $k+2$,

$$J_{k+2}^{\mu}(\mathscr{D}_{k+2}) = \mathrm{E}\left[\boldsymbol{x}_{k+2}^T \boldsymbol{Q}_{k+2}^x \boldsymbol{x}_{k+2} + 2\boldsymbol{x}_{k+2}^T \boldsymbol{Q}_{k+2}^{xu} \boldsymbol{u}_{k+2} + \boldsymbol{u}_{k+2}^T \boldsymbol{Q}_{k+2}^u \boldsymbol{u}_{k+2} \middle| \mathscr{D}_{k+2}\right] \tag{5.27}$$

This cost-to-go is minimised by $\boldsymbol{u}_{k+2} = \mathbf{0}$ since the state $\boldsymbol{x}_{k+2}$ is independent of $\boldsymbol{u}_{k+2}$. The optimal cost-to-go is then $J^{\mu*}_{k+2}(\mathscr{D}_{k+2}) = \mathrm{E}\left[\boldsymbol{x}^T_{k+2}\boldsymbol{Q}^x_{k+2}\boldsymbol{x}_{k+2}\middle|\mathscr{D}_{k+2}\right]$. The cost for time-step $k+1$ becomes

$$\begin{aligned} J^{\mu}_{k+1}(\mathscr{D}_{k+1}) = \mathrm{E}\left[\boldsymbol{x}^T_{k+1}\boldsymbol{Q}^x_{k+1}\boldsymbol{x}_{k+1} + 2\boldsymbol{x}^T_{k+1}\boldsymbol{Q}^{xu}_{k+1}\boldsymbol{u}_{k+1} + \boldsymbol{u}^T_{k+1}\boldsymbol{Q}^u_{k+1}\boldsymbol{u}_{k+1}\middle|\mathscr{D}_{k+1}\right] \\ + \mathrm{E}_{\boldsymbol{z}_{k+2}}\left[\mathrm{E}\left[\boldsymbol{x}^T_{k+2}\boldsymbol{Q}^x_{k+2}\boldsymbol{x}_{k+2}\middle|\mathscr{D}_{k+2}\right]\middle|\mathscr{D}_{k+2}\right] \end{aligned} \tag{5.28}$$

This expression should be recognised from (5.8) in the discussion about dual effect and the derivation for optimal control of the last step of the time horizon. It should be no surprise since nothing has really changed other than the horizon length. The optimal cost and control for time-step $k+1$ can be obtained in the same way, leading to the following cost at time-step k

$$\begin{aligned} J^{\mu}_{k}(\mathscr{D}_{k}) = \mathrm{E}\left[\boldsymbol{x}^T_{k}\boldsymbol{Q}^x_{k}\boldsymbol{x}_{k} + 2\boldsymbol{x}^T_{k}\boldsymbol{Q}^{xu}_{k}\boldsymbol{u}_{k} + \boldsymbol{u}^T_{k}\boldsymbol{Q}^u_{k}\boldsymbol{u}_{k} + \boldsymbol{x}^T_{k+1}\boldsymbol{S}^{\mu}_{k+1}\boldsymbol{x}_{k+1}\middle|\mathscr{D}_{k}\right] \\ + \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\mathrm{tr}(\mathbf{E}_{k+1|k+1}\boldsymbol{C}^{\mu}_{k+1})\middle|\mathscr{D}_{k}\right] \end{aligned} \tag{5.29}$$

where

$$\begin{aligned} \boldsymbol{C}^{\mu}_{k+1} &= (\boldsymbol{Q}^{xu}_{k+1} + \boldsymbol{\Phi}^T_{k+1}\boldsymbol{Q}^x_{k+2}\boldsymbol{\Gamma}_{k+1})(\boldsymbol{Q}^u_{k+1} + \boldsymbol{\Gamma}^T_{k+1}\boldsymbol{Q}^x_{k+2}\boldsymbol{\Gamma}_{k+1})^{-1} \\ &\quad \cdot (\boldsymbol{Q}^{xu}_{k+1} + \boldsymbol{\Phi}^T_{k+1}\boldsymbol{Q}^x_{k+2}\boldsymbol{\Gamma}_{k+1})^T \\ \boldsymbol{S}^{\mu}_{k+1} &= \boldsymbol{Q}^x_{k+1} + \boldsymbol{\Phi}^T_{k+1}\boldsymbol{Q}^x_{k+2}\boldsymbol{\Phi}_{k+1} - \boldsymbol{C}^{\mu}_{k+1} \end{aligned} \tag{5.30}$$

The $\boldsymbol{u}_k$ that minimises $J^{\mu}_{k}(\mathscr{D}_k)$ is then the 3-step myopic control policy. Like the CE-policy this might at a first glance look like it is considering more future measurements than it actually does. Even though it solves the minimisation problem exactly for a 3-step horizon it is only a one-measurement feedback policy. The optimal control for the last two steps is completely independent of the future measurement probabilities, thereby making the control only effectively consider the very next measurement.

5.5 M-Measurement Feedback

What follows is the control policy that will serve as basis for the all methods proposed in the next chapter. That policy is the original M-measurement feedback policy described in [Curry, 1970]. It will also be shown that the 3-step myopic control of the previous section is a special case after the introduction of an extra parameter. The basic formulation is simple; given some M, at time-step k approximate the cost at stage $N_M = k+M$, with the OLOF cost (5.26) for the remainder of the time horizon, $J^{MMF}_{N_M}(\mathscr{D}_{N_M}) = J^{OL*}_{N_M}(\mathscr{D}_{N_M})$. For the LQ-problem this becomes

$$\begin{aligned} J^{MMF}_{k}(\mathscr{D}_k) &= \mathrm{E}\left[\boldsymbol{x}^T_{k}\boldsymbol{Q}^x_{k}\boldsymbol{x}_{k} + 2\boldsymbol{x}^T_{k}\boldsymbol{Q}^{xu}_{k}\boldsymbol{u}_{k} + \boldsymbol{u}^T_{k}\boldsymbol{Q}^u_{k}\boldsymbol{u}_{k}\middle|\mathscr{D}_{k}\right] + \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[J^{MMF}_{k+1}(\mathscr{D}_{k+1})\middle|\mathscr{D}_{k}\right] \\ J^{MMF}_{N_M}(\mathscr{D}_{N_M}) &= \mathrm{E}\left[\boldsymbol{x}^T_{N_M}\boldsymbol{S}_{N_M}\boldsymbol{x}_{N_M}\middle|\mathscr{D}_{N_M}\right] + \mathrm{tr}\big(\mathbf{E}_{N_M|N_M}(\boldsymbol{F}_{N_M} - \boldsymbol{S}_{N_M})\big)\big] \end{aligned} \tag{5.31}$$

This modified problem is then minimised and the sequence $\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, ...\boldsymbol{u}_{N_M}$ is found. $\boldsymbol{u}_k$ is used for control of the system and the process is then repeated in the next time-step and so forth. The main problem with this approach is the minimisation over $\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, ...\boldsymbol{u}_{N_M}$. Even though $N_M < N$ this is a hard, high dimensional problem to solve in real-time, even for moderate value of M. The focus of this thesis has therefore been on the special case $M = 1$ and other simple approaches to make this optimisation problem manageable.

5.6 One-Measurement Feedback

With the measurement horizon being $M = 1$ the cost becomes

$$\begin{aligned} J_k^{OMF}(\mathscr{D}_k) = \mathrm{E}\left[\boldsymbol{x}_k^T \boldsymbol{Q}_k^x \boldsymbol{x}_k + 2\boldsymbol{x}_k^T \boldsymbol{Q}_k^{xu} \boldsymbol{u}_k + \boldsymbol{u}_k^T \boldsymbol{Q}_k^u \boldsymbol{u}_k + \boldsymbol{x}_{k+1}^T \boldsymbol{S}_{k+1} \boldsymbol{x}_{k+1} \middle| \mathscr{D}_k\right] \\ + \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\mathrm{tr}(\mathbf{E}_{k+1|k+1}(\boldsymbol{F}_{k+1} - \boldsymbol{S}_{k+1})) \middle| \mathscr{D}_k\right] \end{aligned} \tag{5.32}$$

This expression should be recognised from the cost for the myopic control (5.29) with $\boldsymbol{F}_{k+1} - \boldsymbol{S}_{k+1}$ in place of $\boldsymbol{C}_{k+1}^{\mu}$ and $\boldsymbol{S}_{k+1}$ instead of $\boldsymbol{S}_{k+1}^{\mu}$. If one studies the expressions for $\boldsymbol{S}_{k+1}^{\mu}$ and $\boldsymbol{C}_{k+1}^{\mu}$ they should also be recognised, but this time from the CE and OLOF costs. $\boldsymbol{S}_{k+1}^{\mu}$ is clearly given by the Riccati equation for an open-loop LQ problem with a time-horizon that ends at $k+2$. This should come as no surprise considering the discussion about the last two steps of the myopic control. It was there established that the costs for those steps are independent of future measurements and should therefore be equivalent to the two step open-loop problem. By introducing the following notation

$$\begin{aligned} \boldsymbol{S}_{k:j} &= \boldsymbol{Q}_k^x + \boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1:j-1} \boldsymbol{\Phi}_k - (\boldsymbol{Q}_k^{xu} + \boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1:j-1} \boldsymbol{\Gamma}_k) \\ &\qquad \cdot (\boldsymbol{Q}_k^u + \boldsymbol{\Gamma}_k^T \boldsymbol{S}_{k+1:j-1} \boldsymbol{\Gamma}_k)^{-1} (\boldsymbol{Q}_k^{xu} + \boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1:j-1} \boldsymbol{\Gamma}_k)^T \\ \boldsymbol{S}_{k:0} &= \boldsymbol{Q}_k^x \\ \boldsymbol{F}_{k:j} &= \boldsymbol{\Phi}_k^T \boldsymbol{F}_{k+1:j-1} \boldsymbol{\Phi}_k + \boldsymbol{Q}_k^x \\ \boldsymbol{F}_{k:0} &= \boldsymbol{Q}_k^x \end{aligned} \tag{5.33}$$

together with the parameter N_{LQ} and

$$\begin{aligned} J_{k:N_{LQ}}^{OMF}(\mathscr{D}_k) = \mathrm{E}\left[\boldsymbol{x}_k^T \boldsymbol{Q}_k^x \boldsymbol{x}_k + 2\boldsymbol{x}_k^T \boldsymbol{Q}_k^{xu} \boldsymbol{u}_k + \boldsymbol{u}_k^T \boldsymbol{Q}_k^u \boldsymbol{u}_k + \boldsymbol{x}_{k+1}^T \boldsymbol{S}_{k+1:N_{LQ}-1} \boldsymbol{x}_{k+1} \middle| \mathscr{D}_k\right] \\ + \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\mathrm{tr}(\mathbf{E}_{k+1|k+1}(\boldsymbol{F}_{k+1:N_{LQ}-1} - \boldsymbol{S}_{k+1:N_{LQ}-1})) \middle| \mathscr{D}_k\right] \end{aligned} \tag{5.34}$$

the one-measurement feedback cost can obtained by setting $N_{LQ} = N - k$, i.e. $J_k^{OMF} = J_{k:N-k}^{OMF}$, since $\boldsymbol{C}_{k+1}^{\mu} = \boldsymbol{F}_{k:1} - \boldsymbol{S}_{k:1}$.

This slightly broader, compared to Curry's original policy, class of one-measurement feedback policies with limited open-loop horizon N_{LQ} clearly contains

the 3-step myopic controller with the cost given by

$$J_k^{\mu}(\mathscr{D}_k) = J_{k:2}^{OMF}(\mathscr{D}_k) \tag{5.35}$$

The introduction of the N_{LQ} horizon gives more control to the designer, allowing for a heuristic adaptation of the policy to problems where the open-loop approximation for the entirety of the remaining time is too pessimistic. By reducing the open-loop horizon more weight can be put on time-steps inside the measurement horizon.

This concept of variable open-loop horizons can be extended even further by having different horizons for calculation of $\boldsymbol{S}_{k:j}$ and $\boldsymbol{F}_{k:j}$. In the derivation of the open-loop cost, the state was split into two parts, mean and error, both having linear dynamics. It was shown that $\boldsymbol{S}_{k:j}$ is the weight put on the mean-dynamics while $\boldsymbol{F}_{k:j}$ corresponds to the error dynamics. The problem in the closed-loop case is that both the mean and error dynamics no longer are linear, if they were, the CE/OLOF policies would be optimal, and how well these dynamics are approximated by the resulting linear open-loop dynamics determines how good the OLOF approximation is. Important to note is that even though the mean dynamics might be well approximated by the linear system, the error dynamics might not or vice versa. Being able to control the relative weight between $\boldsymbol{S}_{k:j}$ and $\boldsymbol{F}_{k:j}$ then gives the designer the possibility to put more weight on the dynamics that are well approximated while putting less weight on the bad one. This is achieved by the introduction of a second horizon N_E

$$\begin{aligned} J_{k:N_{LQ}:N_E}^{OMF}(\mathscr{D}_k) = \mathrm{E}\left[\boldsymbol{x}_k^T\boldsymbol{Q}_k^x\boldsymbol{x}_k + 2\boldsymbol{x}_k^T\boldsymbol{Q}_k^{xu}\boldsymbol{u}_k + \boldsymbol{u}_k^T\boldsymbol{Q}_k^u\boldsymbol{u}_k + \boldsymbol{x}_{k+1}^T\boldsymbol{S}_{k+1:N_{LQ}-1}\boldsymbol{x}_{k+1}\middle|\mathscr{D}_k\right] \\ + \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\mathrm{tr}(\mathbf{E}_{k+1|k+1}(\boldsymbol{F}_{k+1:N_E-1} - \boldsymbol{S}_{k+1:N_{LQ}-1}))\middle|\mathscr{D}_k\right] \end{aligned} \tag{5.36}$$

The concept can be expanded further, making it possible to completely discard one or both of the mean/error dynamics by setting N_{LQ} and/or N_E to zero and defining

$$\begin{aligned} \boldsymbol{S}_{k:-1} &= \mathbf{0} \\ \boldsymbol{F}_{k:-1} &= \mathbf{0} \end{aligned} \tag{5.37}$$

Even with the introduction of these extra parameters it should be noted that all that's changed is the weighting constants, all of which can be calculated offline, making it the same as Curry's original one-measurement feedback from a problem point of view. The same solver can therefore be used for the minimisation problem regardless of k, N_{LQ} and N_E. The choice of N_{LQ} and N_E will however affect the behaviour of the controller. Due to this fact that N_{LQ} and N_E doesn't affect the problem formulation in any way the explicit dependency on them will no longer be written for convenience sake. From now on, including in the not yet introduced M-measurement cost feedback, everywhere $\boldsymbol{F}_{k+1}$ and $\boldsymbol{S}_{k+1}$ appears, they can be replaced by $\boldsymbol{F}_{k+1:N_E-1}$ and $\boldsymbol{S}_{k+1:N_{LQ}-1}$ respectively to give parametric control over the open-loop horizons.

With $N_{LQ} = N_E$ it can be shown that the resulting $(\boldsymbol{F}_{k+1:N_E-1} - \boldsymbol{S}_{k+1:N_{LQ}-1})$ is positive semi-definite [Curry, 1970]. This has the effect that the last term of $J^{OMF}_{k:N_{LQ}:N_E}(\mathscr{D}_k)$ has a minimum and it's achieved when the error $\boldsymbol{e}_{k+1|k+1}$ is minimised. This means that the controller might make actions that drives the mean of $\boldsymbol{x}_{k+1}$ further from the origin in order to reduce the error; this kind of behaviour is called probing. However, this is not true for all N_E and at least for one $N_E < N_{LQ}$ the matrix will become negative semi-definite, making the last term of $J^{OMF}_{k:N_{LQ}:N_E}(\mathscr{D}_k)$ larger when the error is reduced. This means that the controller might drive the mean further from the origin to make the error worse and thereby reducing the cost. This is undesirable behaviour in the vast majority of cases and the open-loop horizons should therefore be selected with care. Nonetheless, in some cases it might still be necessary to choose a $N_E < N_{LQ}$.

It's well known that $\boldsymbol{S}_k$ has a stationary solution if the process is controllable, but the same can not be said for $\boldsymbol{F}_k$, which might grow forever. Intuitively this makes sense since in an open-loop scenario the error grows with time and so should the associated cost. This causes problem with the open-loop approximation when the original problem has a long, or even infinite, horizon. In such cases $\boldsymbol{F}_k$ would grow to such an extent that all the cost associated with the mean is completely insignificant. The policy would then only focus on minimising the error, something that's also undesirable in most cases. The designer will in these cases be needed to choose a smaller N_E and it might perhaps even make sense to choose a smaller N_{LQ} as well. It will usually be easier to motivate a smaller N_E since in most cases the act of measuring will place some bound on the error, making it reasonable to put the restriction on the error dynamics. An example of this will be given in the next chapter.

5.7 M-Measurement Cost Feedback

The one-measurement feedback was introduced since the full M-measurement feedback problem is too expensive to solve, and the following M-measurement cost feedback is introduced for the same reason. Instead of simplifying the problem by reducing the number of measurement considered, it reduces the complexity of the space over admissible controls. By restricting the future control actions, up to the time-step $k+M$ to only depend on the current state $\boldsymbol{x}_k$, they are independent of the future measurements and the M-measurement feedback cost in (5.31) can be simplified to

$$\begin{aligned} J^{MMC}_k(\mathscr{D}_k) = \mathrm{E}\left[\boldsymbol{x}^T_{k+M}\boldsymbol{S}_{k+M}\boldsymbol{x}_{k+M} + \sum_{t=k}^{k+M-1} \boldsymbol{x}^T_t\boldsymbol{Q}^x_t\boldsymbol{x}_t + 2\boldsymbol{x}^T_t\boldsymbol{Q}^{xu}_t\boldsymbol{u}_t + \boldsymbol{u}^T_t\boldsymbol{Q}^u_t\boldsymbol{u}_t \middle| \mathscr{D}_k\right] \\ + \mathrm{E}_{\boldsymbol{Z}_{k+1:k+M}}\left[\mathrm{tr}(\mathbf{E}_{k+M|k+N}(\boldsymbol{F}_{k+M} - \boldsymbol{S}_{k+M}))\middle| \mathscr{D}_k\right] \end{aligned} \tag{5.38}$$

Where the second expected value is taken over all measurements between $k+1$ and $k+M$. The reason for the name "M-measurement cost feedback´´ can be seen here. Although it's not a true M-measurement feedback since the control policies considered are fixed and can not adapt to the future measurement, the measurements are still considered when calculating the cost-to-go.

This method has similarities with the general one-measurement feedback presented in [Curry, 1970] were the measurement considered could be placed at any time after time-step k, not only at $k+1$ which is the case presented earlier. However, in that method the system was assumed to run in open loop up until that measurement while in this method the full cost of the system dynamics are considered. The main advantage of this is the possibility of capturing non-linear effects of the system with dynamics too slow to affect the cost over a one-step horizon.

It's clearly seen that the first term of $J_k^{MMC}(\mathscr{D}_k)$ is the same as the OLOF cost for a problem with horizon M so by using the same method of splitting the state into estimate and estimate error, $\boldsymbol{x}_t = \hat{\boldsymbol{x}}_{t|k} + \boldsymbol{e}_{t|k}$, the following rearrangement can be made.

$$\begin{aligned} J_k^{MMC}(\mathscr{D}_k) = \sum_{t=k}^{k+M-1} \hat{\boldsymbol{x}}_{t|k}^T \boldsymbol{Q}_t^x \hat{\boldsymbol{x}}_{t|k} + 2\hat{\boldsymbol{x}}_{t|k}^T \boldsymbol{Q}_t^{xu} \boldsymbol{u}_t + \boldsymbol{u}_t^T \boldsymbol{Q}_t^u \boldsymbol{u}_t + \mathrm{E}\left[\boldsymbol{e}_{t|k}^T \boldsymbol{Q}_t^x \boldsymbol{e}_{t|k} \middle| \mathscr{D}_k\right] \\ + \hat{\boldsymbol{x}}_{k+M|k}^T \boldsymbol{S}_{k+M} \hat{\boldsymbol{x}}_{k+M|k} + \mathrm{E}\left[\boldsymbol{e}_{k+M|k}^T \boldsymbol{S}_{k+M} \boldsymbol{e}_{k+M|k} \middle| \mathscr{D}_k\right] \\ + \mathrm{E}_{\boldsymbol{Z}_{k+1:k+M}}\left[\mathrm{tr}(\mathbf{E}_{k+M|k+N}(\boldsymbol{F}_{k+M} - \boldsymbol{S}_{k+M})) \middle| \mathscr{D}_k\right] \end{aligned} \tag{5.39}$$

From before it's known that the terms containing the errors are independent of the control and can therefore be discarded, resulting in the final equivalent form:

$$\begin{aligned} J_k^{MMC}(\mathscr{D}_k) = \hat{\boldsymbol{x}}_{k+M|k}^T \boldsymbol{S}_{k+M} \hat{\boldsymbol{x}}_{k+M|k} + \sum_{t=k}^{k+M-1} \hat{\boldsymbol{x}}_{t|k}^T \boldsymbol{Q}_t^x \hat{\boldsymbol{x}}_{t|k} + 2\hat{\boldsymbol{x}}_{t|k}^T \boldsymbol{Q}_t^{xu} \boldsymbol{u}_t + \boldsymbol{u}_t^T \boldsymbol{Q}_t^u \boldsymbol{u}_t \\ + \mathrm{E}_{\boldsymbol{Z}_{k+1:k+M}}\left[\mathrm{tr}(\mathbf{E}_{k+M|k+N}(\boldsymbol{F}_{k+M} - \boldsymbol{S}_{k+M})) \middle| \mathscr{D}_k\right] \end{aligned} \tag{5.40}$$

5.8 Alternative Approaches and Prior Research

The control methods presented and examined in this thesis are all fairly direct in the way that they tackle the dynamic programming problem directly, making the consideration of dual effect and probing action implicitly included in the control problem. A more traditional approach is to more explicitly include this dual effect into the control problem with the help of self-tuning regulators and other adaptive control schemes. Other than getting a broad overview of the adaptive control subject, [Åström, 1983; Wittenmark, 2002; Åström and Wittenmark, 2008], little time

was spent researching these methods for the integration of the particle filter. The reason for this was that other approaches were judged to make more interesting use of the particle filter's information state. With that said, the adaptive controllers would all still benefit from the improved state and variance estimates that the particle filter would provide.

Another approach not related to the solving of the dynamic programming problem that received a little more time was different ways to make simple approximations of the future error dynamics. These approximations can be gotten in a variety of ways, from simple heuristic ad-hoc methods to more complex system identifications and machine learning approaches, resulting in more or less complex models. The reason for these techniques not ending up in this thesis is that either the resulting models were to complex to allow for simple solutions of the dynamic programming problem or not complex enough to capture the interesting non-linearities. However, in a way ended one of these approximative methods up getting included by way of the OLOF control. In OLOF control the future error dynamics can be said to be approximated by the open-loop dynamics. Also, promise still remains for machine learning approaches with the learning being directed directly on the actual cost-to-go function instead of the system dynamics.

Like previously stated, the methods presented in this thesis are direct in the way that they try to solve the dynamic programming problem at each time-step but due to the cost to solve the full equation, the problem solved is modified. How well they perform is therefore directly dependent on how well the modified problem approximates the original problem. The vast majority of time was therefore spent on researching alternative approaches for solving the full problem cheaply, focusing on two different concepts.

The first concept is the one of iterative rollout solvers [Bertsekas, 2005]. They all work from an initial guess of the optimal control sequence and after calculation of the resulting cost, in some way modifying the control sequence to achieve better cost. By repeating this process over and over again, better and better solutions are achieved and eventually it converges to a (locally) optimal solution. The approach for modification of the control sequence usually involves to around the resulting trajectory make a local approximation of the problem and solving it for the new sequence. The approximation can be a quadratic approximation of the entire cost function [Rajamäki et al., 2016] or making a linear approximation of the dynamics and quadratic approximation of the immediate cost to form a locally valid LQR-problem [Li and Todorov, 2004].

This is still expensive due to the need to simulate to end time in order to calculate the cost but has the advantage of being able to be aborted early, before convergence is achieved, and still provide a control better than the initial guess. Another prob-

lem is that these methods are usually formulated for deterministic problems. The LQR-variant has been expanded to include stochastic dynamics [Todorov and Li, 2005; Berg, 2016] but when only partial information about the system is available through stochastic measurements the challenge is how and around what trajectory the quadratic approximation should be made. In [Van Den Berg et al., 2012; Sun et al., 2016] the problem was addressed by making a linear approximation of the error dynamics with the help of an extended Kalman filter but in the problem of quantised measurement, due to the measurement function being a step function with derivative zero, this would result in an open loop approximation with poor performance. In the particle filter context, even with early termination of the iterations, the cost would be a real concern but with that being said, some preliminary work done has shown some promise for an iterative particle filter approach with local quadratic approximation.

The second concept revolves around a general duality between estimation and control [Todorov, 2008]. The main reason for it's appeal is the fact that the particle filter is an iterative optimal estimator so with a reformulation of the control problem to an estimation problem the particle filter itself might be used. This general duality has also given rise to a new family of linearly solvable control problems [Dvijotham and Todorov, 2012] and resulting solution strategies [Todorov, 2009b; Toussaint, 2009; Todorov, 2009a] but once again, although they include stochastic dynamics, they couldn't easily be converted to include systems with partially observed states.

The concept of formulating the control problem as an estimation problem together with a particle filter was utilised in [Stahl and Hauth, 2011] but it doesn't utilise the duality. Instead it directly specifies it's cost function in terms of Gaussian probability densities for the state, centred around the desired state, these densities could after that be used to filter "most likely" control signal. The connection to the original cost function is therefore lost so this together with the late discovery resulted in that the methods were not further explored. The use of particle filter in the dynamic programming/optimal control setting does not seem to be that common so beside this, the other real use has been in [Bayard and Schumitzky, 2010] where it's only use was for cost evaluation. The optimisation method used was just an direct recursive strategy, numerically finding the optimal control at each time step, making the proposed control strategy unsuited for all but the simplest discrete control problems.

6

Feedback Design

The attention will here be turned completely to the linear dynamics with quadratic cost problem studied in this thesis. The goal is to introduce the particle filter into the methods presented in Chapter 5 and develop algorithms for the minimisation problems. The necessary details for an implementation of the algorithms will be provided as well as any process specific details and concerns that need to be dealt with.

First the OLOF/CE control will be handled, the particle filter integration is in that case simple but instead system specific issues regarding controllability will have to be handled. After that, a particle filter based algorithm for use in the one-measurement feedback control will be derived. Two different methods for evaluation of the cost-to-go function will be presented for use in that algorithm. One of these algorithms will be expanded to cover cost-to-go of the M-measurement cost feedback and two different variants of the policy will be presented.

6.1 Certainty Equivalent Control

From (5.16) and (5.17), the CE/OLOF control at time step k is

$$\begin{aligned}
\boldsymbol{u}_k^{CE} =& \boldsymbol{L}_k \hat{\boldsymbol{x}}_{k|k} \\
\boldsymbol{L}_k =& -(\boldsymbol{Q}_k^u + \boldsymbol{\Gamma}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k)^{-1} (\boldsymbol{Q}_k^{xu} + \boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k)^T \\
\boldsymbol{S}_k =& \boldsymbol{Q}_k^x + \boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Phi}_k \\
& - (\boldsymbol{Q}_k^{xu} + \boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k)(\boldsymbol{Q}_k^u + \boldsymbol{\Gamma}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k)^{-1} (\boldsymbol{Q}_k^{xu} + \boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k)^T \\
\boldsymbol{S}_N =& \boldsymbol{Q}_N^x
\end{aligned} \tag{6.1}$$

This is trivially integrated with the particle filter since it gives an approximation of $p(\boldsymbol{x}_k|\mathscr{D}_k)$. With the notation of Chapter 2, where $\mathscr{D}_k = \boldsymbol{D}_k$, the expression for the

conditional expected value is given by

$$p(\boldsymbol{x}_k|\mathscr{D}_k) \approx \widehat{p}(\boldsymbol{x}_k|\boldsymbol{D}_k) = \sum_{i=1}^{N} \omega_k^i \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^i)$$
$$\hat{\boldsymbol{x}}_{k|k} = \sum_{i=1}^{N} \omega_k^i \boldsymbol{x}_k^i \tag{6.2}$$

where $\widehat{p}(\boldsymbol{x}_k|D_k)$ is provided by the particle filter. All that's needed to apply CE-control on the process is to calculate $\boldsymbol{S}_k$ and $\boldsymbol{L}_k$, but here is where the problem lies. The main focus of this thesis will be on the case where the time horizon, N, is infinity but $\boldsymbol{S}_k$ can not be guaranteed to converge when $N \to \infty$, due to the system not being controllable. However, it can be shown, sketch of proof in Appendix C.1, that although $\boldsymbol{S}_k$ doesn't converge, $\boldsymbol{L}_k$ does. This means that although the future cost might grow forever the optimal control action remains the same when considering longer and longer time horizons. In practice this means that when the infinite horizon case is considered, $\boldsymbol{S}_k$ is simply iterated over until $\boldsymbol{L}_k$ has converged to some $\boldsymbol{L}$. $\boldsymbol{L}$ is then used in the CE control law, $\boldsymbol{u}_k^{CE} = \boldsymbol{L}\hat{\boldsymbol{x}}_{k|k}$, at each time-step.

6.2 One-Measurement Feedback

When applying one-measurement feedback to the process, the fact that $\boldsymbol{S}_k$ doesn't converge remains a problem, but, as already stated, $\boldsymbol{F}_k$ doesn't converge either and the open loop horizons, N_{LQ} and N_E, need to be restricted regardless. This isn't a real problem since the open loop cost is an approximation in the first place and changing the horizons simply allows for it to more closely match the real cost. The open loop horizon lengths should therefore be seen as control parameters that need tuning to perform optimally.

As an indication for what reasonable choices would be, it is useful to look at the behaviour of $\boldsymbol{S}_k$ as $N_{LQ} \to \infty$. With $\boldsymbol{S}_k$ partitioned into blocks corresponding to the controllable states, the uncontrollable states and cross-terms between them, it's known from Appendix C.1 that the blocks corresponding to the controllable states and the cross-terms converge. How fast that happens serves as an indication for a suitable time horizon since after that time the ratio between the costs associated with uncontrollable and controllable state will continue to grow to infinity. This phenomenon should not be present in the real system since quantised measurements provide an upper boundary on the state variance.

Using the block notation of Appendix C.1, with $\boldsymbol{S}_k^m$ corresponding to the controllable states, $\boldsymbol{S}_k^d$ to the uncontrollable and $\boldsymbol{S}_k^{md}$ containing the cross-terms, the development of $\boldsymbol{S}_k$ for the studied system as N_{LQ} grows can be seen in Figure 6.1.

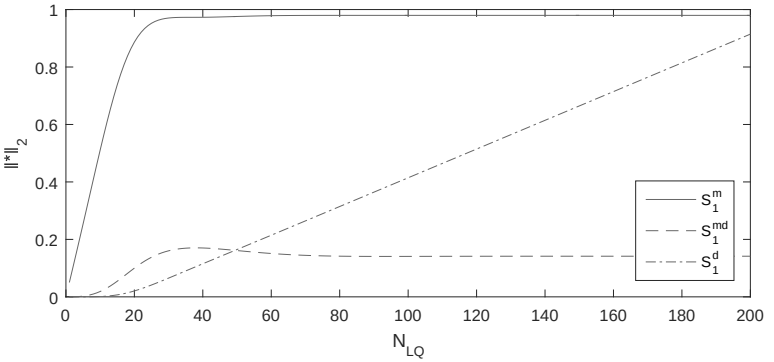

Figure 6.1 2-norm of the partitions of $\boldsymbol{S}_0$ as a function N_{LQ}

From the figure it's seen that with a N_{LQ} horizon of around 80 the converging partitions have done so, with the majority of the development happening for $N_{LQ} < 30$. For $N_{LQ} > 80$ more and more weight will be put on the uncontrollable states so a suitable horizon length should be within this rough range of $50 < N_{LQ} < 150$.

It's also useful to look at the open-loop error dynamics since there is a separate parameter controlling the open-loop error horizon. Given the system in (1.16) an uncertainty bound can be put on $\boldsymbol{y}_k$. With the same reasoning as for the additive noise model, given a single measurement, $\boldsymbol{y}_k$ will with no other information be uniformly distributed in the quantisation interval. This worst case can be used to form an upper bound on the variance, $\text{Var}[\boldsymbol{y}_k] \leq \frac{\Delta q^2}{12}$, which in turn can be used to give an approximate bound on the open loop error horizon.

Given a state variance at time-step k, $\mathbf{E}_{k|k}$, the open loop variance of $\boldsymbol{y}_{t|k}$ for some time $t > k$ will be:

$$\text{Var}[\boldsymbol{y}_{t|k}] = \text{Var}[\boldsymbol{C}\boldsymbol{x}_{t|k}] = \boldsymbol{C}\mathbf{E}_{t|k}\boldsymbol{C}^T \leq \frac{\Delta q^2}{12} \tag{6.3}$$

For t larger than some d this inequality will no longer hold, and that d then gives the open-loop horizon $N_E = d - k$. N_E is found by simply using the open loop error dynamics in (5.23) and iterating until the bound is exceeded.

$$\boldsymbol{e}_{t|k} = \boldsymbol{\Phi}_{t:k}\boldsymbol{e}_{k|k} + \sum_{i=k}^{t-1} \boldsymbol{\Phi}_{t:i+1}\boldsymbol{\nu}_i \tag{6.4}$$

gives

$$\text{Var}[\boldsymbol{y}_t|\mathscr{D}_k] = \boldsymbol{C}\boldsymbol{\Phi}_{t:k}\mathbf{E}_{k|k}\boldsymbol{\Phi}_{t:k}^T\boldsymbol{C}^T + \sum_{i=k}^{t-1} \boldsymbol{\Phi}_{t:i+1}\widehat{\boldsymbol{R}}_k\boldsymbol{\Phi}_{t:i+1}^T \leq \frac{\Delta q^2}{12} \tag{6.5}$$

It's clear that the resulting horizon, N_E, depends on the state variance $\mathbf{E}_{k|k}$. Although it's possible to expand the one-measurement feedback to recalculate N_E based on the current $\mathbf{E}_{k|k}$, for simplicity this will not be done. Instead, a system with perfect knowledge, $\mathbf{E}_{k|k} = \mathbf{0}$, will be used to calculate N_E to serve as an indication of a suitable horizon length. For the system studied this gives $N_E = 103$, which is roughly the same as the indications given for N_{LQ} and to keep the number of variables down they will not be varied separately and instead always be set to the same open loop horizon N_{OL}.

$$N_{LQ} = N_E = N_{OL} \tag{6.6}$$

With these rough indications of suitable open-loop horizons established the attention can be turned to minimisation algorithms. The cost function that should be minimized at each time step has the form

$$\begin{aligned} J_k^{OMF}(\mathscr{D}_k) = \mathrm{E}\left[\boldsymbol{x}_k^T \boldsymbol{Q}_k^x \boldsymbol{x}_k + 2\boldsymbol{x}_k^T \boldsymbol{Q}_k^{xu} \boldsymbol{u}_k + \boldsymbol{u}_k^T \boldsymbol{Q}_k^u \boldsymbol{u}_k + \boldsymbol{x}_{k+1}^T \boldsymbol{S}_{k+1} \boldsymbol{x}_{k+1} \middle| \mathscr{D}_k\right] \\ + \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\mathrm{E}\left[\boldsymbol{e}_{k+1|k+1}^T \boldsymbol{C}_{k+1} \boldsymbol{e}_{k+1|k+1} \middle| \mathscr{D}_{k+1}\right] \middle| \mathscr{D}_k\right] \end{aligned} \tag{6.7}$$

For the sake of computational convenience some restructuring of the last term will be made. First note that $\hat{\boldsymbol{x}}_{k+1|k+1} = \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k+1} + \boldsymbol{\Gamma}_k \boldsymbol{u}_k$ which allows the error $\boldsymbol{e}_{k+1|k+1}$ to be written as

$$\begin{aligned} \boldsymbol{e}_{k+1|k+1} =& \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k \boldsymbol{u}_k + \boldsymbol{\nu}_k - \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k+1} - \boldsymbol{\Gamma}_k \boldsymbol{u}_k \\ =& \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\nu}_k - \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k+1} \end{aligned} \tag{6.8}$$

which with the assumption of zero mean process noise $\boldsymbol{\nu}_k$ gives

$$\begin{aligned} &\mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\mathrm{E}\left[\boldsymbol{e}_{k+1|k+1}^T \boldsymbol{C}_{k+1} \boldsymbol{e}_{k+1|k+1} \middle| \mathscr{D}_{k+1}\right] \middle| \mathscr{D}_k\right] \\ &= \mathrm{E}\left[\boldsymbol{x}_k^T \boldsymbol{\Phi}_k^T \boldsymbol{C}_{k+1} \boldsymbol{\Phi}_k \boldsymbol{x}_k \middle| \mathscr{D}_k\right] - \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\hat{\boldsymbol{x}}_{k|k+1}^T \boldsymbol{\Phi}_k^T \boldsymbol{C}_{k+1} \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k+1} \middle| \mathscr{D}_k\right] \end{aligned} \tag{6.9}$$

Since $\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k \boldsymbol{u}_k + \boldsymbol{\nu}_k$, and since the distribution of $\boldsymbol{x}_k | \mathscr{D}_k$ is independent of $\boldsymbol{u}_k$ an equivalent cost can be formed as

$$\begin{aligned} J_k^{OMF}(\mathscr{D}_k) = 2\hat{\boldsymbol{x}}_{k|k}^T (\boldsymbol{Q}_k^{xu} + \boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k) \boldsymbol{u}_k + \boldsymbol{u}_k^T (\boldsymbol{Q}_k^u + \boldsymbol{\Gamma}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k) \boldsymbol{u}_k \\ - \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\hat{\boldsymbol{x}}_{k|k+1}^T \boldsymbol{\Phi}_k^T \boldsymbol{C}_{k+1} \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k+1} \middle| \mathscr{D}_k\right] \end{aligned} \tag{6.10}$$

Finding the $\boldsymbol{u}_k^*$ that minimises this expression then gives the one-measurement feedback control. For the system studied, this is a one dimensional problem and direct search can and will then be utilised even though both first and second order derivatives of $J_k^{OMF}(\mathscr{D}_k)$ could be extracted from the particle filter. Due to the sampled nature of the particle filter the derivatives calculated are not of high enough quality to be used directly without some form of line-search or relaxation. The increase of convergence speed was not deemed large enough to compensate for the significant

increase in evaluation costs. Using direct methods allowed for the search strategy used here to be directly applied to the M-measurement cost feedback presented in the next section.

To efficiently find $\boldsymbol{u}_k^*$, the focus lied on cheap evaluation of $J_k^{OMF}(\mathscr{D}_k)$, or more precisely the last term since the first two terms are easily evaluated given the estimate of $\hat{\boldsymbol{x}}_{k|k}^T$ given by the particle filter. Two probability distributions are needed for evaluation, $p(\boldsymbol{z}_{k+1}|\mathscr{D}_k)$ and $p(\boldsymbol{x}_k|\mathscr{D}_{k+1})$, both of which can be approximated with the particle filter. For clarity, when talking about a particle filter that should be run into the future, a standard SIR particle filter is meant, regardless of what filter type that was used to generate the state estimate.

Starting with the smoothing density, $p(\boldsymbol{x}_k|\mathscr{D}_{k+1})$, it can be obtained by running the particle filter forward in time, storing the particles $\boldsymbol{x}_k^i$ at time-step k, and computing the particle weights at time-step $k+1$, $\boldsymbol{\omega}_{k+1}^i$. The smoothing density is then approximated by

$$\begin{aligned} p(\boldsymbol{x}_k|\mathscr{D}_{k+1}) &\approx \widehat{p}(\boldsymbol{x}_k|D_{k+1}) = \sum_{i=1}^{N} \boldsymbol{\omega}_{k+1}^i \boldsymbol{\delta}(\boldsymbol{x}_k - \boldsymbol{x}_k^i) \\ \hat{\boldsymbol{x}}_{k|k+1} &\approx \sum_{i=1}^{N} \boldsymbol{\omega}_{k+1}^i \boldsymbol{x}_k^i \end{aligned} \tag{6.11}$$

A few things should be noted here. Because of the resampling, the particles $\boldsymbol{x}_k^i$ need not have a corresponding weight $\boldsymbol{\omega}_{k+1}^i$ if they were discarded during the resampling-step. This is a well known problem when trying to use particle filters in smoothing problems so for that reason, and a few more that will be apparent soon, the resampling step will be omitted when running the filter into the future. This effectively sets a maximum length into the future the filter can be run until the particle density isn't high enough to give good estimate. However, since it for now only needs to be run one step ahead this will not be a problem.

In regards to the calculation of the probability of future measurements, $p(\boldsymbol{z}_{k+1}|\mathscr{D}_k)$, remember that the particle weights of the particle filter are normalised. The normalisation factor is the total probability of that measurement happening, resulting in

$$p(\boldsymbol{z}_{k+1}|\mathscr{D}_k) \approx \widehat{p}(\boldsymbol{z}_{k+1}|\boldsymbol{D}_k) = \sum_{i=1}^{N} p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}^i)\boldsymbol{\omega}_k^i \tag{6.12}$$

Since $\boldsymbol{z}_k$ is discrete, $\mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\hat{\boldsymbol{x}}_{k|k+1}^T \boldsymbol{\Phi}_k^T \boldsymbol{C}_{k+1} \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k+1} \middle| \mathscr{D}_k\right]$ can be calculated for a given $\boldsymbol{u}_k$ by, for each $\boldsymbol{z}_k$, running the particle filter one step ahead, calculating $\hat{\boldsymbol{x}}_{k|k+1}^T \boldsymbol{\Phi}_k^T \boldsymbol{C}_{k+1} \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k+1} \widehat{p}(\boldsymbol{z}_{k+1}|\boldsymbol{D}_k)$ for that $\boldsymbol{z}_k$ and adding all results together. This approach has a couple of problems, the first being that there are infinitely many

possible $\boldsymbol{z}_k$. Due to the quantisation intervals being assumed to be large, the effect of the measurement intervals far away from the current state have little effect on future states. For that reason, only the three intervals around the zero-reference were ever considered. One further simplification was made, the upper and lower most intervals of these three were assumed to not have any upper and lower boundary respectively. This simplifies some of the calculations and had no effect on the result since the particle cloud never interacted with the removed boundaries in the forward simulation.

The second problem is that it is computationally expensive. Even when only considering the three closest intervals it means that the entire particle system needs to be simulated forward in time three times for every control input $\boldsymbol{u}_k$ that needs to be evaluated when minimizing the cost. This can be avoided so only one state prediction can be made for all cost evaluations by seeing that the control is additive to the state and that

$$p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}+\boldsymbol{\Gamma}_k\boldsymbol{u}_k) = p(\boldsymbol{z}_{k+1}-\boldsymbol{C}\boldsymbol{\Gamma}_k\boldsymbol{u}_k|\boldsymbol{x}_{k+1}) \tag{6.13}$$

Instead of adding the control to the state, the state can be predicted with zero control and the quantisation intervals can be adjusted, to give the same effect on the particle weights. This way all the predicted particles can be stored and not needed to be re-predicted for each $\boldsymbol{u}_k$. The particle weights still need to be recalculated for each $\boldsymbol{u}_k$ and $\boldsymbol{z}_k$ but since all quantisation intervals are mutually exclusive, each particle only needs to be checked once for which interval it's in. With this batch processing of $\boldsymbol{z}_k$, the resulting algorithm can be found in Algorithm 6.1. Note that it was used that $\boldsymbol{\Phi}_k\hat{\boldsymbol{x}}_{k|k+1} = \hat{\boldsymbol{x}}^0_{k+1|k+1}$, using the notation in Algorithm 6.1. With this the total cost can easily be calculated and any standard direct solver be used, in this thesis MATLAB's *fminbnd* was used that utilises golden section search and parabolic interpolation.

However, the discontinuous nature of the particle filter approximation causes some problems since the resulting cost-to-go approximation also will be discontinuous. The discontinuities are small enough not to cause any major issues but certain allowances still need to be made. For example, the minimum value of $\mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\hat{\boldsymbol{x}}^T_{k|k+1}\boldsymbol{\Phi}_k^T\boldsymbol{C}_{k+1}\boldsymbol{\Phi}_k\hat{\boldsymbol{x}}_{k|k+1}\middle|\mathscr{D}_k\right]$ should vary smoothly between time-steps. Due to particle resampling and the random nature of the particle filter, the minimum values of the resulting approximation will jitter around the true minima. This causes problem with the optimal control jumping from being close to the minimum of $\mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\hat{\boldsymbol{x}}^T_{k|k+1}\boldsymbol{\Phi}_k^T\boldsymbol{C}_{k+1}\boldsymbol{\Phi}_k\hat{\boldsymbol{x}}_{k|k+1}\middle|\mathscr{D}_k\right]$ at k and close to the minimum of $2\hat{\boldsymbol{x}}^T_{k|k}(\boldsymbol{Q}^{xu}_k+\boldsymbol{\Phi}_k^T\boldsymbol{S}_{k+1}\boldsymbol{\Gamma}_k)\boldsymbol{u}_k+\boldsymbol{u}_k^T(\boldsymbol{Q}^u_k+\boldsymbol{\Gamma}_k^T\boldsymbol{S}_{k+1}\boldsymbol{\Gamma}_k)\boldsymbol{u}_k$ at $k+1$ or vice versa. The problem gets worse since the expected value isn't convex in $\boldsymbol{u}_k$ and has several local minima, providing even more different places the control can settle in. This all results in a control signal that jitters and a couple approaches were used to combat

Algorithm 6.1: One-measurement Feedback - Cost Evaluation

Data: System matrices, $\{\boldsymbol{\Phi}_k, \boldsymbol{\Gamma}_k, \boldsymbol{C}, \boldsymbol{C}_{k+1}, \Delta q\}$, N particles and weights from a particle filter estimate of the state at time-step k, $\{\omega_k^i, \boldsymbol{x}_k^i\}$, open loop, zero control prediction of these particles $\boldsymbol{x}_{k+1}^{0,i} \sim p_{\boldsymbol{\nu}_k}(\boldsymbol{x}_{k+1} - \boldsymbol{\Phi}_k \boldsymbol{x}_k^i)$, the linear measurement of the predicted particles, $y_{k+1}^i = \boldsymbol{C}\boldsymbol{x}_{k+1}^{0,i}$, and the control for which to evaluate the cost, $\boldsymbol{u}_k$

Result: $J_{uc} \approx \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\hat{\boldsymbol{x}}_{k|k+1}^T \boldsymbol{\Phi}_k^T \boldsymbol{C}_{k+1} \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k+1} \middle| \mathscr{D}_k\right]$

begin

```
/* Initialize accumulating variables for each
   interval                                          */
```

$\boldsymbol{x}_s^u \leftarrow \boldsymbol{0}, \quad \boldsymbol{x}_s^m \leftarrow \boldsymbol{0}, \quad \boldsymbol{x}_s^l \leftarrow \boldsymbol{0}$

$p_z^u \leftarrow 0, \quad p_z^m \leftarrow 0, \quad p_z^l \leftarrow 0$

```
/* Displace the interval boundaries according to
   control                                           */
```

$z^u \leftarrow \frac{\Delta q}{2} - \boldsymbol{C}\boldsymbol{\Gamma}_k \boldsymbol{u}_k, \quad z^l \leftarrow -\frac{\Delta q}{2} - \boldsymbol{C}\boldsymbol{\Gamma}_k \boldsymbol{u}_k$

foreach y_{k+1}^i **do**

```
    /* Find which interval each particle lies in
       and add to accumulative sums                  */
```

    if $y_{k+1}^i > z^u$ **then**

$\boldsymbol{x}_s^u \leftarrow \boldsymbol{x}_s^u + \omega_k^i \boldsymbol{x}_{k+1}^{0,i}, \quad p_z^u \leftarrow p_z^u + \omega_k^i$

    else if $y_{k+1}^i < z^l$ **then**

$\boldsymbol{x}_s^l \leftarrow \boldsymbol{x}_s^l + \omega_k^i \boldsymbol{x}_{k+1}^{0,i}, \quad p_z^l \leftarrow p_z^l + \omega_k^i$

    else

$\boldsymbol{x}_s^m \leftarrow \boldsymbol{x}_s^m + \omega_k^i \boldsymbol{x}_{k+1}^{0,i}, \quad p_z^m \leftarrow p_z^m + \omega_k^i$

    end

end

```
/* Calculate state mean for each quantisation
   interval                                          */
```

$\widehat{\boldsymbol{x}}^u \leftarrow \frac{\boldsymbol{x}_s^u}{p_z^u}, \quad \widehat{\boldsymbol{x}}^l \leftarrow \frac{\boldsymbol{x}_s^l}{p_z^l}, \quad \widehat{\boldsymbol{x}}^m \leftarrow \frac{\boldsymbol{x}_s^m}{p_z^m}$

```
/* Add together total cost                           */
```

$J_{uc} \leftarrow (\widehat{\boldsymbol{x}}^u)^T \boldsymbol{C}_{k+1} \widehat{\boldsymbol{x}}^u \, p_z^u + (\widehat{\boldsymbol{x}}^l)^T \boldsymbol{C}_{k+1} \widehat{\boldsymbol{x}}^l \, p_z^l + (\widehat{\boldsymbol{x}}^m)^T \boldsymbol{C}_{k+1} \widehat{\boldsymbol{x}}^m \, p_z^m$

end

this.

The first approach involves controlling the search intervals. If the optimal control has been found in the vicinity of a local minimum of $\mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\hat{\boldsymbol{x}}_{k|k+1}^T \boldsymbol{\Phi}_k^T \boldsymbol{C}_{k+1} \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k+1} \middle| \mathscr{D}_k\right]$,

for the next few time-steps the search is restricted to that area, this way the most extreme of jumps are avoided. In relation to this, in order to better localise the search in the first place, the expected value is minimized on its own. It will later be seen that it is small in comparison with the quadratic term, so doing a separate search on it means that smaller search intervals can be made for when the total cost should be minimized. The effects of the expectation term is then less likely to be lost in the parabolic interpolation of the solver. All this does not eliminate the high frequency jitter, so as a final step to give a smoother control signal, a simple second order low-pass filter is applied, with the exception of when the measurement interval just has changed since a large change in control behaviour is expected then.

The algorithm presented in Algorithm 6.1 has a computational cost that is linear in the number of particles, but with some further assumptions, the computational cost can be improved to be logarithmic instead. This logarithmic algorithm can be found in Algorithm 6.2 and it requires that the input particles $\{\boldsymbol{x}_k^i, \boldsymbol{x}_{k+1}^{0,i}\}$ are sorted according to y_{k+1}^i. Sorting is well known to have cost $O(N\log N)$ making the total cost of the new algorithm more expensive if the particles weren't already sorted which unfortunately can't be guaranteed. However, it's still beneficial to make the effort to sort them since in the course of the minimisation, the cost has to be evaluated several times while the particles only have to be sorted once. As long as the number of cost evaluations needed to find the minimum are smaller than $O(\log N)$ a net gain can be obtained, especially when considering that the per-iteration cost of sorting a vector of scalars is much cheaper than for the cost minimisation. One important note about both Algorithm 6.1 and Algorithm 6.2 is that in a real implementation some edge cases need to be handled to avoid divide by zero. These cases were left out since they give no further understanding of the core idea behind the algorithms.

6.3 M-Measurement Cost Feedback

When M-measurement cost feedback should be used, first note that the same simplifications can be made to the MMC cost-to-go (5.40) as the ones made to the one-measurement feedback with regards to the state covariance $\mathbf{E}_{k+M|k+M}$. This gives an equivalent cost of:

$$J_k^{MMC}(\mathscr{D}_k) = \hat{\boldsymbol{x}}_{k+M|k}^T \boldsymbol{S}_{k+M} \hat{\boldsymbol{x}}_{k+M|k} + \sum_{t=k}^{k+M-1} \hat{\boldsymbol{x}}_{t|k}^T \boldsymbol{Q}_t^x \hat{\boldsymbol{x}}_{t|k} + 2\hat{\boldsymbol{x}}_{t|k}^T \boldsymbol{Q}_t^{xu} \boldsymbol{u}_t + \boldsymbol{u}_t^T \boldsymbol{Q}_t^u \boldsymbol{u}_t - \mathrm{E}_{\boldsymbol{Z}_{k+1:k+M}} \left[\hat{\boldsymbol{x}}_{k|k+N}^T \boldsymbol{\Phi}_{k:k+M}^T (\boldsymbol{F}_{k+M} - \boldsymbol{S}_{k+M}) \boldsymbol{\Phi}_{k:k+M} \hat{\boldsymbol{x}}_{k|k+N} \middle| \mathscr{D}_k \right] \tag{6.14}$$

Before minimising the cost one further restriction will be made to the control policy used in the cost-to-go evaluation for the measurement horizon. M-measurement cost was derived by restricting this policy to not depend on any of the states

Algorithm 6.2: One-measurement Feedback - Logarithmic Cost Evaluation

Data: System matrices,$\{\boldsymbol{\Phi}_k, \boldsymbol{\Gamma}_k, \boldsymbol{C}, \boldsymbol{C}_{k+1}, \Delta q\}$, the control for which to evaluate the cost, $\boldsymbol{u}_k$, N sets of weights, particles and linear measurements, $\{\omega_k^i, \boldsymbol{x}_k^i, \boldsymbol{x}_{k+1}^{0,i}, y_{k+1}^i\}$, sorted such that $y_{k+1}^i \leq y_{k+1}^{i+1}$, and the vector of the accumulative sum of $\boldsymbol{x}_{k+1}^{0,i}$ and ω_k^i, $s_x^i = \sum_{j=1}^i \omega_k^j \boldsymbol{x}_{k+1}^{0,j}$ and $s_\omega^i = \sum_{j=1}^i \omega_k^j$ respectively

Result: $J_{uc} \approx \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\hat{\boldsymbol{x}}_{k|k+1}^T \boldsymbol{\Phi}_k^T \boldsymbol{C}_{k+1} \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k+1} \middle| \mathscr{D}_k\right]$

begin

```
/* Displace the interval boundaries according to
   control                                          */
```

$z^u \leftarrow \frac{\Delta q}{2} - \boldsymbol{C}\boldsymbol{\Gamma}_k \boldsymbol{u}_k, \quad z^l \leftarrow -\frac{\Delta q}{2} - \boldsymbol{C}\boldsymbol{\Gamma}_k \boldsymbol{u}_k$

```
/* Find particles in intervals                      */
```

$i^u \leftarrow$ find largest index i that satisfies $y_{k+1}^i < z^u$

$i^l \leftarrow$ find smallest index i that satisfies $y_{k+1}^i > z^l$

```
/* Calculate state means and measurement
   probabilities                                    */
```

$p_z^u = 1 - s_\omega^{i^u}, \quad p_z^m = s_\omega^{i^u} - s_\omega^{i^l}, \quad p_z^l = s_\omega^{i^l}$

$\widehat{\boldsymbol{x}}^u \leftarrow \frac{s_x^N - s_x^{i^u}}{p_z^u}, \quad \widehat{\boldsymbol{x}}^m \leftarrow \frac{s_x^{i^u} - s_x^{i^l}}{p_z^m}, \quad \widehat{\boldsymbol{x}}^l \leftarrow \frac{s_x^{i^l}}{p_z^l}$

```
/* Add together total cost                          */
```

$J_{uc} \leftarrow (\widehat{\boldsymbol{x}}^u)^T \boldsymbol{C}_{k+1} \widehat{\boldsymbol{x}}^u p_z^u + (\widehat{\boldsymbol{x}}^l)^T \boldsymbol{C}_{k+1} \widehat{\boldsymbol{x}}^l p_z^l + (\widehat{\boldsymbol{x}}^m)^T \boldsymbol{C}_{k+1} \widehat{\boldsymbol{x}}^m p_z^m$

end

$\boldsymbol{x}_{k+1}, \boldsymbol{x}_{k+2}, \ldots \boldsymbol{x}_{k+M}$ but it is now restricted to be a functional f on the form

$$\boldsymbol{u}_t = f(\boldsymbol{x}_k, t, p_u) \tag{6.15}$$

where p_u is a one dimensional parameter. The minimisation of the cost then becomes a one dimensional minimisation problem over p_u instead of the M dimensional problem that was had before when all of $\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \ldots \boldsymbol{x}_{k+M-1}$ needed to be determined. This allows for the exact same low-pass filtered direct approach that was used for the one-measurement feedback case but it's also a necessary restriction. Multi-dimensional direct search methods are to slow, and derivative information for more than one step ahead problems is exceedingly hard to obtain, making other solution methods unobtainable.

The quality of the control will depend on the choice of functional f and it should be chosen with care. Two different choices of f will therefore be examined, both inspired by deterministic LQR control. The first will simply be called LQR M-

measurement cost feedback and the second will be called a Reference-LQR M-measurement feedback. For the LQR-MMC feedback the parameter will be chosen such that $p_u = \boldsymbol{u}_k$ and the following control actions will simply follow from CE/LQR-based control of the open loop predictions of the state mean.

$$\boldsymbol{u}_t = f(\boldsymbol{x}_k, t, p_u) = \begin{cases} p_u, & \text{if } t \geq k \\ \boldsymbol{L}_t \hat{\boldsymbol{x}}_{t|k}, & \text{otherwise} \end{cases} \tag{6.16}$$

where $\boldsymbol{L}_t$ is given by (6.1) and $\hat{\boldsymbol{x}}_{t|k}$ from (5.20) with the recursion started at $\hat{\boldsymbol{x}}_{k|k}$. Reference-LQR is very similar but the parameter is chosen to $p_u = r$ and the control follows from a CE-control of the first state against that reference value r, resulting in

$$\boldsymbol{u}_t = f(\boldsymbol{x}_k, t, p_u) = \boldsymbol{L}_t \left(\hat{\boldsymbol{x}}_{t|k} - \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} \right) \tag{6.17}$$

The direct search over p_u will result in $J_k^{MMC}(\mathscr{D}_k)$ being evaluated over and over again for different known sequences of $\boldsymbol{u}_t$. The quadratic terms are easily evaluated with the known control sequence and deterministic open-loop dynamics of $\hat{\boldsymbol{x}}_{t|k}$, and the expected value can be tackled much in the same way as it was in the one-measurement feedback case, by running the filter forward in time and forming the filtered estimate as

$$\hat{\boldsymbol{x}}_{k|k+M} = \sum_{i=1}^{N} \omega_{k+M}^i \boldsymbol{x}_k^i \tag{6.18}$$

The probability for a certain measurement sequence $\boldsymbol{Z}_{k+1:k+M}$ is formed by adding together all the un-normalised particle weights.

$$p(\boldsymbol{Z}_{k+1:k+M}|\mathscr{D}_k) \approx \widehat{p}(\boldsymbol{z}_{k+1}|\boldsymbol{D}_k) = \sum_{i=1}^{N} \omega_k^i \prod_{j=k+1}^{k+M} p(\boldsymbol{z}_j|\boldsymbol{x}_j^i) \tag{6.19}$$

However, unlike the one-measurement feedback, the calculation can not be sped up by sorting and forming the accumulative sums, the problem being that the particles can not be guaranteed to be sorted according to all linear measurements y_k^i at the same time. The naive/simple way of cost evaluation is then to simply loop through all possible sequences of $\boldsymbol{z}_k, \boldsymbol{z}_{k+1}, \ldots \boldsymbol{z}_{k+M}$ and because of time restriction, this is what was done in this thesis. Even when only considering the three closest quantisation intervals, this scales very badly. For each of the 3^M possible sequences all particles have to be iterated over for each time step, making the computational cost $O(NM3^M)$. As previously mentioned, the particle filter resolution will be limited forward since no resampling is used but it is the exponential computational cost that is the real forward restriction.

The worst case exponential cost can not be completely removed but, like before, the computational cost can be drastically reduced by utilising the fact that the particles can only be in one quantisation interval at the time. The algorithm presented

in Algorithm 6.3 has a worst case cost of $O(3^M)$ and an average cost significantly cheaper than that. As mentioned before, the algorithm will however not be further examined because of time restrictions. Another approach for reducing computational cost worth mentioning and which was experimented with was evaluating the expected value over the sequences $\boldsymbol{Z}_{k+1:k+M}$ with Monte Carlo sampling. The problem with that approach was that the number of sampled sequences, $\boldsymbol{Z}_{k+1:k+M}$, needed to achieve good enough precision was not small enough to result in cheaper computation.

Algorithm 6.3: MMC Feedback - Cost Evaluation

Data: System matrices,$\{\boldsymbol{\Phi}_t, \boldsymbol{\Gamma}_t, \boldsymbol{C}, \boldsymbol{C}_{k+M}, \Delta q\}$, the control sequence for which to evaluate the cost, $\{\boldsymbol{u}_k\}$, N sets of weights, zero-control-predicted particles, and linear measurements, $\{\omega_k^i, \boldsymbol{x}_k^i, \boldsymbol{x}_{k+1}^{0,i}, ...\boldsymbol{x}_{k+M}^{0,i}, y_{k+i}^i, ...y_{k+M}^i\}$

Result: $J_{uc} \approx \mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\hat{\boldsymbol{x}}_{k|k+M}^T \boldsymbol{\Phi}_{k:k+M}^T \boldsymbol{C}_{k+M} \boldsymbol{\Phi}_{k:k+M} \hat{\boldsymbol{x}}_{k|k+M} \middle| \mathscr{D}_k\right]$

begin

- $L_Z \leftarrow$ empty list of measurement sequence
- $L_x \leftarrow$ empty list of state vector $\boldsymbol{x}$
- $L_p \leftarrow$ empty list of probabilities P
- **foreach** ω_k^i **do**
  - $\boldsymbol{U} \leftarrow \boldsymbol{0}$
  - $\boldsymbol{Z} \leftarrow$ empty measurement sequence
  - **foreach** $t \in \mathbb{Z}; k < t \leq k+M$ **do**
    - $\boldsymbol{U} \leftarrow \boldsymbol{\Phi}_{t-1}\boldsymbol{U} + \boldsymbol{\Gamma}_{t-1}\boldsymbol{u}_{t-1}$
    - $z^u \leftarrow \frac{\Delta q}{2} - \boldsymbol{C}\boldsymbol{U}, \quad z^l \leftarrow -\frac{\Delta q}{2} - \boldsymbol{C}\boldsymbol{U}$
    - **if** $y_{k+t}^i > z^u$ **then**
      - Add identifier for upper interval to end of $\boldsymbol{Z}$
    - **else if** $y_{k+t}^i < z^l$ **then**
      - Add identifier for lower interval to end of $\boldsymbol{Z}$
    - **else**
      - Add identifier for mid interval to end of $\boldsymbol{Z}$
    - **end**
  - **end**
  - **if** $\boldsymbol{Z}$ *is in* L_Z **then**
    - $idx \leftarrow$ index of $\boldsymbol{Z}$ in L_Z
    - $L_x[idx] \leftarrow L_x[idx] + \omega_k^i \boldsymbol{x}_{k+M}^{0,i}, \quad L_p[idx] \leftarrow L_p[idx] + \omega_k^i$
  - **else**
    - Add $\boldsymbol{Z}$, $\omega_k^i \boldsymbol{x}_{k+M}^{0,i}$ and ω_k^i to end of L_Z, L_x and L_p respectively
  - **end**
- **end**
- $J_{uc} \leftarrow 0$
- **foreach** *Element* $\boldsymbol{Z}$ *of* L_Z **do**
  - $idx \leftarrow$ index of $\boldsymbol{Z}$ in L_Z
  - $p_z \leftarrow L_p[idx], \quad \hat{\boldsymbol{x}} \leftarrow \frac{L_x[idx]}{p_z}$
  - $J_{uc} \leftarrow J_{uc} + \hat{\boldsymbol{x}}^T \boldsymbol{C}_{k+M} \hat{\boldsymbol{x}}\, p_z$
- **end**

end

7

Closed-Loop Analysis

This chapter covers the evaluation of the feedback policies presented before via simulation of the studied system. Like the open-loop analysis, the random seeds will be kept the same for all simulations, keeping it fair and equal for all controllers. The same seeds that resulted in the open-loop trajectories seen in Figure 4.1 will be used since it has a well balanced variety in the load disturbance.

After an introduction of the results of the CE/OLOF, which will be used for reference, a quick look will be had at the performance and shortcomings of the myopic controller. Following that, a variety of different open-loop horizons will be tested for the one-measurement feedback and the effect on the control action and resulting cost will be examined. After a suitable open-loop horizon has been determined the effect of the measurement horizon on MMC-feedback will be studied. The chapter will finish with a short summary and comparison of the different methods' computational cost. If nothing else is stated, the default process parameters found in Table 1.1 are used together with the best performing GA-PF with 10000 particles from the open-loop analysis chapter.

7.1 Performance Metrics

The main performance metric used will be the actual cost according to (1.7). Since the cost is in the form of an expected value, a proper empirical evaluation would require testing the policies on several different open-loop trajectories, but in order to simplify the testing and reduce the time needed this is not done, the long simulation time of 1000 seconds is deemed long enough to compensate. If nothing else is stated, in order to allow for the stochastic nature of the particle filter are the simulations repeated 10 times and sample mean and standard deviation are calculated. For proper estimation of the mean and standard deviation a lot more repetitions are needed but this will be enough to make simple statements on the relative performance of the methods. Besides the objective metric of cost, a more subjective

Table 7.1 Mean and sample standard deviation for the cost for certainty equivalent control

	Mean Cost	STD Cost	Normalized Mean Cost	Normalized STD Cost
CE/OLOF	43.46	0.895	1	0.020

approach will also be taken. The resulting closed-loop trajectories, as well as the state estimate and control signal, will be visually compared in order to evaluate the existence and effectiveness of probing effect in the different policies.

The certainty equivalent/open-loop optimal feedback will be used as a reference and the resulting cost of its use can be found in Table 7.1. For ease of comparison, when the cost of the other policies are presented they will all be normalized by the cost of the CE-control, making everything below 1 an improvement in cost.

A typical example of the control behaviour of the CE control can be found in Figure 7.1. As expected, it shows no sign of probing and no control action is taken before a new measurement, and therefore an updated estimate, is received. Also note that due to the control having no direct effect on the third state which models the load disturbance, the estimate of that is particularity bad. The only time new information about the state of the load disturbance is received is at, and right after, a new measurement is gotten.

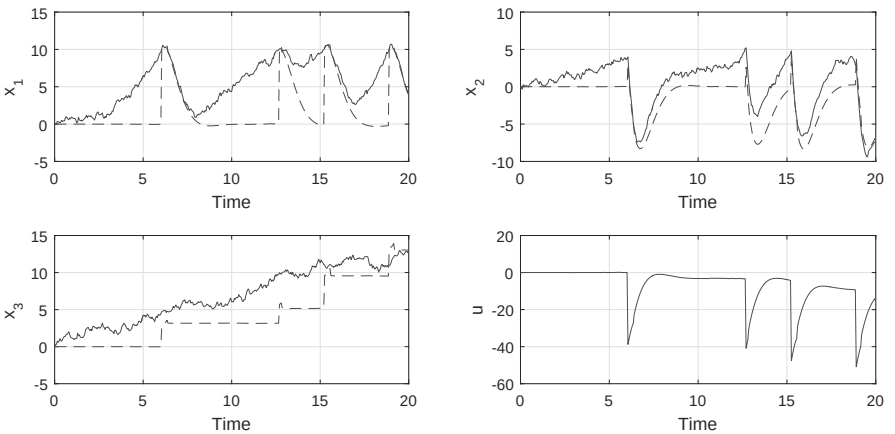

Figure 7.1 The resulting trajectories (solid) and estimates (dashed) from certainty equivalent control

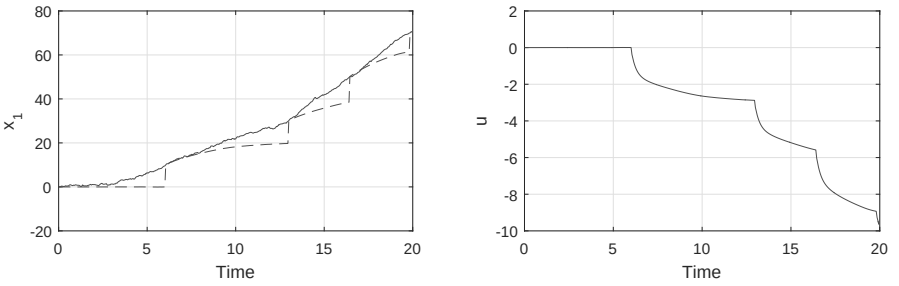

Figure 7.2 The resulting trajectory and estimate from myopic control

7.2 Results

Myopic Control

To test the myopic control the one-measurement feedback algorithm was used with an open-loop horizon of two, $N_{OL} = 2$. The resulting control and trajectory of the first state can be found in Figure 7.2 and it's clear that the controller does not manage to control the system. In this case, myopic control is not computationally cheaper than OMF but for a general system, myopic control is computationally attractive. However, it's clear that by not looking long enough into the future, the potential gain of an aggressive control action at the moment will not be seen. A myopic controller will therefore be cautious and will prioritise the cost associated with the control, causing too weak control input. Myopic control is therefore only really suitable for processes where the control has a strong direct effect on the output.

One-Measurement Feedback

Due to failure of the myopic controller, it's clear that a longer open-loop horizon is needed. The resulting cost of different horizons between $N_{OL} = 10$ and $N_{OL} = 250$ can be seen in Table 7.2. It's clear that the best results are obtained for open-loop horizons between 50-150 which is in line with the discussion in the previous chapter. Note however that the OMF doesn't perform any better than CE-control and the results for N_{OL} between 50-150 can't really be separated since they're all

Table 7.2 Control cost for One-measurement feedback, depending on the open-loop horizon, N_{OL}

	One-Measurement Feedback					
N_{OL}	10	50	100	150	200	250
Mean Cost	2.108	1.028	1.024	1.024	1.174	1.572
STD Cost	0.135	0.055	0.013	0.034	0.010	0.012

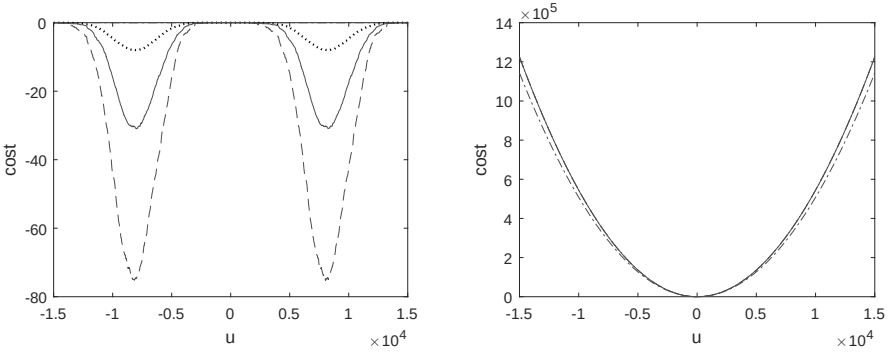

Figure 7.3 The one-measurement feedback cost-to-go as a function of control signal for $N_{OL} = 50$ (dotted), $N_{OL} = 100$ (solid), and $N_{OL} = 150$ (dashed). Cost calculated at 2 seconds after the start. **Left:** The non-quadratic cost related to state uncertainty. **Right:** The quadratic cost directly related to control and mean state.

within the error margins of each other. A better understanding for why this is can be achieved if one plots the cost function $J_k^{OMF}(\mathscr{D}_k)$ for different N_{OL} at some example state distribution. This is done in Figure 7.3 and the cost is further split up into the quadratic and non-quadratic part of $J_k^{OMF}(\mathscr{D}_k)$ given in (6.10). The quadratic part is given by

$$2\hat{\boldsymbol{x}}_{k|k}^T(\boldsymbol{Q}_k^{xu} + \boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1}\boldsymbol{\Gamma}_k)\boldsymbol{u}_k + \boldsymbol{u}_k^T(\boldsymbol{Q}_k^u + \boldsymbol{\Gamma}_k^T \boldsymbol{S}_{k+1}\boldsymbol{\Gamma}_k)\boldsymbol{u}_k \tag{7.1}$$

which is equal to the cost minimised in the CE-control. The non-quadratic part is related to the state-uncertainty and is given by the last term of (6.10).

$$-\mathrm{E}_{\boldsymbol{z}_{k+1}}\left[\hat{\boldsymbol{x}}_{k|k+1}^T \boldsymbol{\Phi}_k^T \boldsymbol{C}_{k+1}\boldsymbol{\Phi}_k\hat{\boldsymbol{x}}_{k|k+1}\middle|\mathscr{D}_k\right] \tag{7.2}$$

From the plot it's clearly seen that the quadratic part doesn't change much for $N_{OL} > 50$ which once again is in agreement with the previous discussion that showed $\boldsymbol{S}_k$ has in large converged by then. The behaviour of the non-quadratic part also follows the previous discussion with it's valleys not showing any signs of converging. What are of importance here are the scale of the axes, the non-quadratic part is 4 to 5 order of magnitudes smaller than the quadratic part. The range of $\boldsymbol{u}_k$ over which the graphs are drawn are also a couple orders of magnitude larger than the control signals resulted from the CE-control. Therefore is the impact of the non-quadratic part pretty much negligible, the minima of the non-quadratic part are too small and too far away from the minima of the quadratic term.

The valleys of the non-quadratic term are of course dependent on the distribution of $\boldsymbol{x}_k$, when the distribution becomes wider, so does the valleys. Eventually they

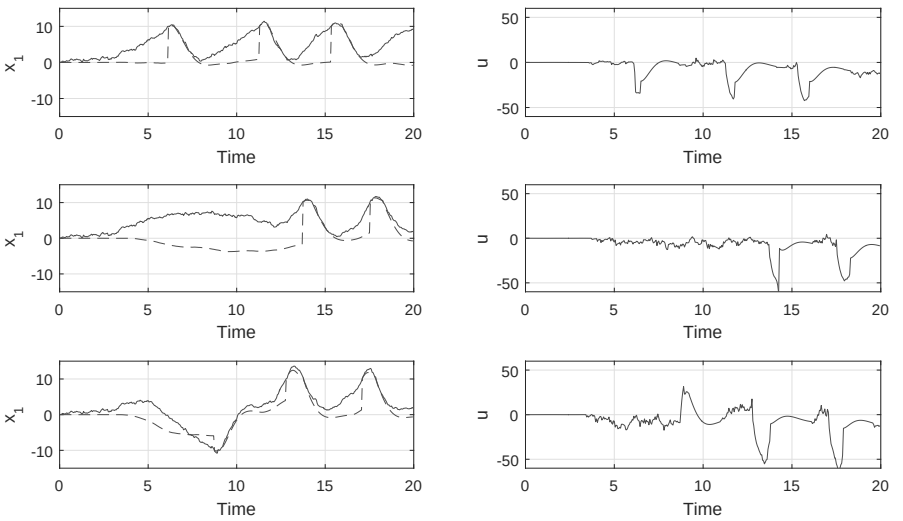

Figure 7.4 Actual (solid) and estimated (dashed) position and resulting control signal when using one-measurement feedback for different open-loop horizons. **Top:** $N_{OL} = 150$, **Middle:** $N_{OL} = 200$, **Bottom:** $N_{OL} = 250$

will meet in the middle, completely removing the flat part in between them. At this point the effects of the non-quadratic part will be felt and result in a slight shift in the control-signal compared to CE-control. However, the effect is small and will be sensitive to changes in the state-distribution. The limited resolution of the particle filter will then introduce jitter in the control signal which can be seen in Figure 7.4 and was discussed in the previous chapter. The impact of the non-quadratic part can of course be increased with longer open-loop horizons but that will affect the accuracy of approximation. This can be seen in the worse performance of the longer horizons. The impact will also be larger for processes where the control has larger direct impact on the output, this is the reason for why the minimum of the non-quadratic minimum are located at so large $\boldsymbol{u}_k$. It requires huge control inputs in order to have an effective impact on the state-distribution, the effectiveness of OMF control is therefore largely dependent on the process itself.

The effect on the control signal of when the non-quadratic part is given more and more weight by increasing the open-loop horizon can seen in Figure 7.4. It's clear that for $N_{OL} = 150$ very little probing effect is present and the control is largely reactive in the same way as CE-control, no real change of control action is made until a new measurement is received. The same behaviour was also present for $N_{OL} < 150$

with the only real difference being that the jitter, as expected, became smaller for shorter horizons. Given this and the fact that the quadratic terms of the cost-to-go completely dominate, it is no surprise that these controllers perform very similar to the CE control. It is believed that if it wasn't for the jitter introduced by the PF they would be equivalent. Probing effect is only really seen when $N_{OL} > 150$, with the probing being stronger for larger N_{OL}.

The probing actions seen for when $N_{OL} = 200$ and $N_{OL} = 250$ clearly do not translate into lower final cost, meaning that the open-loop approximation the probing is based upon, clearly isn't good enough. However, worth noting is that over the 20 second interval displayed in Figure 7.4, the careful probing of $N_{OL} = 200$ results in a significantly lower cost, with an improvement of around 25-30% by avoiding the large control effort of steering back the system after a measurement change. These gains are sadly random in nature and do not manage to result in an improvement over the entire 1000 second benchmark and the reason for it managing to reduce the cost is a result of a couple of lucky outcomes.

The first lucky outcome is the probe direction. In this case the state distribution is symmetric up until the first new measurement or control action, meaning that the first probe direction is a function of the random distribution of particles in the PF. If it would have probed in the other direction it would have simply steered faster towards the upper measurement boundary resulting in a control very similar to the one in $N_{OL} = 150$. The other reason for it's success is that the probing rate manages to match the load disturbance well during the interval, once again a random event. An example of what happens when the probing doesn't match the load disturbance can be seen for $N_{OL} = 250$. In that case the controller will probe in the right direction but too aggressively, the system is then simply driven quickly to the other measurement boundary and is never left without control input. With it always be-

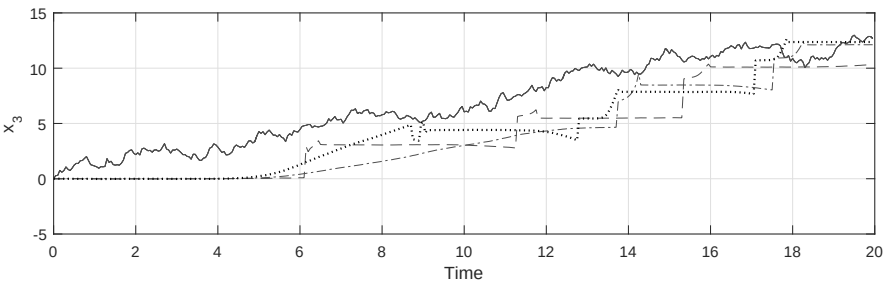

Figure 7.5 Actual (solid) and estimated load disturbance when using one-measurement feedback for different open-loop horizons, $N_{OL} = 150$ (dashed), $N_{OL} = 200$ (dashed-dotted), and $N_{OL} = 250$ (dotted)

Table 7.3 Control cost for M-measurement cost feedback, depending on the measurement and open-loop horizon, M and N_{OL} respectively.

			M			
			1	2	3	4
$N_{OL} = 100$	LQR-MMC	Mean Cost	1.024	1.044	1.122	N/A
		STD Cost	0.013	0.046	0.011	N/A
	Reference-LQR-MMC	Mean Cost	1.024	1.027	1,025	N/A
		STD Cost	0.013	0.034	0.021	N/A
$N_{OL} = 150$	LQR-MMC	Mean Cost	1.024	1.399	1.826	2.237
		STD Cost	0.034	0.011	0.013	0.018
	Reference-LQR-MMC	Mean Cost	1.024	1.152	1.349	1.540
		STD Cost	0.034	0.036	0.013	0.010

ing driven towards either zero or some boundary the controller does not allow for any kind of natural probing, i.e. a new measurement being received without actively probing for it. This over-probing is what would happen for $N_{OL} = 200$ if the load disturbance changed character, and this is exactly what happens later during the full 1000 second benchmark.

One other benefit of the probing which can be seen in Figure 7.5 is that it gives information about the load disturbance without a change in the measurement even though the effect is small. It's clear that the rate of which the load disturbance estimate is improved depends on how aggressive the probing is. However, from the figure alone it's hard to determine if this measurement-less information gain results in a better estimate seen over time. Also, since identifying the load disturbance is more of an indirect goal, this ability is more of a curiosity.

M-Measurement Cost (MMC) Feedback

Based on the results from the one-measurement feedback, suitable choices of open-loop horizon were deemed to be $N_{OL} = 100$ and $N_{OL} = 150$ so they where used when examining the MMC feedback. The results for both LQR-MMC and Reference-LQR-MMC for a couple of different measurement horizons, M, can be found in Table 7.3. From the table it's immediately clear that increasing M does not automatically give a better approximation. This should of course be expected since during the first M steps the policy is fixed and can not adapt to the measurement. In fact, with $M = \infty$ both LQR-MMC and Reference-LQR-MMC revert back to the CE-control since the measurement at which the policy is allowed to adapt never arrives. The real advantage is that the MMC-approach allows for processes with slower dynamics since the closing of the loop is done M measurements ahead. Control actions that take a couple of iterations to really take effect can not be accounted

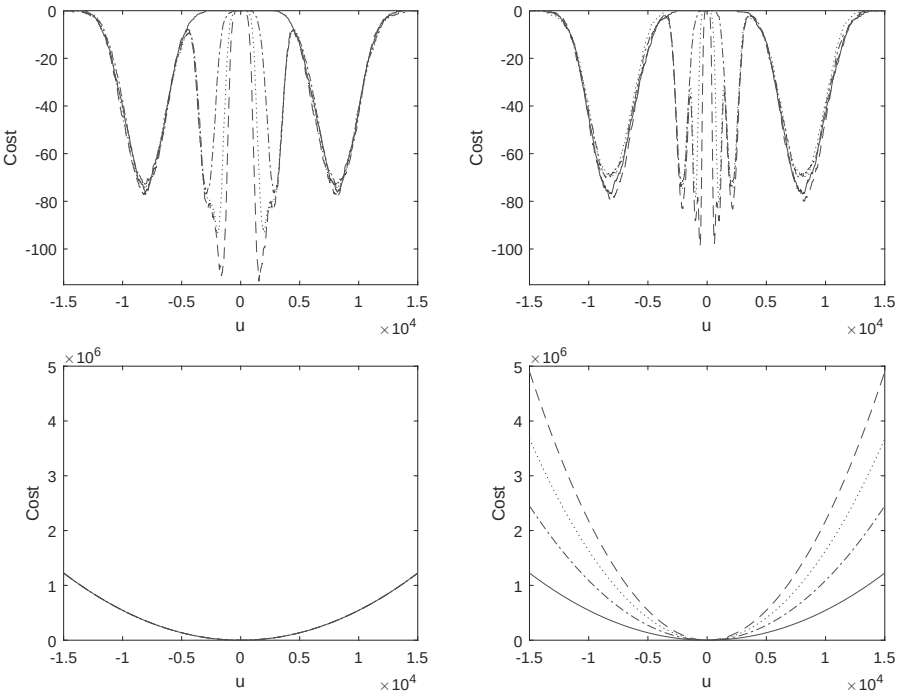

Figure 7.6 MMC feedback cost-to-go with $N_{OL} = 150$ as a function of control signal for $M = 1$ (solid), $M = 2$ (dashed-dotted), $M = 3$ (dotted), and $M = 4$ (dashed). Cost calculated at 2 seconds after the start. **Left:** LQR-MMC, **Right:** Reference-LQR-MMC, **Top:** Non-quadratic uncertainty term of cost, **Bottom:** Quadratic certainty terms of cost.

for. The importance of the choice of f is also seen from the results. To illustrate both of these properties, the non-quadratic and quadratic terms of the cost function is plotted in Figure 7.6.

From Figure 7.6 the effect of larger M is clear, increasing M creates local minima closer to zero. This is because it is mainly the position of the state distribution in relation to the measurement intervals that determines the non-quadratic term. The same amount of movement in the system can be achieved with a smaller control input over a longer period of time, as a large control input over a short period of time. This is the reason for these local minima since there now are multiple ways of reaching the same position over different lengths of time.

A difference between LQR-MMC and Reference-LQR-MMC can also be seen with

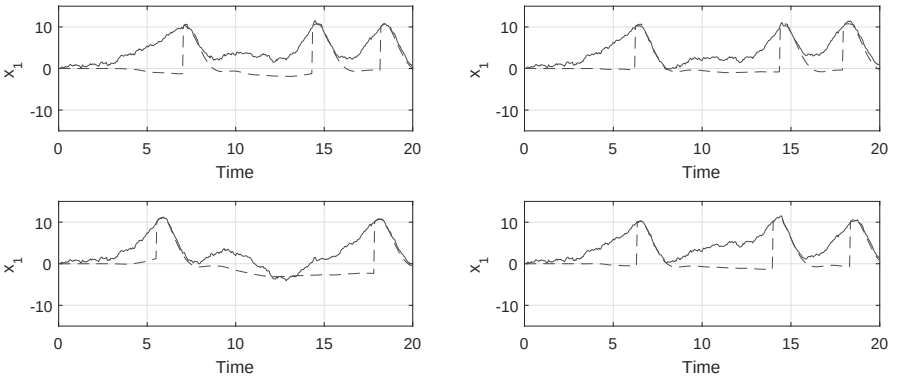

Figure 7.7 Actual (solid) and estimated (dashed) position when using MMC feedback with $N_{OL} = 100$ for different measurement horizons. **Left:** LQR-MMC, **Right:** Reference-LQR-MMC, **Top:** $M = 2$, **Bottom:** $M = 3$

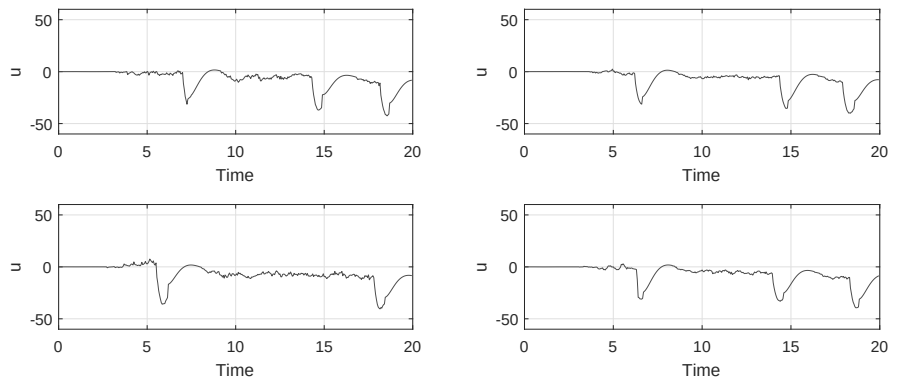

Figure 7.8 Control signal when using MMC feedback with $N_{OL} = 150$ for different measurement horizons. **Left:** LQR-MMC, **Right:** Reference-LQR-MMC, **Top:** $M = 2$, **Bottom:** $M = 3$

Reference-LQR-MMC having the minima of the non-quadratic part even closer to zero. This can be reasoned about in the same way in terms of different ways of achieving the same amount of movement in the system. For Reference-LQR policy, a control action $\boldsymbol{u}_k$ is in this case followed by $\boldsymbol{u}_{k+1}, \boldsymbol{u}_{k+2}...$ that are larger than $\boldsymbol{u}_k$ and the opposite is true for the LQR-policy. This means that Reference-LQR-MMC expects a greater amount of movement for the same $\boldsymbol{u}_k$ compared to LQR-MMC, meaning that the same minima are reached for smaller $\boldsymbol{u}_k$, moving the minima close to zero.

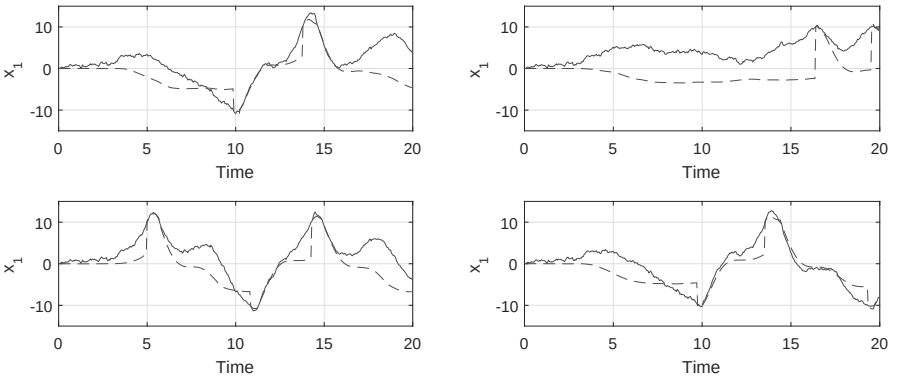

Figure 7.9 Actual (solid) and estimated (dashed) position when using MMC feedback with $N_{OL} = 150$ for different measurement horizons. **Left:** LQR-MMC, **Right:** Reference-LQR-MMC, **Top:** $M = 2$, **Bottom:** $M = 3$

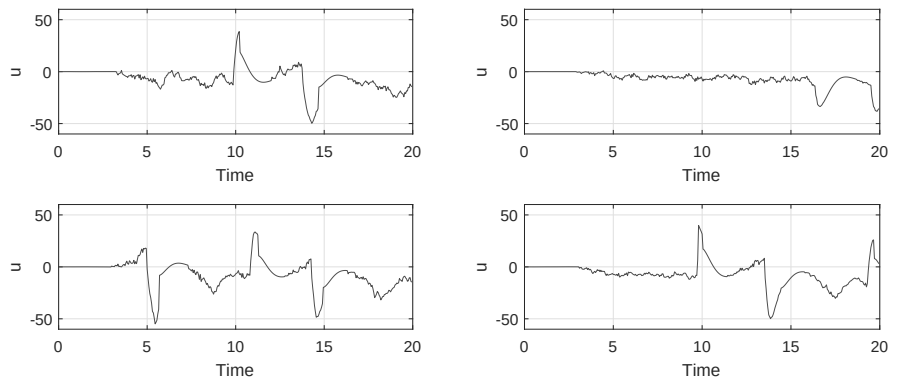

Figure 7.10 Control signal when using MMC feedback with $N_{OL} = 150$ for different measurement horizons. **Left:** LQR-MMC, **Right:** Reference-LQR-MMC, **Top:** $M = 2$, **Bottom:** $M = 3$

These minima close to zero have the effect of increasing the impact of the non-quadratic uncertainty term, making probing actions to occur at shorter open-loop horizons. The fact that they are closer to zero and the valleys are much narrower and well defined also makes them slightly less sensitive to the particle filter jitter. This is helped further by the fact that for Reference-LQR-MMC the quadratic term becomes more narrow with larger M. This doesn't happen for the LQR-policy since it's the policy that minimises CE cost-to-go which is the same as the quadratic terms of the MMC cost-to-go. All this results in the Reference-LQR-MMC being

more cautious than the LQR-MMC and it's demonstrated in the example trajectories presented in Figure 7.7 to 7.10.

Other than what's previously been discussed the behaviour of the MMC-feedback is very similar to the one-measurement feedback. The best performance is achieved when the probing effect is minimal as seen in the resulting position Figure 7.7 and control Figure 7.8 for $N_{OL} = 100$. The Reference-LQR-MMC has no real probing regardless of M and performs the best while the performance of LQR-MMC deteriorates as M increases and introduces very light probing. For $N_{OL} = 150$ all controllers suffer from over-probing as can be seen in Figure 7.9 and Figure 7.10 but once again with Reference-LQR-MMC being more cautious and thereby performing better. However, note the similarities in both trajectories and cost between Reference-LQR-MMC with $M = 2$ and $N_{OL} = 150$ and OMF with $N_{OL} = 200$.

Computational Cost and Feasibility

Up until now little mention has been made of the feasibility of using these methods in a real-time application. In Table 7.4 are the average simulation times for the 1000 second benchmark for the different controllers. The simulations were made on a 4-core i5 based desktop with 8 GiB of ram, and for comparison, a stand alone simulation without any filter or control took around 15 seconds. The results are here quite telling in that these methods are really expensive. Running a particle filter alone is not something that usually is possible on a simple embedded system and even the OMF-controllers is here around 2.5 times more expensive, forcing the intended system to have hardware equivalent to modern pc/desktop computational power. None of the MMC-controllers are really feasible, especially considering that they didn't provide any real advantage in performance.

Note however that the algorithm for the MMC controller here is the naive approach that iterates over all possible future measurement sequences, causing an unnecessary expensive evaluation. The algorithm presented in Algorithm 6.3 is expected to perform much better. Although it has a computational cost of $O(3^M)$, a more accurate cost would be $O(NM + 3^M)$ where N is the number of particles in the PF. For small M the NM term is expected to dominate, meaning the cost should scale

Table 7.4 Simulation time for different feedback policies

	CE/OLOF	OMF	MMC: $M = 1$	MMC: $M = 2$	MMC: $M = 3$	MMC: $M = 4$
Simulation time $[s]$	82	189	615	1317	3426	10702

linearly with M. Furthermore, the OMF algorithm used here is a special case of Algorithm 6.3 with $M = 1$ so with this discussion, it would be reasonable to expect that a sub-real-time simulation would be achievable on the hardware used here for $M = 3$ or perhaps even $M = 4$.

7.3 Conclusion

The general approach used for all the tested controllers is certainly no catch-all, plug-and-play solution, and all controllers required tuning of horizon lengths to achieve results even on par with CE control. The computational cost is a real concern, and since overall no real performance gain was had over CE control, it makes it hard to motivate the use of it in any real-world scenario. However, the methods described here can not be completely ruled out since it's clear that the behaviour clearly depends on both the system and the weights of the cost function and only one set of process and weights was tested here. The probing effects might turn out more useful for perhaps even more extreme quantisation interval sizes.

The methods tested did serve as a good indication of the difficulties of inducing the right amount of probing action into the control since all results point to that no probing at all achieves to best result. This perceived property of the system is something that should be further examined, especially something can be said about the balance between the natural and the induced probing of the process in an optimal controller. The meaning of natural probing being the occurrence of normal random events that reduce the state variance, in this case a new measurement being received without actively probing for it. What's suspected to happen here is that when the state distribution gets wide enough for the controller to see a cost improvement from a measurement change, the probability of that measurement change happening spontaneous is already high enough for the control effort not being worth it. However, there are processes where the need for active probing is higher, making the methods proposed in this thesis potentially better suited for them.

Part III

Conclusion

8

Summary and Future Prospects

The reason for the success of the particle filter over the last 20 years is obvious: the simple framework it's based on provides intuitive solution methods and accurate estimates. However, the results of this work have shown it is still important not to forget its shortcomings when applying a PF to a system. Care should be taken to design a PF that takes into account as much of the system dynamics as possible in the prediction step. The cost of running a PF is also great, especially when comparing to the simplicity of the Kalman filter. Low computational cost approximations of the optimal filter, like the ones presented in Chapter 3.1, that can provide close to optimal estimates are therefore still of great interest. Especially when considering the kind of future predictions used here when solving the dynamic programming problem. A computationally cheaper filter would allow for methods looking further in the future compared to a full-scale particle filter.

The problem of dual control and the balancing of probing and control actions needs further examination. The cost-to-go approximations used here have at the very least been shown to be inadequate for producing a suitable probing effect in the controller. In order to gain further understanding for how the optimal control action would look, and from that create better approximations, it would be useful to, in an offline setting, numerically calculate an optimal control sequence. However, at the moment there's a lack of suitable solvers for this task.

Most of the development of solvers has been focused on fully observed stochastic systems and the methods present for partially observed systems are unsuitable since they rely on a Gaussian approximation together with a linearisation of the measurement function. A fact that was mentioned very briefly, that derivative information had been extracted cheaply for a one-measurement problem, is therefore very promising. This opens up for locally optimal roll-out solvers that iterate over

linear quadratic approximations around a state and measurement trajectory. Such a method could potentially solve any partially observed stochastic dynamic programming problem. The same iterative solver could most likely be used in real-time for the OMF problem if that policy turns out to be useful in some other situation. The iterative solver wouldn't suffer as much from the jitter introduced by the PF and with minor modifications could also handle the delayed one-measurement feedback that was shortly discussed in relation to the MMC-feedback. Since no real changes in behaviour were seen between the MMC-feedback and OMF, the delayed OMF might be a suitable replacement for the MMC-feedback when it is needed to handle systems with slow dynamics.

Although the goal of improving the control over a certainty equivalent approach wasn't achieved, some important initial steps have been taken in a few areas. A very low cost method for an approximative filter, utilising a dynamic model of the measurement error and a Kalman filter, was presented and showed early promises in preliminary testing. The particle filter was introduced to a dynamic programming setting and even though this has been done before, methods designed specifically around the PF have not. By utilising the structure of the particle filter, the computational cost could greatly be reduced by not re-predicting the particles multiple times. Both of these concepts can serve as important stepping stones, leading to a future with cheaper and better dual controllers.

Part IV

Appendices

A

Discretization

A.1 Discretized State-Space Model

Given a LTI system on state-space form

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{N}\boldsymbol{\nu} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} \end{aligned} \tag{A.1}$$

where $\boldsymbol{\nu}$ is continuous zero mean white noise with power spectral density $\boldsymbol{R}$, i.e. the integral of $\boldsymbol{\nu}$ between a and b is a random variable with zero mean and covariance $\boldsymbol{R}(b-a)$. The concept of continuous white noise is in reality more nuanced but this will suffice for this simple purpose.

By integrating from t_k to t_{k+1}, an interval with length h_k, and defining $\boldsymbol{x}(t_k) = \boldsymbol{x}_k$, $\boldsymbol{u}(t_k) = \boldsymbol{u}_k$ and $\boldsymbol{y}(t_k) = \boldsymbol{y}_k$ we get.

$$\begin{aligned} \boldsymbol{x}_{k+1} &= e^{\boldsymbol{A}h_k}\boldsymbol{x}_k + \int_{t_k}^{t_{k+1}} e^{\boldsymbol{A}(t_{k+1}-s)}\boldsymbol{B}\boldsymbol{u}(s)ds + \int_{t_k}^{t_{k+1}} e^{\boldsymbol{A}(t_{k+1}-s)}\boldsymbol{N}\boldsymbol{\nu}(s)ds \\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k \end{aligned} \tag{A.2}$$

Consider the third term. It's a discrete random sequence we can call $\boldsymbol{\nu}_k$. The sequence mean is

$$\mathrm{E}[\boldsymbol{\nu}_k] = \mathrm{E}\left[\int_{t_k}^{t_{k+1}} e^{\boldsymbol{A}(k+1-s)}\boldsymbol{N}\boldsymbol{\nu}(s)ds\right] = \int_{t_k}^{t_{k+1}} e^{\boldsymbol{A}(t_{k+1}-s)}\boldsymbol{N}\underbrace{\mathrm{E}[\boldsymbol{\nu}(s)]}_{0}ds = 0 \tag{A.3}$$

While the autocovariance is given by

$$\begin{aligned}\mathrm{Cov}[\boldsymbol{\nu}_k,\boldsymbol{\nu}_l] &= \mathrm{E}\left[\int_{t_k}^{t_{k+1}}\int_{t_l}^{t_{l+1}} e^{\boldsymbol{A}(t_{k+1}-s)}\boldsymbol{N}\boldsymbol{\nu}(s)\boldsymbol{\nu}^T(t)\boldsymbol{N}^T e^{\boldsymbol{A}^T(t_{l+1}-t)}dtds\right] \\ &= \int_{t_k}^{t_{k+1}}\int_{t_l}^{t_{l+1}} e^{\boldsymbol{A}(t_{k+1}-s)}\boldsymbol{N}\underbrace{\mathrm{E}\left[\boldsymbol{\nu}(s)\boldsymbol{\nu}^T(t)\right]}_{0}\boldsymbol{N}^T e^{\boldsymbol{A}^T(t_{l+1}-t)}dtds = 0\end{aligned} \tag{A.4}$$

when $k \neq l$ since the interval $[t_k, t_{k+1}]$ and $[t_l, t_{l+1}]$ are disjoint and per definition $\boldsymbol{\nu}$ is uncorrelated on disjoint intervals. By being somewhat casual about the formalism for continuous white noise we get

$$\begin{aligned}\mathrm{Cov}[\boldsymbol{\nu}_k,\boldsymbol{\nu}_k] &= \iint_{t_k}^{t_{k+1}} e^{\boldsymbol{A}(t_{k+1}-s)}\boldsymbol{N}\,\mathrm{E}\left[\boldsymbol{\nu}(s)\boldsymbol{\nu}^T(t)\right]\boldsymbol{N}^T e^{\boldsymbol{A}^T(t_{k+1}-t)}dtds \\ &= \iint_{t_k}^{t_{k+1}} e^{\boldsymbol{A}(t_{k+1}-s)}\boldsymbol{N}\boldsymbol{R}\delta(t-s)\boldsymbol{N}^T e^{\boldsymbol{A}^T(t_{k+1}-t)}dtds \\ &= \int_{t_k}^{t_{k+1}} e^{\boldsymbol{A}(t_{k+1}-s)}\boldsymbol{N}\boldsymbol{R}\boldsymbol{N}^T e^{\boldsymbol{A}^T(t_{k+1}-s)}ds = \widehat{\boldsymbol{R}}_k\end{aligned} \tag{A.5}$$

for the case when $k = l$. From this it's clear that $\boldsymbol{\nu}_k$ is a zero mean white noise sequence where $\widehat{\boldsymbol{R}}_k$ denotes the power spectrum of the process.

Now consider the second term of (A.2). By using a zero-order hold on the control signal $\boldsymbol{u}(t)$ will be constant on the interval $[t_k, t_{k+1})$ which gives.

$$\int_{t_k}^{t_{k+1}} e^{\boldsymbol{A}(t_{k+1}-s)}\boldsymbol{B}\boldsymbol{u}(s)ds = \int_{t_k}^{t_{k+1}} e^{\boldsymbol{A}(t_{k+1}-s)}\boldsymbol{B}ds\,\boldsymbol{u}(t_k) = \boldsymbol{\Gamma}_k\boldsymbol{u}_k \tag{A.6}$$

Finally by identifying $e^{\boldsymbol{A}h_k}$ as $\boldsymbol{\Phi}_k$, Figure A.2 can be written as the discrete linear system.

$$\begin{aligned}\boldsymbol{x}_{k+1} &= \boldsymbol{\Phi}_k\boldsymbol{x}_k + \boldsymbol{\Gamma}_k\boldsymbol{u}_k + \boldsymbol{\nu}_k \\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k\end{aligned} \tag{A.7}$$

where $\boldsymbol{\nu}_k$ is a zero mean white noise sequence with power spectrum $\widehat{\boldsymbol{R}}_k$.

A.2 Positive Definiteness of Discrete White Noise Power Spectrum

The power spectrum given by (A.5) is positive definite given $\boldsymbol{N}$ is non-singular and R is a full rank covariance matrix.

Proof Since $\boldsymbol{R}$ is a covariance matrix it's positive definite and therefore has non-negative eigenvalues. The added restriction of it having full rank then gives that all eigenvalues are positive and diagonalisation then gives positive definiteness.

$$\boldsymbol{z}^T \boldsymbol{R}\boldsymbol{z} > 0, \quad \forall \boldsymbol{z} \neq \boldsymbol{0} \tag{A.8}$$

By setting $\boldsymbol{z} = \boldsymbol{N}^T e^{\boldsymbol{A}^T(t_{k+1}-s)}\boldsymbol{x}$ and noting that $\boldsymbol{N}$ and $e^{\boldsymbol{A}^T(t_{k+1}-s)}$ have full rank, exponential matrices always have an inverse $(e^{\boldsymbol{M}})^{-1} = e^{-\boldsymbol{M}}$, we can conclude that $\boldsymbol{x} = \boldsymbol{0} \iff \boldsymbol{z} = \boldsymbol{0}$

$$\boldsymbol{z}^T \boldsymbol{R}\boldsymbol{z} = \boldsymbol{x}^T e^{\boldsymbol{A}(t_{k+1}-s)} \boldsymbol{N}\boldsymbol{R}\boldsymbol{N}^T e^{\boldsymbol{A}^T(t_{k+1}-s)}\boldsymbol{x} > 0, \quad \forall \boldsymbol{x} \neq \boldsymbol{0} \tag{A.9}$$

This gives

$$\begin{aligned} &\int_{t_k}^{t_{k+1}} \boldsymbol{x}^T e^{\boldsymbol{A}(t_{k+1}-s)} \boldsymbol{N}\boldsymbol{R}\boldsymbol{N}^T e^{\boldsymbol{A}^T(t_{k+1}-s)}\boldsymbol{x} ds \\ &= \boldsymbol{x}^T \int_{t_k}^{t_{k+1}} e^{\boldsymbol{A}(t_{k+1}-s)} \boldsymbol{N}\boldsymbol{R}\boldsymbol{N}^T e^{\boldsymbol{A}^T(t_{k+1}-s)} ds\, \boldsymbol{x} \\ &= \boldsymbol{x}^T \widehat{\boldsymbol{R}}_k \boldsymbol{x} > 0, \quad \forall \boldsymbol{x} \neq \boldsymbol{0} \end{aligned} \tag{A.10}$$

□

A.3 Discretisation of Continuous Cost Function

Cost functions defined in continuous time also need to be discretised when the problem is transferred to discrete time. Given a cost function on the form

$$\begin{aligned} V &= \mathrm{E}\left[\|\boldsymbol{x} - \boldsymbol{x}_r^*\|^2_{\boldsymbol{Q}^x} + \|\boldsymbol{u} - \boldsymbol{u}_r^*\|^2_{\boldsymbol{Q}^u} \right] = \mathrm{E}\left[\|\widetilde{\boldsymbol{x}}\|^2_{\boldsymbol{Q}^x} + \|\widetilde{\boldsymbol{u}}\|^2_{\boldsymbol{Q}^u} \right] \\ &= \mathrm{E}\left[\int_0^{t_f} \widetilde{\boldsymbol{x}}^T \boldsymbol{Q}^x \widetilde{\boldsymbol{x}} dt + \int_0^{t_f} \widetilde{\boldsymbol{u}}^T \boldsymbol{Q}^u \widetilde{\boldsymbol{u}} dt \right] \end{aligned} \tag{A.11}$$

where $\boldsymbol{x}_r^*$ and $\boldsymbol{u}_r^*$ are a stationary solution that gives $\boldsymbol{y} = r$ when no noise is present. By considering the integration on the discretisation intervals using a zero-order hold on the input, making it constant on each interval, one gets:

$$\begin{aligned} V &= \mathrm{E}\left[\int_0^{t_f} \widetilde{\boldsymbol{x}}^T \boldsymbol{Q}^x \widetilde{\boldsymbol{x}} dt + \int_0^{t_f} \widetilde{\boldsymbol{u}}^T \boldsymbol{Q}^u \widetilde{\boldsymbol{u}} dt \right] \\ &= \mathrm{E}\left[\sum_{k=0}^{N} \left(\int_{t_k}^{t_{k+1}} \widetilde{\boldsymbol{x}}^T \boldsymbol{Q}^x \widetilde{\boldsymbol{x}} dt + \int_{t_k}^{t_{k+1}} \widetilde{\boldsymbol{u}}^T \boldsymbol{Q}^u \widetilde{\boldsymbol{u}} dt \right) \right] \end{aligned} \tag{A.12}$$

By using a zero-order hold on $\boldsymbol{u}$ the second integral can be simplified to

$$\mathrm{E}\left[\int_{t_k}^{t_{k+1}} \widetilde{\boldsymbol{u}}^T \boldsymbol{Q}^u \widetilde{\boldsymbol{u}} dt \right] = \mathrm{E}\left[\widetilde{\boldsymbol{u}}_k^T \boldsymbol{Q}^u \widetilde{\boldsymbol{u}}_k h_k \right] \tag{A.13}$$

where like before h_k is the interval length.

The first term can be simplified by, similar to Appendix A.1, integrate the system between $[t_k, t]$, $t_k < t < t_{k+1}$ to give an expression for $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}$ on the interval.

$$\begin{aligned} \boldsymbol{x}(t) &= \boldsymbol{\Phi}_k(t)\boldsymbol{x}_k + \boldsymbol{\Gamma}_k(t)\boldsymbol{u}_k + \boldsymbol{\nu}_k(t) \\ &\Longrightarrow \\ \widetilde{\boldsymbol{x}}(t) &= \boldsymbol{\Phi}_k(t)\widetilde{\boldsymbol{x}}_k + \boldsymbol{\Gamma}_k(t)\boldsymbol{u}_k + \boldsymbol{\nu}_k(t) \end{aligned} \tag{A.14}$$

where $\boldsymbol{\Phi}_k(t) = e^{\boldsymbol{A}(t-t_k)}$, $\boldsymbol{\Gamma}_k(t) = \int_{t_k}^{t} e^{\boldsymbol{A}(t-t_k)}\boldsymbol{B}ds$ and $\boldsymbol{\nu}_k(t)$ is zero mean white noise. This can be inserted into the first integral of (A.12).

$$\begin{aligned} &\mathrm{E}\left[\int_{t_k}^{t_{k+1}} \widetilde{\boldsymbol{x}}^T \boldsymbol{Q}^x \widetilde{\boldsymbol{x}} dt\right] = \\ &\mathrm{E}\left[\int_{t_k}^{t_{k+1}} \left(\widetilde{\boldsymbol{x}}_k^T \boldsymbol{\Phi}_k^T(t) + \widetilde{\boldsymbol{u}}_k^T \boldsymbol{\Gamma}_k^T(t) + \boldsymbol{\nu}_k(t)^T\right)\boldsymbol{Q}^x\left(\boldsymbol{\Phi}_k(t)\widetilde{\boldsymbol{x}}_k + \boldsymbol{\Gamma}_k(t)\widetilde{\boldsymbol{u}}_k + \boldsymbol{\nu}_k(t)\right) dt\right] \end{aligned}$$

From definition $\boldsymbol{\nu}_k(t)$ is uncorrelated with the past, i.e. $\widetilde{\boldsymbol{x}}_k$ and $\widetilde{\boldsymbol{u}}_k$. All cross-terms will then be zero, resulting in:

$$\begin{aligned} &\mathrm{E}\left[\int_{t_k}^{t_{k+1}} \widetilde{\boldsymbol{x}}^T \boldsymbol{Q}^x \widetilde{\boldsymbol{x}} dt\right] = \\ &\mathrm{E}\left[\int_{t_k}^{t_{k+1}} \left(\widetilde{\boldsymbol{x}}_k^T \boldsymbol{\Phi}_k^T(t) + \widetilde{\boldsymbol{u}}_k^T \boldsymbol{\Gamma}_k^T(t)\right)\boldsymbol{Q}^x\left(\boldsymbol{\Phi}_k(t)\widetilde{\boldsymbol{x}}_k + \boldsymbol{\Gamma}_k(t)\widetilde{\boldsymbol{u}}_k\right) dt\right. \\ &\qquad\left. + \int_{t_k}^{t_{k+1}} \boldsymbol{\nu}_k(t)^T \boldsymbol{Q}^x \boldsymbol{\nu}_k(t)\, dt\right] \end{aligned} \tag{A.15}$$

The second term can't be controlled and an equivalent control cost function can therefore be constructed by discarding it. Combining (A.13) and (A.15) to break out all constant term then gives the discretised cost function $J[\boldsymbol{x}_k]$, acting on the sequence $\{\boldsymbol{x}_k\}$.

$$\begin{aligned} J &= \mathrm{E}\left[\sum_{k=0}^{N} \widetilde{\boldsymbol{x}}_k^T \boldsymbol{Q}_k^x \widetilde{\boldsymbol{x}}_k + 2\widetilde{\boldsymbol{x}}_k^T \boldsymbol{Q}_k^{xu} \widetilde{\boldsymbol{u}}_k + \widetilde{\boldsymbol{u}}_k^T \boldsymbol{Q}_k^u \widetilde{\boldsymbol{u}}_k\right] \\ &= \mathrm{E}\left[\sum_{k=0}^{N} (\boldsymbol{x}_k - \boldsymbol{x}_k^*)^T \boldsymbol{Q}_k^x (\boldsymbol{x}_k - \boldsymbol{x}_k^*) + 2(\boldsymbol{x}_k - \boldsymbol{x}_k^*)^T \boldsymbol{Q}_k^{xu} (\boldsymbol{u}_k - \boldsymbol{u}_k^*)\right. \\ &\qquad\left. + (\boldsymbol{u}_k - \boldsymbol{u}_k^*)^T \boldsymbol{Q}_k^u (\boldsymbol{u}_k - \boldsymbol{u}_k^*)\right] \end{aligned} \tag{A.16}$$

where

$$\begin{aligned} \boldsymbol{Q}_k^x &= \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}_k^T(t)\boldsymbol{Q}^x\boldsymbol{\Phi}_k(t)\,dt, \quad \boldsymbol{Q}_k^{xu} = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}_k^T(t)\boldsymbol{Q}^x\boldsymbol{\Gamma}_k(t)\,dt \\ \boldsymbol{Q}_k^u &= \int_{t_k}^{t_{k+1}} \boldsymbol{\Gamma}_k^T(t)\boldsymbol{Q}^x\boldsymbol{\Gamma}_k(t)\,dt + \boldsymbol{Q}^u h_k \end{aligned} \tag{A.17}$$

A.4 Numerical Matrix Calculations

Expressions for numerical computation of the discrete system matrices from Appendix A.1 together with the discretised cost function weights from Appendix A.3 will here be derived. Note that even though expressions for $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Gamma}_k$ will be given no original code was used when computing them in this work, instead MATLAB's built in functions were used. For the other matrices original code was developed and we will therefore briefly touch on some computational aspects although no actual code will be provided. With that being said, the main purpose of this section was to alleviate some curiosity and I'm aware of better ways of performing these calculations.

Extensive use of the Taylor expansion of the matrix exponential will be made in order to express the matrices as infinite sums. When doing numerical calculations tail of the sum can be truncated when desired precisions has been achieved. This will be the procedure for all these calculations so therefore this truncation will not be explicitly noted and all sum will be left as infinite.

$\boldsymbol{\Phi}_k$ is gotten directly from the Taylor expansion

$$\boldsymbol{\Phi}_k = e^{\boldsymbol{A}h_k} = \sum_{i=0}^{\infty} \frac{\boldsymbol{A}^i h_k^i}{i!} \tag{A.18}$$

For $\boldsymbol{\Gamma}_k$, the Taylor expansion is integrated after a variable change in the integral to give a cleaner notation.

$$\begin{aligned} \boldsymbol{\Gamma}_k &= \int_{t_k}^{t_{k+1}} e^{\boldsymbol{A}(t_{k+1}-s)}\boldsymbol{B}ds = \int_0^{h_k} e^{\boldsymbol{A}s}ds\,\boldsymbol{B} \\ &= \int_0^{h_k} \sum_{i=0}^{\infty} \frac{\boldsymbol{A}^i s^i}{i!}ds\,\boldsymbol{B} = \sum_{i=0}^{\infty} \frac{\boldsymbol{A}^i}{i!}\int_0^{h_k} s^i ds\,\boldsymbol{B} \\ &= \sum_{i=0}^{\infty} \frac{\boldsymbol{A}^i h_k^{i+1}}{(i+1)i!}\boldsymbol{B} = \sum_{i=0}^{\infty} \frac{\boldsymbol{A}^i h_k^{i+1}}{(i+1)!}\boldsymbol{B} \end{aligned} \tag{A.19}$$

The procedure is largely the same for $\widehat{\boldsymbol{R}}_k$

$$\begin{aligned}\widehat{\boldsymbol{R}}_k &= \int_0^{h_k} e^{\boldsymbol{A}(s)}\boldsymbol{N}\boldsymbol{R}\boldsymbol{N}^T e^{\boldsymbol{A}^T(s)}ds = \int_0^{h_k}\sum_{i=0}^{\infty}\frac{\boldsymbol{A}^i s^i}{i!}\boldsymbol{N}\boldsymbol{R}\boldsymbol{N}^T\sum_{j=0}^{\infty}\frac{\boldsymbol{A}^{T\,j}s^j}{j!}ds \\ &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{\boldsymbol{A}^i\boldsymbol{N}\boldsymbol{R}\boldsymbol{N}^T\boldsymbol{A}^{T\,j}}{i!\,j!}\int_0^{h_k}s^{i+j}ds \\ &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{\boldsymbol{A}^i\boldsymbol{N}\boldsymbol{R}\boldsymbol{N}^T\boldsymbol{A}^{T\,j}}{i!\,j!(i+j+1)}h_k^{i+j+1}\end{aligned} \tag{A.20}$$

Since the structure of $\boldsymbol{Q}_k^x$ is almost identical to $\widehat{\boldsymbol{R}}_k$ we directly get

$$\boldsymbol{Q}_k^x = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{\boldsymbol{A}^{T\,i}\boldsymbol{Q}^x\boldsymbol{A}^j}{i!\,j!(i+j+1)}h_k^{i+j+1} \tag{A.21}$$

The expressions $\boldsymbol{Q}_k^{xu}$ and $\boldsymbol{Q}_x^u$ are more cumbersome because of the nested integrals but the derivations follow the same pattern. For brevity they will therefore be stated directly.

$$\begin{aligned}\boldsymbol{Q}_k^{xu} &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{\boldsymbol{A}^{T\,i}\boldsymbol{Q}^x\boldsymbol{A}^j\boldsymbol{B}}{i!(j+1)!(i+j+2)}h_k^{i+j+2} \\ \boldsymbol{Q}_x^u &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{\boldsymbol{B}^T\boldsymbol{A}^{T\,i}\boldsymbol{Q}^x\boldsymbol{A}^j\boldsymbol{B}}{(i+1)!(j+1)!(i+j+3)}h_k^{i+j+3} + \boldsymbol{Q}^u h_k\end{aligned} \tag{A.22}$$

By identifying

$$\boldsymbol{\Theta}_j^i = \frac{\boldsymbol{A}^{T\,i}\boldsymbol{Q}^x\boldsymbol{A}^j}{i!\,j!}h_k^{i+j+1} \tag{A.23}$$

we can write the discretised weights as

$$\begin{aligned}\boldsymbol{Q}_k^x &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{\boldsymbol{\Theta}_j^i}{(i+j+1)}, \quad \boldsymbol{Q}_k^{xu} = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{\boldsymbol{\Theta}_j^i h_k}{(j+1)(i+j+2)}\boldsymbol{B} \\ \boldsymbol{Q}_x^u &= \boldsymbol{B}^T\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{\boldsymbol{\Theta}_j^i h_k^2}{(i+1)(j+1)(i+j+3)}\boldsymbol{B} + \boldsymbol{Q}^u h_k\end{aligned} \tag{A.24}$$

This is useful for computational purposes since it's enough to compute $\boldsymbol{\Theta}_j^i$ once for each i, j and updating each $\boldsymbol{Q}_k$ individually, reducing the number of matrix multiplications needed. Even further computational reductions can be made by noting the symmetry $\boldsymbol{\Theta}_j^i = (\boldsymbol{\Theta}_i^j)^T$. Both of these properties were utilised in the work for this thesis and the infinite series was truncated at $i, j = l$, where l is the smallest integer fulfilling $\left\|\boldsymbol{\Theta}_l^l\right\|_2 < \varepsilon$ for some parameter ε. No statement is made about the quality of this termination criterion other than it was chosen for it being a good fit for the access pattern of i, j used in the algorithm and it was deemed to have adequate accuracy when ε was small.

A.5 Model Partitioning

Given a LTI system on state-space form

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{N}\boldsymbol{\nu} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} \end{aligned} \tag{A.25}$$

satisfying

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A^m} & \boldsymbol{B^m} \\ \boldsymbol{0} & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{B^m} \\ 0 \end{bmatrix} \tag{A.26}$$

for some column vector $\boldsymbol{B^m}$ and some upper triangular square matrix $\boldsymbol{A^m}$, the discretised system

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k \boldsymbol{u}_k + \boldsymbol{\nu}_k \\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k \end{aligned} \tag{A.27}$$

will satisfy

$$\boldsymbol{\Phi}_k = \begin{bmatrix} \boldsymbol{\Phi^m}_k & \boldsymbol{\Gamma^m}_k \\ \boldsymbol{0} & 1 \end{bmatrix}, \quad \boldsymbol{\Gamma}_k = \begin{bmatrix} \boldsymbol{\Gamma^m}_k \\ 0 \end{bmatrix} \tag{A.28}$$

where $\boldsymbol{\Phi^m}_k$ is square and $\boldsymbol{\Gamma^m}_k$ is a column vector.

Proof We first note from (A.18) and (A.19) that

$$\begin{aligned} \boldsymbol{\Phi}_k &= \boldsymbol{I} + \boldsymbol{A}\boldsymbol{\Psi}_k \\ \boldsymbol{\Gamma}_k &= \boldsymbol{\Psi}_k \boldsymbol{B} \end{aligned} \tag{A.29}$$

where

$$\boldsymbol{\Psi}_k = \sum_{i=0}^{\infty} \psi_k^i, \quad \psi_k^i = \frac{\boldsymbol{A}^i h_k^{i+1}}{(i+1)!} \tag{A.30}$$

From this we can see that the bottom row of $\boldsymbol{\Phi}_k$ is $\begin{bmatrix} \boldsymbol{0} & 1 \end{bmatrix}$.

The constraints on $\boldsymbol{A}$ now gives:

$$\begin{aligned} &\boldsymbol{A}\begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix} = \boldsymbol{B} \implies \boldsymbol{A}^i\boldsymbol{A}\begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix} = \boldsymbol{A}^i\boldsymbol{B} \implies \boldsymbol{A}\boldsymbol{A}^i\begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix} = \boldsymbol{A}^i\boldsymbol{B} \\ \implies &\boldsymbol{A}\psi_k^i\begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix} = \psi_k^i\boldsymbol{B} \implies \boldsymbol{A}\boldsymbol{\Psi}_k\begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix} = \boldsymbol{\Psi}_k\boldsymbol{B} \implies (\boldsymbol{\Phi}_k - \boldsymbol{I})\begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix} = \boldsymbol{\Gamma}_k \end{aligned} \tag{A.31}$$

□

B

Sampling and Random Variables

B.1 Bayes' Rule - Mixed Random Variables

Bayes' rule holds when mixing discrete and continuous random variables. The probability density function (PDF) can be exchanged for the discrete variables probability mass function (PMF).

Proof Given a discrete random variable X with PMF $P(x)$ and PMF conditioned on some continuous random variable A, $P(x|a)$. Generalise the PMFs by introducing the Dirac delta function.

$$p(x) = \sum_{i=0}^{\infty} P(x)\delta(x-x_i), \quad p(x|a) = \sum_{i=0}^{\infty} P(x|a)\delta(x-x_i^a) \tag{B.1}$$

where x_i and x_i^a are all the point $P(x)$ and $P(x|a)$ are non-zero.

Bayes' rule for continuous random variables gives

$$\begin{gathered} p(a|x)p(x) = p(x|a)p(a) \\ \Longrightarrow \\ \int_{\Omega} p(a|x)p(x)dx = \int_{\Omega} p(x|a)p(a)dx, \quad \forall \Omega \end{gathered} \tag{B.2}$$

Choose Ω such that it only contains one of x_i.

$$\begin{gathered} \int_{\Omega} p(a|x) \sum_{i=0}^{\infty} P(x)\delta(x-x_i)dx = \int_{\Omega} p(a) \sum_{i=0}^{\infty} P(x|a)\delta(x-x_i^a)dx \\ \Longrightarrow \\ p(a|x_i)P(x_i) = P(x_i|a)p(a) \iff p(a|x_i) = \frac{P(x_i|a)p(a)}{P(x_i)} \end{gathered} \tag{B.3}$$

Same result is given if Ω is chosen to contain x_i^a or none of x_i or x_i^a.

$$\implies p(a|x) = \frac{P(x|a)p(a)}{P(x)} \tag{B.4}$$

□

B.2 Approximate Sampling From the Truncated Gaussian Distribution

A coarse approximative method will here be derived for sampling from a truncated Gaussian distribution. It's based on the rudimentary method of inverse transform sampling [Blom et al., 2005]. Let F be a univariate cumulative distribution function (CDF) for some random variable X. Random samples, x_i, can then be generated from F by

$$x_i = F^{-1}(u_i), \quad u_i \sim \mathscr{U}(0,1) \tag{B.5}$$

where $\mathscr{U}$ is the uniform distribution.

Truncating a distribution between a and b with $a < b$ is the same as rescaling and translating the CDF as

$$F_a^b(x) = \frac{F(x) - F(a)}{F(b) - F(a)} \tag{B.6}$$

The inverse of this expression is

$$(F_a^b)^{-1}(x) = F^{-1}\Big(\big(F(b) - F(a)\big)x + F(a)\Big) \tag{B.7}$$

This expression can be used with the inverse sampling to generate samples from any truncated distribution of an invertible CFD.

The unit Gaussian distribution has the following CFD

$$F(x) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right) \tag{B.8}$$

with inverse

$$F^{-1}(x) = \sqrt{2}\operatorname{erf}^{-1}(2x-1) \tag{B.9}$$

By replacing the error function with the inverse of the approximation (3.18)

$$\widetilde{\operatorname{erf}}^{-1}(x) = \begin{cases} \sqrt{\frac{x}{4(1-x)}} & \text{if } x < -0.5 \\ x & \text{if } -0.5 \leq x \leq 0.5 \\ -\sqrt{\frac{-x}{4(1-x)}} & \text{if } x > 0.5 \end{cases} \tag{B.10}$$

an approximate sampling can be obtained. A comparison between this method and the method from [Botev, 2015] can be found in Chapter 4.

C

LQ Optimal Control

C.1 Partitioning of the Riccati Equation

Given the dynamic Riccati equation and optimal feedback gain for the standard deterministic discrete time LQR problem

$$\begin{aligned}
\boldsymbol{S}_k =& \boldsymbol{Q}_k^x + \boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Phi}_k \\
&- (\boldsymbol{Q}_k^{xu} + \boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k)(\boldsymbol{Q}_k^u + \boldsymbol{\Gamma}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k)^{-1} (\boldsymbol{Q}_k^{xu} + \boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k)^T \\
\boldsymbol{S}_N =& \boldsymbol{Q}_N^x \\
\boldsymbol{L}_k =& -(\boldsymbol{Q}_k^u + \boldsymbol{\Gamma}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k)^{-1} (\boldsymbol{Q}_k^{xu} + \boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k)^T
\end{aligned} \tag{C.1}$$

the following partitioning of the system matrices

$$\begin{gathered}
\boldsymbol{S}^k = \begin{bmatrix} \boldsymbol{S}_k^m & \boldsymbol{S}_k^{md} \\ \boldsymbol{S}_k^{md\,T} & \boldsymbol{S}_k^d \end{bmatrix}, \quad \boldsymbol{\Phi}_k = \begin{bmatrix} \boldsymbol{\Phi^m}_k & \boldsymbol{\Gamma^m}_k \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{\Gamma}_k = \begin{bmatrix} \boldsymbol{\Gamma^m}_k \\ \boldsymbol{0} \end{bmatrix} \\
\boldsymbol{Q}_k^x = \begin{bmatrix} \widehat{\boldsymbol{Q}}_k^x & \boldsymbol{Q}_k^{xd} \\ \boldsymbol{Q}_k^{xd\,T} & \boldsymbol{Q}_k^d \end{bmatrix}, \quad \boldsymbol{Q}_k^{xu} = \begin{bmatrix} \widehat{\boldsymbol{Q}}_k^{xu} \\ \boldsymbol{Q}_k^{du} \end{bmatrix}
\end{gathered} \tag{C.2}$$

and assumption that the linear system given by $\boldsymbol{\Phi}_k^m$ and $\boldsymbol{\Gamma}_k^m$ is controllable, $\boldsymbol{L}_k$ will still converge even though the system given by $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Gamma}_k$ isn't controllable and $\boldsymbol{S}_k$ diverges as $N \to \infty$. By direct insertion of the partitioned matrices and carrying out

of the block multiplication

$$\begin{aligned}
\boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Phi}_k &= \begin{bmatrix} \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T & \boldsymbol{0} \\ \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{S}_{k+1}^m & \boldsymbol{S}_{k+1}^{md} \\ \boldsymbol{S}_{k+1}^{md\,T} & \boldsymbol{S}_{k+1}^d \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k & \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k & \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k + \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^{md} \\ \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k + \boldsymbol{S}_{k+1}^{md\,T} \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k & \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k + \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^{md} \\ & \quad + \boldsymbol{S}_{k+1}^{md\,T} \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k + \boldsymbol{S}_{k+1}^d \end{bmatrix} \\
\boldsymbol{\Phi}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k &= \begin{bmatrix} \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T & \boldsymbol{0} \\ \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{S}_{k+1}^m & \boldsymbol{S}_{k+1}^{md} \\ \boldsymbol{S}_{k+1}^{md\,T} & \boldsymbol{S}_{k+1}^d \end{bmatrix} \begin{bmatrix} \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k \\ \boldsymbol{0} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k \\ \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k + \boldsymbol{S}_{k+1}^{md\,T} \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k \end{bmatrix} \\
\boldsymbol{\Gamma}_k^T \boldsymbol{S}_{k+1} \boldsymbol{\Gamma}_k &= \begin{bmatrix} \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T \\ \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{S}_{k+1}^m & \boldsymbol{S}_{k+1}^{md} \\ \boldsymbol{S}_{k+1}^{md\,T} & \boldsymbol{S}_{k+1}^d \end{bmatrix} \begin{bmatrix} \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k
\end{aligned} \tag{C.3}$$

the following expressions can be gotten for the partitions of $\boldsymbol{S}_k$.

$$\begin{aligned}
\boldsymbol{S}_N^m =& \widehat{\boldsymbol{Q}}_N^x \\
\boldsymbol{S}_k^m =& \widehat{\boldsymbol{Q}}_k^x + \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k - (\widehat{\boldsymbol{Q}}_k^{xu} + \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k) \\
& \cdot (\boldsymbol{Q}_k^u + \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k)^{-1} (\widehat{\boldsymbol{Q}}_k^{xu} + \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k)^T \\
\boldsymbol{S}_N^{md} =& \boldsymbol{Q}_N^{xd} \\
\boldsymbol{S}_k^{md} =& \boldsymbol{Q}_k^{xd} + \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k + \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^{md} - (\widehat{\boldsymbol{Q}}_k^{xu} + \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k) \\
& \cdot (\boldsymbol{Q}_k^u + \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k)^{-1} (\boldsymbol{Q}_k^{du} + \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k + \boldsymbol{S}_{k+1}^{md\,T} \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k)^T \\
=& \boldsymbol{Q}_k^{xd} + \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k + \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^{md} \\
& + \boldsymbol{L}_k^{mT} (\boldsymbol{Q}_k^{du} + \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k + \boldsymbol{S}_{k+1}^{md\,T} \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k)^T \\
=& \boldsymbol{C}(\boldsymbol{S}_{k+1}^m) + (\boldsymbol{\Phi}^{\boldsymbol{m}}{}_k + \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k \boldsymbol{L}_k^m)^T \boldsymbol{S}_{k+1}^{md}
\end{aligned} \tag{C.4}$$

Where $\boldsymbol{C}(\boldsymbol{S}_{k+1}^m)$ gathered all other terms that do not depend on $\boldsymbol{S}_{k+1}^{md}$ and

$$\begin{aligned}
\boldsymbol{L}_k =& [\boldsymbol{L}_k^m \; \boldsymbol{L}_k^d] \\
\boldsymbol{L}_k^m =& -(\boldsymbol{Q}_k^u + \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k)^{-1} (\widehat{\boldsymbol{Q}}_k^{xu} + \boldsymbol{\Phi}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k)^T \\
\boldsymbol{L}_k^d =& -(\boldsymbol{Q}_k^u + \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k)^{-1} \\
& \cdot (\boldsymbol{Q}_k^{du} \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k^T \boldsymbol{S}_{k+1}^m \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k + \boldsymbol{S}_{k+1}^{md\,T} \boldsymbol{\Gamma}^{\boldsymbol{m}}{}_k)^T
\end{aligned} \tag{C.5}$$

The expression for $\boldsymbol{L}_k^m$ only depends on $\boldsymbol{S}_k^m$ and is the exact same as for a LQR problem for just the reduced system given by $\boldsymbol{\Phi}_k^m$ and $\boldsymbol{\Gamma}_k^m$. Since that system is

assumed controllable, $\boldsymbol{S}_k^m$ will converge as $N \to \infty$ and since $\boldsymbol{S}_k^m$ is independent of $\boldsymbol{S}_k^{md}$, it will converge regardless of $\boldsymbol{S}_k^{md}$. After convergence of $\boldsymbol{S}_k^m$, all matrices in the expression for $\boldsymbol{S}_k^{md}$ will turn into constant, turning the expression to a affine fixed point iteration over $\boldsymbol{S}_k^{md}$.

The Jacobian of the affine function is $\boldsymbol{\Phi^m}_k + \boldsymbol{\Gamma^m}_k \boldsymbol{L}_k^m$ which is the system matrix for the closed loop system given by $\boldsymbol{\Phi}_k^m$, $\boldsymbol{\Gamma}_k^m$ and $\boldsymbol{L}_k^m$ which is known to be stable from the standard theory of LQR-control given some further light restrictions on the weighs $\boldsymbol{Q}_k$. The eigenvalues of $\boldsymbol{\Phi^m}_k + \boldsymbol{\Gamma^m}_k \boldsymbol{L}_k^m$ is then inside the unit circle. Banach fixed-point theorem then gives the convergence of the fixed point iteration since the affine function then is a contraction. The full expression for $\boldsymbol{S}_k^d$ is left out for brevity but it's easily seen that the same can not be said for $\boldsymbol{S}_k^d$ even after the convergence of $\boldsymbol{S}_k^m$ and $\boldsymbol{S}_k^{md}$. The expression will be on the form

$$\boldsymbol{S}_k^d = \boldsymbol{C}'(\boldsymbol{S}_{k+1}^m, \boldsymbol{S}_{k+1}^{md}) + \boldsymbol{S}_{k+1}^d \tag{C.6}$$

The Jacobian of this expression clearly has eigenvalues 1 so convergence can not be guaranteed. However, since it's clear that $\boldsymbol{L}_k$ doesn't depend on $\boldsymbol{S}_k^d$, it will converge as long as $\boldsymbol{S}_k^m$ and $\boldsymbol{S}_k^{md}$ converges.

Bibliography

Åström, K. J. (1983). "Theory and applications of adaptive control—a survey". *Automatica* **19**:5, pp. 471–486.

Åström, K. J. and B. Wittenmark (2008). *Adaptive Control (2 rev. Dover ed.)* Dover Publications.

Åström, K. J. and B. Wittenmark (2011). *Computer-Controlled Systems*. Dover Publications.

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

Bayard, D. S. and A. Schumitzky (2010). "Implicit dual control based on particle filtering and forward dynamic programming". *International journal of adaptive control and signal processing* **24**:3, p. 155.

Bellman, R. E. (1957). *Dynamic Programming*. Princeton University Press.

Berg, J. van den (2016). "Extended LQR: locally-optimal feedback control for systems with non-linear dynamics and non-quadratic cost". In: *Robotics Research*. Springer, pp. 39–56.

Bertsekas, D. P. (2005). "Dynamic programming and suboptimal control: a survey from adp to mpc". *European Journal of Control* **11**:4-5, pp. 310–334.

Blom, G., J. Enger, G. Englund, J. Grandell, and L. och Holst (2005). *Sannolikhetsteori och statistikteori med tillämpningar*. Studentlitteratur Lund.

Botev, Z. (2015). *Truncated normal generator*. MATLAB Central File Exchange. URL: `http://se.mathworks.com/matlabcentral/fileexchange/53180-truncated-normal-generator` (visited on 01/02/2017).

Botev, Z. (2017). "The normal law under linear restrictions: simulation and estimation via minimax tilting". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **79**:1, pp. 125–148.

Carpenter, J., P. Clifford, and P. Fearnhead (1999). "Improved particle filter for nonlinear problems". *IEE Proceedings-Radar, Sonar and Navigation* **146**:1, pp. 2–7.

Curry, R. E. (1970). *Estimation and control with quantized measurements*. MIT press.

Douc, R. and O. Cappé (2005). "Comparison of resampling schemes for particle filtering". In: *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*. IEEE, pp. 64–69.

Doucet, A., N. de Freitas, and N. Gordon (2001). *Sequential Monte Carlo Methods in Practice. Series Statistics For Engineering and Information Science*. Springer New York.

Dreyfus, S. E. (1965). *Dynamic programming and the calculus of variations*. Vol. 21. Academic Press New York.

Dvijotham, K and E Todorov (2012). "Linearly solvable optimal control". *Reinforcement learning and approximate dynamic programming for feedback control*, pp. 119–141.

Feldbaum, A. (1960). "Dual control theory. i". *Avtomat. i Telemekh.* **21**:9, pp. 1240–1249.

Glad, T. and L. Ljung (2000). *Control theory*. CRC press.

Gordon, N. J., D. J. Salmond, and A. F. Smith (1993). "Novel approach to nonlinear/non-gaussian bayesian state estimation". In: *IEE Proceedings F (Radar and Signal Processing)*. Vol. 140. 2. IET, pp. 107–113.

Karlsson, R. and F. Gustafsson (2005). "Particle filtering for quantized sensor information". In: *Signal Processing Conference, 2005 13th European*. IEEE, pp. 1–4.

Li, W. and E. Todorov (2004). "Iterative linear quadratic regulator design for nonlinear biological movement systems." In: *ICINCO (1)*, pp. 222–229.

Pitt, M. K. and N. Shephard (1999). "Filtering via simulation: auxiliary particle filters". *Journal of the American statistical association* **94**:446, pp. 590–599.

Rajamäki, J., K. Naderi, V. Kyrki, and P. Hämäläinen (2016). "Sampled differential dynamic programming". In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, pp. 1402–1409.

Schön, T. B. (2010). "Solving nonlinear state estimation problems using particle filters-an engineering perspective". *Technical Report. Automatic Control at Linkopings Universitet*.

Smith, A. F. and A. E. Gelfand (1992). "Bayesian statistics without tears: a sampling–resampling perspective". *The American Statistician* **46**:2, pp. 84–88.

Stahl, D. and J. Hauth (2011). "Pf-mpc: particle filter-model predictive control". *Systems & Control Letters* **60**:8, pp. 632–643.

Sun, W., J. van den Berg, and R. Alterovitz (2016). "Stochastic extended LQR for optimization-based motion planning under uncertainty". *IEEE Transactions on Automation Science and Engineering* **13**:2, pp. 437–447.

Todorov, E. (2008). "General duality between optimal control and estimation". In: *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, pp. 4286–4292.

Todorov, E. (2009a). "Compositionality of optimal control laws". In: *Advances in Neural Information Processing Systems*, pp. 1856–1864.

Todorov, E. (2009b). "Efficient computation of optimal actions". *Proceedings of the national academy of sciences* **106**:28, pp. 11478–11483.

Todorov, E. and W. Li (2005). "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems". In: *American Control Conference, 2005. Proceedings of the 2005*. IEEE, pp. 300–306.

Toussaint, M. (2009). "Robot trajectory optimization using approximate inference". In: *Proceedings of the 26th annual international conference on machine learning*. ACM, pp. 1049–1056.

Tse, E. and Y. Bar-Shalom (1975). "Generalized certainty equivalence and dual effect in stochastic control". *IEEE Transactions on Automatic Control* **20**:6, pp. 817–819.

Van Den Berg, J., S. Patil, and R. Alterovitz (2012). "Motion planning under uncertainty using iterative local optimization in belief space". *The International Journal of Robotics Research* **31**:11, pp. 1263–1278.

Wittenmark, B. (2002). "Adaptive dual control". In: *Control Systems, Robotics and Automation, Encyclopedia of Life Support Systems (EOLSS), Developed under the auspices of the UNESCO*. Eolss Publishing CO. Ltd.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER´S THESIS

Date of issue
June 2017

Document Number
ISRN LUTFD2/TFRT--6031--SE

Author(s)
Martin Vikström Morin

Supervisor
Anton Cervin, Dept. of Automatic Control, Lund University, Sweden
Bo Bernhardsson, Dept. of Automatic Control, Lund University, Sweden (examiner)

Sponsoring organization

Title and subtitle

Optimal Control under Quantised Measurements
A Particle Filter and Reduced Horizon Approach

Abstract

This thesis covers the optimal control of stochastic systems with coarsely quantised measurements. A particle filter approach is used both for the estimation and control problem. Three main families of particle filters are examined for state estimation, standard SIR filters, SIR filters with generalised sampling and auxiliary filters. A couple of different proposal distributions and weight functions were examined for the generalised SIR and auxiliary filter respectively. The choice of proposal distribution had the greatest impact on performance but the unrivalled best filter was achieved with a combination of generalised sampling and the auxiliary particle filter. For the problem of control the particle filter was used for cost-to-go evaluation by forward simulation in time. Simplifications of the full dynamic programming problem were done by reducing the time horizon resulting in M-measurement feedback policies and a new M-measurement cost feedback policy. One-measurement feedback and M-measurement cost feedback was examined for $M \leq 4$ and although probing behaviour was observed none of the examined controllers managed to outperform a certainty equivalent controller.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-115

Recipient's notes

Security classification