

A comparative performance evaluation of nonlinear observers for a fed-batch evaporative crystallization process

I. Mora Moreno

Master of Science Thesis

Supervisors:

prof.dr.ir. P.M.J. Van den Hof (Paul)

ir. A.E.M. Huesman (Adrie)

dr.ir. A. Mesbah (Ali)

A comparative performance evaluation of nonlinear observers for a fed-batch evaporative crystallization process

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Mechanical Engineering at Delft
University of Technology

I. Mora Moreno

May 26, 2011

Abstract

Different nonlinear observers are compared throughout this work where they are part of an NMPC framework to control a fed-batch crystallization process. We study which observer-optimizer pair offers the best control performance while maintaining adequate computational burden so that a posterior real-time implementation is feasible. At the same time, the relationship between state estimation accuracy and control performance is covered. Along the way we distinguish between stochastic and deterministic observers and compare which class is more suitable for our case study. The observers we make use of are: the moving horizon estimator (MHE), a nonlinear version of a Luenberger observer (extended Luenberger observer, ELO) and nonlinear variants of the Kalman filter such as extended Kalman filter (EKF), unscented Kalman filter (UKF) and ensemble Kalman filter (EnKF). Special variants of UKF and EKF that make use of a non constant system covariance matrix, which according to some literature is suitable to describe uncertainty distribution in batch processes, are also included in the analysis.

The analysis focuses on how four main error sources such as unmeasured disturbances, uncertain initial conditions, model mismatch, and stochastic disturbances may impact observer estimation accuracy as well as their repercussion on control effectiveness and consequently on process performance. Results show that unmeasured disturbances are the most detrimental to observer and process performance in our case study. In spite of this finding, we present a methodology to tackle and solve this problem.

All the analysis is first made under an open-loop configuration and then moves onto a closed-loop setup. All testing is based on computer simulations of the crystallization process. The evaluation criterion is based on the magnitude of a normalized root-mean squared error throughout 50 batch runs. The results are then used to identify if a link between estimation accuracy and control performance exists. The computational burden is also evaluated along 50 batch simulations, and is measured on the basis of CPU time required by every observer at every estimation stage.

Table of Contents

1	Introduction and Problem Statement	1
1-1	Introduction	1
1-2	Problem statement	3
2	Nonlinear Model Predictive Control	5
2-1	Systems control	5
2-2	Nonlinear Model Predictive Control (NMPC)	6
3	Observers	9
3-1	The Observation problem	9
3-2	Definition of an observer and its relevance in NMPC	11
3-3	Observers	13
3-3-1	Extended Kalman Filter (EKF)	15
3-3-2	Unscented Kalman Filter (UKF)	20
3-3-3	Ensemble Kalman Filter (EnKF)	24
3-3-4	Moving Horizon Estimator (MHE)	27
3-3-5	Extended Luenberger Observer (ELO)	30
3-4	Observer Design	32
3-5	Summary	35
3-6	Remarks	36
4	Case study	37
4-1	Crystallization	37
4-2	Description of the case study	42
4-3	Observability analysis	45
4-4	Summary	50
4-5	Remarks	50

5 Simulation Results	53
5-1 Analyzed Scenarios	53
5-2 Evaluation Criteria	56
5-3 Open-loop analysis	58
5-4 Closed-loop analysis	63
5-5 Disturbance suppression	76
5-6 Summary	88
5-7 Remarks	88
6 Conclusions and Recommendations	91
Bibliography	95
A Open-loop case: complementary figures	99
B Closed-loop nominal case: complementary figures	103
C Closed-loop unknown initial conditions case: complementary figures	107
D Closed-loop systematic measurement error case: complementary figures	111
E Closed-loop model mismatch case: complementary figures	115
F Disturbance rejection	121
F-1 Disturbance rejection for linear systems	121
G CSD, nominal case vs. general case: complementary figures	131

Acknowledgements

I would like to express sincere gratitude to:

- prof.dr.ir. P.M.J. Van den Hof for being my main tutor,
- ir. Adrie Huesman for keeping me in the right track throughout my thesis work while offering fruitful and clever feedback whenever necessary,
- dr.ir. Ali Mesbah as he was a model of supervisor, very talented, supportive, and I would also like to stress the fact that he was very patient with me. He dedicated a lot of time polishing my work and literally towed me up to the finish line in order for me to obtain my degree. I will always be in debt with him,
- I would also like to thank the Universidad Autonoma del Estado de Mexico as they funded part of my studies.

Delft, University of Technology
May 26, 2011

I. Mora Moreno

*With love, to my parents and my brother ...
thank you for giving me much more than what I deserved.*

Introduction and Problem Statement

1-1 Introduction

Crystallization is the formation of solid crystals precipitating from a solution or a melt. Its importance relies on the fact that more than 80% of the substances used in pharmaceuticals, fine chemicals, agrochemicals, food and cosmetics are produced in their solid form. In fact, few branches of the chemical and process industries do not, at some stage, employ crystallization for production or separation purposes. Even more, highly demanded crystalline substances such as: sodium chloride and sucrose, in the food sector; ammonium nitrate, potassium chloride, ammonium phosphates and urea, among fertilizers, for instance, have very high production rates worldwide and crystallization is used in their making. Even though compared to the latter goods, crystalline products for the pharmaceutical, organic fine chemical, and dye industries are produced in relatively low amounts, they still represent a valuable and important industrial sector. It is also a widely used production and purification method of solid materials in the chemical and process industries. Even more, it is a key operation in, for example, desalination of seawater, concentration of fruit juices, removal of unwanted materials or recovery of valuable constituents in many industrial processes. It is also increasingly employed in production of material for the electronics industry, Mullin [26].

Crystallization is governed by complex, interacting variables. It is a simultaneous heat and mass transfer process with a strong dependence on fluid and particle mechanics. It occurs in a multiphase, multicomponent system and involves particulate solids whose characteristics vary with time, Mullin [26]. Despite this complexity, as in any other process, there are specifications that production must fulfill. Some of these specifications may be required for downstream operation, like filtration or drying for instance, or for acceptance of the client. In the case of crystals, specifications may be related to concepts such as crystal size distribution, shape of the crystals, purity, or some others. In order to assure product quality but simultaneously maximize productivity, it should be possible to operate systems on the boundaries of the allowed operating region. Such goal can only be achieved by adequate process control, Nagy [28].

Control of crystallization processes is a challenging task due to the complexity of the process. Describing the dynamics of a crystallization process requires nonlinear models. For this reason, Nonlinear Model Predictive Control (NMPC) has been the sticking point on these applications, and was the choice for this project.

Our work focuses on the control of an evaporative 75-liter fed-batch crystallizer where ammonium sulphate dissolved in water is the crystallization system.

The process goal is to produce crystals with some desired crystal size distribution. In order to achieve this, the controller must keep the crystals' growth rate as close as possible to a predetermined value.

Several studies concerning this setup have been done in the past, see Mesbah et al. [23], Mesbah et al. [24], where the model describing the system dynamics was deterministic. The dynamics of the crystallizer are described by a population balance equation, consisting of a set of partial differential equations. However, the characteristics of the problem posed by the 75-liter crystallizer allow transforming the full population balance equation into a less complex model, although still nonlinear, known as the moment model which consists of a set of nonlinear differential and algebraic equations. The moment model is expected to be complex enough to be used as a benchmark to determine the most suitable observer in case a more complicated system model had to be used. It is however still possible to improve previous studies. First, creating a more realistic scenario by perturbing the system with different error sources. Such error sources can be divided in stochastic error, unmeasured disturbances, uncertain initial conditions and model mismatch. Introduction of error sources of random nature force us to introduce stochastic variables in the process model, which in paper pose a tougher challenge for the controller and intrinsically for the observer-optimizer pair. This motivates selection and testing of observers other than deterministic. For some alternatives the reference, Mora Moreno [25] may offer an overview of available choices.

The observers we will make use of are:

- Extended Kalman filter (EKF)- in literature it appears to be the default option when a nonlinear observer is needed. It is also claimed to be computationally cheap, somehow. It has shown however, some limitations since its algorithm relies on linearization of the system model.
- Unscented Kalman filter (UKF)- conceived by Julier S. J. [19], it is claimed to be more accurate than EKF and does not require differentiation to generate estimates while maintaining a computational burden in the same order than EKF. No applications in the area of batch crystallization had been realized, so it should be interesting to include it in our study.
- Ensemble Kalman filter (EnKF)- its main applications were found in cases involving complex nonlinear models with large number of variables, and like UKF, does not make use of derivatives in their calculations either.
- Extended Luenberger observer (ELO)- previous works concerning our case study, Mesbah et al. [24], had dealt with an extended Luenberger observer, although the application was in a non stochastic scenario, yet we selected this deterministic observer for comparison with the others.

- moving horizon estimator (MHE)- As this type of observer is based on an optimization procedure it seemed appealing to evaluate how well it could deal with the stochastic factor.

Concerning UKF and EKF, modified variants of both observers are available. These variants make use of a non-constant system covariance matrix. Computation of such matrices is based on an algorithm developed by Valappil and Georgakis [43]. The assumptions made to come up with the algorithm are suitable for batch crystallization, so it is appealing to study their applicability.

Finally, in the references concerning previous studies of the 75-liter crystallizer it has already been shown the importance of a closed-loop control strategy, over an open-loop approach, to adequately achieve quality and production goals. Nevertheless, since we will investigate the effects of using non-deterministic observers, and we will also perturb the system with different error sources, we will first take on an open-loop scenario. There, all the observers will be tested to determine how good an open-loop approach is, and determine if it is necessary to implement a closed-loop control strategy once again. Both scenarios will be evaluated under the same criteria introduced in the next section.

1-2 Problem statement

We may formulate our problem statement as “to select the best observer to be used in an NMPC framework, such that predefined control objectives are met while real-time implementation of the observer-optimizer pair remains feasible”.

It should be noticed that our optimizer is a fixed element of the NMPC framework, that is, all observers will be coupled to the same optimizer.

The optimizer requires information from the system. A first impulse would be to measure any required quantity directly on the system. Unfortunately, due to economical, physical or technological reasons it is not always possible to perform measurements of any desired variables. Under these circumstances, it is only possible to estimate their value. Fortunately, estimating system variables may be done by exploiting the relations binding the state variables to the inputs, outputs and their time derivatives in terms of measurable quantities. This allows to know the state of a system without actually measuring it. A mathematical tool which offers a solution to this problem is known as an observer.

Observation, however, requires the system fulfills a property known as observability. This property must be, and will be, assessed before any observer implementation.

Part of the assesment to determine the best observer-optimizer pair will be based on analyzing the influence each error source has on the estimation capabilities of every observer. Thereby, our system will be perturbed with every single error source we mentioned before, one at a time. The effects will be described and we will explain the reasons behind such effects. A final exercise where all error sources will be simultaneously applied will allow us evaluate to what extent the superposition principle holds when trying to interpret the results and should ultimately be consider as the worst case scenario that could be faced in a real implementation.

It is of special interest to determine which error source(s) is/are more detrimental to observer and control performance. In this way we should be able to describe up to which point an

observer may account for different error sources and will let us be aware of the limitations when a real-time implementation is carried out.

Even more, since fulfilling control goals leads to meet or exceed product specifications, which is what really matters from a practical point of view, an important part of our assessment is based on whether our observer-optimizer pair achieves adequate system control. Control performance will be evaluated by looking at the tracking error of a desired crystal growth trajectory. Then an interesting question derives from the latter exercise, that is whether estimation accuracy plays a role in control performance or not. So, is it necessary to spend time building an observer that delivers accurate estimates, or are rough approximations of the state variables enough to achieve good control?

We will make use of computer simulations to recreate all the cases previously described. The estimation accuracy will be evaluated on the basis of a normalized root-mean squared error (NRMSE) throughout 50 simulations.

Again, in the case of tracking error we will assess its magnitude based on an NRMSE over 50 simulations of this quantity.

This approach should suffice to find a correlation between estimation accuracy and control performance, if anyone exists.

Concerning real-time feasibility of the controller, we will determine the computational burden of each observer, as well as that for the observer-optimizer pair (that is, the controller).

Computational times will be recorded every time an observer calculates state estimates. We will plot the average, taken over 50 simulations, of the required CPU time at each step. This will allow us verify that there are no large peaks in the simulation time that would endanger a successful real-time implementation.

All simulations stated above will take place in both, open- and closed-loop scenarios.

The outline of the report is as follows, Chapter 2 gives an overview of NMPC and how observers integrate in this framework, we highlight their importance as part of the control strategy. Chapter 3 describes the selected observers, it also discusses the design of the observers, Chapter 4 thoroughly describes and discusses the case study, there we present the model describing the system under investigation together with the specifications that must be met by the controlled system. We also present an observability analysis required before attempting to design an observer. In Chapter 5 we present, analyze and discuss all the results generated by a computer simulator of the controlled system. Finally we give our conclusions and recommendations in Chapter 6. Appendices A to E present complementary figures and briefly discuss the findings of the simulations for all the remaining error sources not discussed in Chapter 5. Appendix F gives a complete overview of disturbance rejection by means of integral action and also by the usage of augmented states, two concrete examples are given and compared for each methodology, both for linear systems. Finally, Appendix G gives complementary figures of the CSD for all the filters in the general case.

Nonlinear Model Predictive Control

In this chapter we give a short introduction to MPC and link it with NMPC as the latter is required for our case study. We highlight important features characterizing MPC as well as the elements that make it a interesting option for systems as that analyzed in this work.

2-1 Systems control

Independently of the field where a control issue arises, three different elements are common to all cases; a system (with an arbitrary number of elements and degrees of complexity) whose behavior wants to be molded according to certain parameters, those parameters might be set by the person facing the challenge, and may be thought as goals that must be achieved by the system. Some others may be set by the environment surrounding the system, which may be thought as limitations or constraints as the designer cannot manipulate them, or a combination of the two. There will also be disturbances perturbing the system which according to their intensity may have, or not, an impact on the system. Once the goal, constraints and disturbances have been recognized/set, the designer needs to choose a control strategy to be applied. To come up with an adequate choice, the selection should be based on the characteristics of the system the designer is dealing with. For instance, as the system in front exhibits fast dynamics, so will need the control strategy. If constraints are present, the control strategy must be capable of dealing with such restrictions. Besides, it is also important that perturbations present on the system, which the designer might or might not be aware of, can be overcome.

For our particular application concerning batch crystallization processes we may say that they are well known for showing complex dynamics requiring nonlinear models to describe them and acquire reasonable results, the number of variables to describe the system behavior is usually large compared to other applications, they also exhibit dominant time constants of large magnitude, meaning they can be considered to have slow dynamics. Other characteristics include the uncertainty in the knowledge of system parameters, the inherent variability of those parameters along production and presence of disturbances.

The characteristics just mentioned lead us to select a control strategy known as nonlinear model predictive control (NMPC), this methodology is further described in the following sections, however we start off by introducing model predictive control (MPC) differing from NMPC in the fact that only linear models are used.

2-2 Nonlinear Model Predictive Control (NMPC)

van den Boom and Backx [44] state that MPC originated in the late seventies and has developed considerably since then. The term MPC does not designate a specific control strategy but rather an ample range of control methods which make explicit use of a model of the process (thereby its name) to obtain the control signal by minimizing an objective function. MPC may also be described as a control scheme where it is possible to identify five items as part of the design procedure:

- 1 process model and disturbance model: this allows to describe the system dynamics as well as disturbance(s) acting on the system. The models may be the output of a first-principle, an empirical or a combined approach
- 2 performance index: is related to a cost function which sets a criterion that must be totally fulfilled, or at least within certain boundaries defined by the designer in order to achieve certain system behavior
- 3 constraints: mathematical representation of actual constraints in the system
- 4 optimization: related to the solution of the cost function imposed by the performance index
- 5 receding horizon principle: the optimization part takes place over a certain period ahead in time, the receding horizon uses only part of the computed solution and establishes that optimization must be redone after a predetermined period.

Figure 2-1 shows the complete picture of a system controlled under the MPC scheme where the elements listed before can be identified.

It is possible to mathematically represent all the elements of MPC mentioned before as follows,

$$\begin{aligned}
 \min J \quad & s.t. \\
 & \tilde{f} = 0 \\
 & \tilde{h} = 0 \\
 & \tilde{\phi} = 0 \\
 & \tilde{\psi} \leq \tilde{\Psi}
 \end{aligned} \tag{2-1}$$

where J represents the cost function to be minimized, \tilde{f} represents a set of equations describing the dynamics of the system, \tilde{h} represents the output function, $\tilde{\phi}$ represents the set of equality constraints and $\tilde{\psi}$ the set of inequality constraints that must be fulfilled. The variables upon

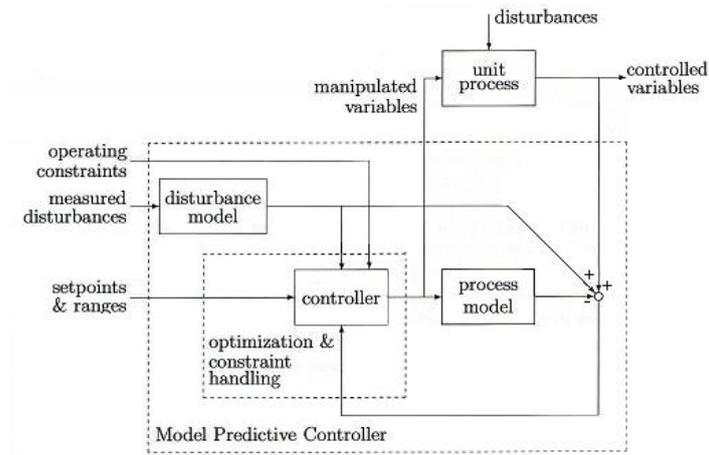


Figure 2-1: MPC scheme, as presented in van den Boom and Backx [44].

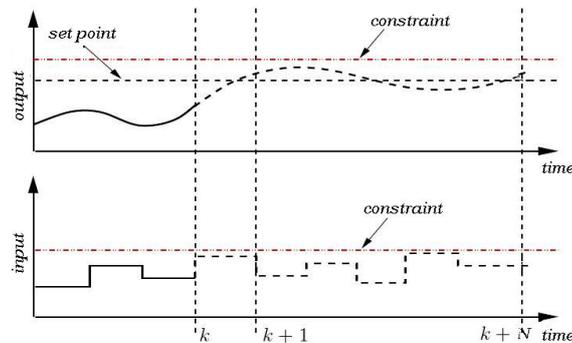


Figure 2-2: receding horizon principle.

which the functions depend are excluded as they depend on the representation of the models, that is, an input output model has a different structure than a state space model.

In Figure 2-2 we observe the evolution of a discrete-time system starting at time k , we see the history of the system output and input (solid lines), and their respective predictions given by an MPC scheme. In the upper graph we see a set point line which represents a trajectory that must be followed by the system; however, the output should not exceed a certain value marked by the constraint therein. In the lower graph we see a control input profile product of solving an optimization problem from time $k + 1$ up to $k + N$, subject to an input constraint also depicted, such that, by applying these inputs to the system it will generate a trajectory given by the broken line. In the receding horizon we obtain a control input sequence over N time steps, nonetheless, we only implement the result for the first time step and restart the process all over again.

According to van den Boom and Backx [44], some of the features that make MPC an attractive tool for control are:

- it can handle multivariable systems,
- it can handle processes that may contain one or more drawbacks like instability, large

time delays, non-minimum phase behavior,

- it is an easy-to-tune method, as there are only a few basic parameters to be tuned,
- processes have limitations, for instance in terms of actuation a valve cannot exceed a determined capacity. MPC can handle that and other constraints in a systematic way during the design and implementation of the controller,
- finally, MPC can handle structural changes, such as sensor and actuator failures, changes in system parameters and system structure by adapting the control strategy on a sample-by-sample basis.

From all what we have mentioned so far, perhaps the main reason for the popularity of this control strategy is due to its constraint-handling capabilities and the easy extension to multivariable systems.

MPC basically refers to the methodologies where only linear equations are used to describe everything in the problem formulation, beginning by the plant and disturbances and including the constraints, nevertheless, there are many applications that require usage of nonlinear models. This variant is known as Nonlinear Model Predictive Control, and its functioning is similar to that in MPC, even though modifications to cope with the nonlinearities must be applied. NMPC possesses the same advantages that make MPC so attractive for certain applications, plus, it incorporates the advantage of working with nonlinear systems. This justifies the selection of NMPC for the present case study and will be more evident once we introduce the model in the next chapter.

Chapter 3

Observers

We begin this chapter by giving an overview of what is known as an observation problem. We make a distinction between two cases, one involving stochastic behavior and another one without it. Then we continue by defining an observer, a classification found in literature is introduced, and based on the needs of the case study concerning us we present the selected observers that later on will be used in this work. A short summary of the algorithms, main characteristics and tuning issues are presented for each observer.

3-1 The Observation problem

Before we proceed to mathematically formulate the observation problem, let us make a distinction of two groups of models, namely, deterministic and stochastic ones. They may be distinguished by the fact that a stochastic model contains random variables, that is, variables whose exact value is not exactly known but may be described by a probability density function. It follows that a deterministic model does not contain any random variable. Deterministic models can be thought as a subcase of a stochastic one when the uncertainty attached to it is zero. Therefore by taking on the stochastic problem we ensure covering both cases.

Given a (nonlinear) state-space discrete-time system representation,

$$x_{k+1} = f(x_k, u_k, v_k) + d_k, \quad (3-1)$$

$$y_k = h(x_k, u_k, w_k) \quad (3-2)$$

with

$$x_k \in \mathcal{R}^{n \times 1} \quad (3-3)$$

$$u_k \in \mathcal{R}^{q \times 1} \quad (3-4)$$

$$y_k \in \mathcal{R}^{p \times 1} \quad (3-5)$$

$$v_k \in \mathcal{R}^{q \times 1} \quad (3-6)$$

$$w_k \in \mathcal{R}^{p \times 1} \quad (3-7)$$

$$d_k \in \mathcal{R}^{n \times 1} \quad (3-8)$$

where x , u , y , v , w are the state, input, output, system noise, measurement noise and disturbance vectors respectively. System and measurement noise can be interpreted as the uncertainty faced when trying to measure either a state variable or an output variable. Depending on the characteristics of the modeled system the probability density function of each noise variable may vary; here we assume that noise vectors v_k and w_k , are zero-mean and

$$E[v_i v_j^T] = \delta_{ij} Q_{ij} \quad (3-9)$$

$$E[w_i w_j^T] = \delta_{ij} R_{ij} \quad (3-10)$$

$$E[v_i w_j^T] = 0; \quad \forall i, j. \quad (3-11)$$

with

$$Q_{ij} \in \mathcal{R}^{q \times q} \quad (3-12)$$

$$R_{ij} \in \mathcal{R}^{p \times p} \quad (3-13)$$

$$\delta_{ij} = 1 \quad \text{if } i = j \quad \text{and} \quad \delta_{ij} = 0 \quad \text{otherwise}$$

where Q_{ij} , R_{ij} , δ_{ij} are the system noise covariance matrix, measurement noise covariance matrix and Kronecker's delta, respectively. Disturbances can be deterministic or stochastic, measured or unmeasured, that is to say, it might or might not be possible to determine the magnitude of some of the disturbances acting on the system, this information should help to more faithfully represent reality via our system model. Functions f and h are nonlinear functions, and k is the discrete-time index.

The observation problem can be mathematically described as follows. The objective is to find the best estimate (according to a certain criterion) of the variable x_{k+1} given observations y up to and including time k . This issue is regarded as a sequential probabilistic inference problem.

The hidden system state x_{k+1} , with initial distribution $p(x_0)$, evolves over time.

The state transition density $p(x_k | x_{k-1})$ is fully specified by f and the process noise distribution $p(v_k)$, whereas h and the observation noise distribution $p(w_k)$ fully specify the observation likelihood $p(y_k | x_k)$. The exogenous input to the system, u_k is assumed known.

According to Kalman [20], consider the following situation,

- for a specific case that equations (3-1) and (3-2) represent a linear system, we are given signal x_k and noise ν_k . Only the sum $y_k = x_k + \nu_k$ can be observed. Suppose we have observed and know exactly the values of y_0, \dots, y_k . What can we infer from this knowledge in regard to the unobservable value of the signal x_k at k_a , where k_a may be less than, equal to, or greater, than k ? If $k_a < k$, this is a data smoothing (interpolation) problem. If $k_a = k$ this is called filtering. If $k_a > k$, we have a prediction problem. In this report the filtering case is the only one that will be analyzed. Therefore it must be noted that, since we are doing filtering, we only account for improving the current states and parameters, in fact for our case study we will only estimate state variables.

In a Bayesian framework, the posterior density $p(x_k|Y_k)$ of the state given all the observations $Y_k = \{y_1, y_2, \dots, y_k\}$ constitutes the complete solution to the sequential probabilistic inference problem, and allows us to calculate any “optimal” estimate of the state, such as the conditional mean $x_{k+1} = E[x_k|Y_k] = \int x_k p(x_k|Y_k) dx_k$.

Although this is the optimal recursive solution, it is usually only tractable for linear, Gaussian systems in which case the closed-form recursive solution of the Bayesian integral equations is the well known Kalman filter, Kalman [20]. For most general real-world (nonlinear, non-Gaussian) systems however, the multi-dimensional integrals are intractable and approximate solutions must be used. These include methods such as Gaussian approximations (extended Kalman filter and other variants such as the linear regression Kalman filters, ensemble Kalman filters, etc.), hybrid Gaussian methods (score function extended Kalman filter, Gaussian sum filters), direct and adaptive numerical integration (grid-based filters), sequential Monte Carlo methods (particle filters) and variational methods (Bayesian mixture of factor analyzers) to name but a few.

We should now highlight that for our case study and from a control point of view we are mostly interested in the input-output behavior of a system, therefore, not all the state variables need to be estimated. Given the fact that there may exist different models with the same input-output behavior it turns out that estimating all states might be superfluous. Even more, in the NMPC framework the cost function that must be minimized might not need knowledge of all the states.

3-2 Definition of an observer and its relevance in NMPC

Chapter 2 introduced that an essential element to make use of NMPC is a model describing the system under investigation. As soon as we use a system modeling approach in front of a problem, the issue of observer design arises whenever one needs some internal information from external (directly available) measurements. Those signals of interest roughly include time-varying signals characterizing the system (state variables), constant ones (parameters), and measured or unmeasured external ones (disturbances). This need for internal information can be motivated by various purposes: modeling (identification), monitoring (fault detection), or driving (control) the system, all these being required for keeping a system under control, Besançon [2]. Sometimes it is not possible to perform measurements for a given variable, thus their value must be estimated by means of an available model whose variables are updated with measured data from the actual system. In general indeed, due to sensor limitations (for cost reasons, technological constraints, etc.), the directly measured signals do not coincide with all signals characterizing the system behavior. This makes the observation problem the heart of a general control problem.

Definition of an observer

We may now, more formally and shortly state that **an observer relies on a model, with on-line adaptation based on available measurements, and aiming at information reconstruction, *i.e.* it can be characterized as a model-based, measurement-based, closed-loop information reconstructor.** Figure 3-1 illustrates this concept.

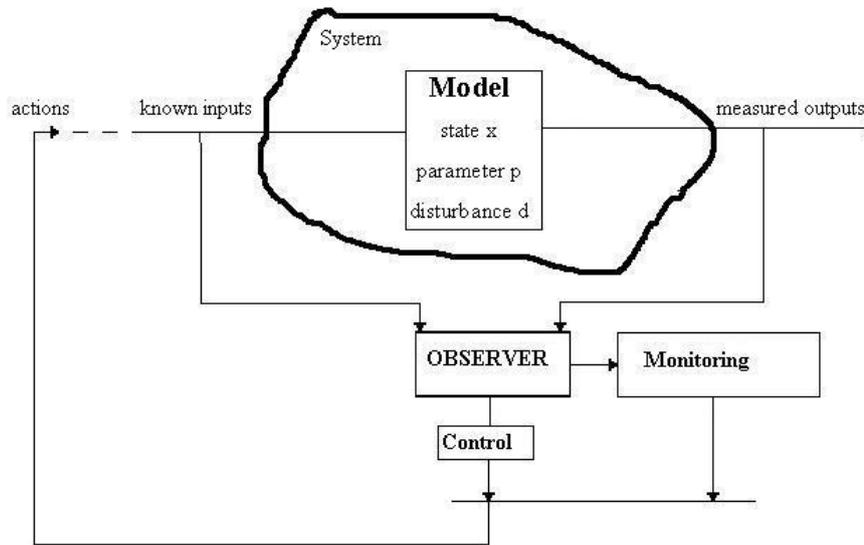


Figure 3-1: System and observer embedded in a control loop. Adapted from Besançon [2]

Basically what an observer does is to use a model to predict values of the system state variables, and relies on measurements to improve those estimates. This improvement is made at the same time instant the measurement was made, that is to say, if we have made an estimation (prediction) of a state variable for time k and we represent it by \hat{x}_k , then we can use an available measurement taken at time k to improve our prediction. Figure 3-2 depicts how the estimation error may be totally/partially compensated for by relying on measurements.

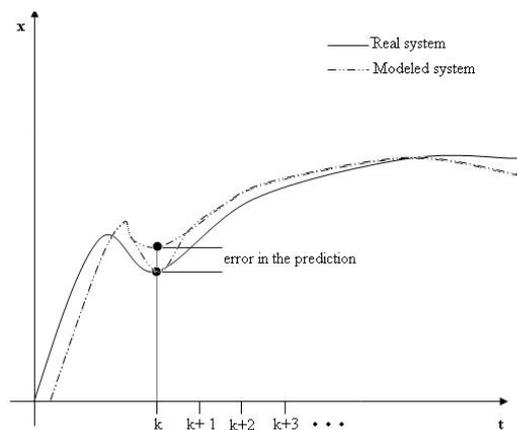


Figure 3-2: Compensation of the error between the predicted value of a system variable and its real value, notice it can only be alleviated in the current time step k .

We are able to decrease estimation errors when measurements are available, but this does not improve a model prediction's capability.

Discrepancies between model predictions and reality are due to error sources such as:

- Model mismatch: the model is not able to faithfully represent the dynamics of the real system.
- Unmeasured disturbances: unwanted inputs that cannot be manipulated, and that may or may not be measured, are acting on the system, therefore drifting it away a prespecified trajectory.
- Measurement error: corrupted measurements lead to apply incorrect inputs
- Unknown initial conditions: the system may start from different points in state-space causing it to evolve in time through different trajectories that might not be desirable.

Disturbances

As disturbances may not be foreseen when controlling a process, they can easily bring down a control strategy. It is important to have an efficient methodology to reject them, that is, make it robust against perturbations. We will now give a short description of a methodology suitable to deal with disturbances in NMPC. Appendix F treats the task of disturbance suppression more deeply.

Disturbances may be modeled at the input, output or some combination; they are, as the the reader might have expected, uncontrollable and can be incorporated in the state estimator, Rawlings and Bakshi [36].

If disturbances are acting on the system, two basic ways to cope with them are,

- based upon system knowledge, propose a model describing the disturbance dynamics and incorporate this model in the observer. Therefore, it is necessary to augment the state vector with a term aiming to estimate the disturbance. Once a disturbance estimate is available it will be possible to establish control action to suppress it,
- it is also possible to augment the system model with a term representing the estimation error, such that in a closed-loop scheme any estimation error can be integrated and driven to zero. We can add that tracking nonzero targets or desired trajectories that approach nonzero values may be done by augmenting the plant model dynamics with integrators and shifting the set point, such that again, the driving error is brought to zero.¹

Detailed analysis concerning augmented states to reject disturbances can be found in the work of Davison and Smith [7] and Bitmead et al. [4].

3-3 Observers

As we distinguished stochastic and deterministic models, we will also need observers capable of dealing with one or both types of systems. There may be different criteria to make a classification of observer, however, according to Findeisen [12] observers may be divided in three variants,

¹Appendix F gives more insight on this topic and includes two examples for linear systems.

- *deterministic*, do not assume presence of stochasticity in the system.
- *stochastic*, they do take stochasticity into account, their results are usually optimal in the sense that they are more probable under given output information.
- *optimization based*, they may or may not include stochastic information about the system and may include information about constraints on the variables constituting the system.

Figure 3-3 shows an observers' classification based on the definitions above.

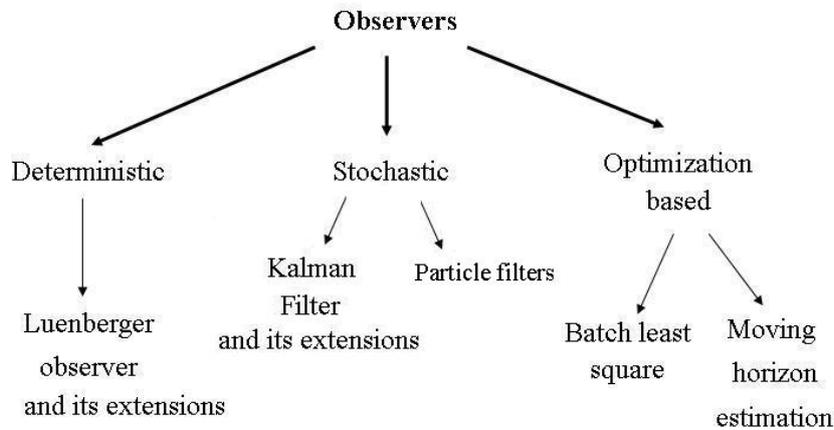


Figure 3-3: Classification of observers. Adapted from Findeisen [12].

The next chapter will introduce and discuss our case study, we can anticipate that our problem involves a stochastic nonlinear system, and as such, it will require observers that can handle those two characteristics. Thus we will focus on nonlinear variants of the Kalman filter, Luenberger observer and MHE.

Selection of the extensions of the Kalman filter are justified as follows:

- extended Kalman filter (EKF), it has been widely used with successful results in the area of process control, it is arguably computationally cheap and easy to implement, although tuning can become an issue, the number of variables in our application might allow for an easy tuning. It is claimed that this filter has poor performance when used on complex nonlinear systems and the situation is aggravated if discontinuities are present, however, our case study should not pose a big threat on this regard.
- unscented Kalman filter (UKF), it is a relatively new extension of the Kalman filter for nonlinear systems which does not include derivatives in its algorithm, it is said to be more accurate than the EKF while in the same order of computational burden. It also has the advantage that, when the analyzed system complies with certain structure, there exist some variants that decrease the computational demand even more. Besides, it has been implemented in very few cases in the process industries so it would be interesting to try it out.

- ensemble Kalman filter (EnKF), it presents an algorithm free from derivatives, it has proved to be very effective when dealing with highly nonlinear systems and to our knowledge, no implementation has been done in the area of process control, yet.
- extended Luenberger observer (ELO), it had been previously applied to the present case study, however, in the previous application no stochastic behavior had been taken into account so it was an opportunity to compare its efficacy compared to stochastic observers.
- moving horizon estimator (MHE), since it is based on an optimization algorithm it should be able to deal with uncertainties and it can also handle constraints.

We now present an overview of the observers used in our work, further information can be found in references cited in the text.

3-3-1 Extended Kalman Filter (EKF)

If we make the basic assumption that the state, observation and noise terms can be modeled as Gaussian random variables (GRVs), then the Bayesian recursion can be greatly simplified. In this case, only the conditional mean $x_k = E[x_k|Y_k]$ and covariance P_{x_k} need to be maintained in order to recursively calculate the posterior density $p(x_k|Y_k)$, which, under these Gaussian assumptions, is itself a Gaussian distribution. It can be shown that the recursive estimation is given by

$$x_{k+1|k+1} = (\text{prediction of } x_k) + K_k[y_{k+1} - (\text{prediction of } y_k)] \quad (3-14)$$

$$\begin{aligned} &= x_{k+1|k} + K_k(y_{k+1} - y_{k+1|k}) \\ P_{x_k} &= P_{x_k}^f - K_k P_{y_k} K_k^T. \end{aligned} \quad (3-15)$$

While this is a linear recursion, we have not assumed linearity of the model. The optimal terms in this recursion are given by

$$x_{k+1|k} = E[f(x_k|k, v_{k+1}, u_{k+1})] \quad (3-16)$$

$$y_{k+1|k} = E[h(x_k + 1|k, u_{k+1}, w_{k+1})] \quad (3-17)$$

$$K_k = E[(x_k - x_{k+1|k})(y_k - y_{k+1|k})^T] E[(y_k - y_{k+1|k})(y_k - y_{k+1|k})^T]^{-1} \quad (3-18)$$

$$= P_{x_k y_k} P_{y_k}^{-1} \quad (3-19)$$

where the optimal prediction (i.e. prior mean at time $k + 1$) of x_{k+1} is written as $x_{k+1|k}$, and corresponds to the expectation (taken over the posterior distribution of the state at time k) of a nonlinear function of the random variables x_k and v_k . A similar interpretation applies to the optimal prediction $y_{k+1|k}$, except the expectation is taken over the prior distribution of the state at time k . The optimal gain term K_k is expressed as a function of the expected cross correlation matrix (covariance matrix) of the state prediction error and the observation prediction error, and the expected auto-correlation matrix of the observation prediction error.

The Kalman filter calculates all terms in these equations exactly in the linear case, and can be viewed as an efficient method for analytically propagating a GRV through linear system

dynamics. As stated before, a certain criterion should be stated for optimality, it can be shown that for the case considered, *i.e.* a linear process with a linear measurement equation and Gaussian white process and measurement noises, the KF is also optimal in the sense that it minimizes the conditional covariance of the state, and thus the estimate is the conditional mean.

For more clarity, let us introduce a linear state space system given by

$$x_{k+1} = Ax_k + Bu_{k+1} + v_{k+1} \quad (3-20)$$

$$y_k = Hx_k + w_k \quad (3-21)$$

with

$$A \in \mathcal{R}^{n \times n} \quad (3-22)$$

$$B \in \mathcal{R}^{n \times q} \quad (3-23)$$

$$H \in \mathcal{R}^{q \times n} \quad (3-24)$$

$$(3-25)$$

and the rest of the variables as defined in (3-1)-(3-2) having the same covariance matrices (3-10) and (3-9). Although all the matrices may change in time, they are considered constants here. It is, we refer to a Linear Time Invariant system (LTI).

We can then define *a priori* (at step $k + 1$ given knowledge of the process up to step k) and *a posteriori* (at step $k + 1$ given the measurement y_{k+1}) estimate errors as

$$e_{k+1|k} = x_{k+1} - x_{k+1|k} \quad (3-26)$$

$$e_{k+1|k+1} = x_{k+1} - x_{k+1|k+1} \quad (3-27)$$

The *a priori* estimate error covariance is then

$$P_{k+1|k} = E[e_{k+1|k}e_{k+1|k}^T], \quad (3-28)$$

and the *a posteriori* estimate error covariance is

$$P_{k+1|k+1} = E[e_{k+1|k+1}e_{k+1|k+1}^T]. \quad (3-29)$$

In deriving the equations for the Kalman filter, we begin with the goal of finding an equation that computes an *a posteriori* state estimate $x_{k+1|k+1}$ as a linear combination of an *a priori* estimate $x_{k+1|k}$ and a weighted difference between an actual measurement y_{k+1} and a measurement prediction $Hx_{k+1|k}$ as shown below in equation (3-30):

$$x_{k+1|k+1} = x_{k+1|k} + K(y_{k+1} - Hx_{k+1|k}). \quad (3-30)$$

Some justification for equation (3-30) is given in Welch and Bishop [46]. The difference in (3-30) ($y_{k+1} - Hx_{k+1|k}$) is called the measurement innovation, or the residual. The residual reflects the discrepancy between the predicted measurement and the actual measurement. A residual of zero means that the two are in complete agreement. The matrix K in equation (3-30) is chosen to be the gain or blending factor that minimizes the *a posteriori* error covariance (3-29). This minimization can be accomplished by first substituting equation

(3-30) into the above definition for $e_{k+1|k+1}$, substituting that into equation (3-29), performing the indicated expectations, taking the derivative of the trace of the result with respect to K , setting that result equal to zero, and then solving for K . One form of the resulting K that minimizes equation (3-29) is given by

$$K_{k+1|k} = P_{k+1|k} H^T (H P_{k+1|k} H^T + R)^{-1}. \quad (3-31)$$

As can be seen, no approximations for the function elements were needed to calculate the optimal solution.

For nonlinear models, however, the Extended Kalman filter (EKF) must be used; this filter first linearizes the system equations through a Taylor-series expansion around the mean of the relevant Gaussian RV van der Merwe and Wan [45], i.e.

$$\begin{aligned} y &= h(x) = h(\bar{x} + \delta_x) \\ &= h(\bar{x}) + \nabla h \delta_x + \frac{1}{2} \nabla^2 h \delta_x^2 + \frac{1}{3} \nabla^3 h \delta_x^3 + \dots \end{aligned} \quad (3-32)$$

where the zero mean random variable δ_x has the same covariance, P_x , as x . The mean and covariance used in the EKF is thus obtained by taking the first-order truncated expected value of equation (3-32) for the mean, and its outer-product for the covariance, i.e. $\bar{y} \approx h(\bar{x})$ and $P_y^{LIN} = \nabla h P_x (\nabla h)^T$ respectively. Applying this result to equations (3-14) and (3-15), as in van der Merwe and Wan [45], we obtain

$$x_{k+1|k} \approx f(x_{k|k}, v_{k+1}, u_{k+1}) \quad (3-33)$$

$$y_{k+1|k} \approx h(x_{k|k}, u_{k+1}, w_{k+1}) \quad (3-34)$$

$$K_{k+1|k} \approx P_{x_{k+1|k} y_{k+1|k}}^{LIN} (P_{y_{k+1|k} y_{k+1|k}}^{LIN})^{-1}. \quad (3-35)$$

In other words, the EKF approximates the state distribution by a GRV, which is then propagated analytically through the first order linearization of the nonlinear system. Figure 3-4 graphically illustrates this characteristic. The explicit equations can be divided in two stages, the time update or forecast stage and the measurement or correction stage, these are the following:

- time update

$$x_{k+1|k} = f(x_{k|k}, u_k, v_k) \quad (3-36)$$

$$P_{k+1|k} = \mathcal{F}_k P_{k|k} \mathcal{F}_k^T + \mathcal{W}_k Q_k \mathcal{W}_k^T \quad (3-37)$$

- measurement update

$$K_k = P_{k+1|k} \mathcal{H}_k^T (\mathcal{H}_k P_{k+1|k} \mathcal{H}_k^T + \mathcal{V}_k R_k \mathcal{V}_k^T)^{-1} \quad (3-38)$$

$$x_{k+1|k+1} = x_{k+1|k} + K_k (y_k - h(x_{k+1|k}, u_k, w_k)) \quad (3-39)$$

$$P_{k+1|k+1} = (I - K_k \mathcal{H}_k) P_{k+1|k} \quad (3-40)$$

where

- \mathcal{F} is the Jacobian matrix of partial derivatives of f with respect to x , that is

$$\mathcal{F} = \frac{\partial f_i}{\partial x_j}, \quad (3-41)$$

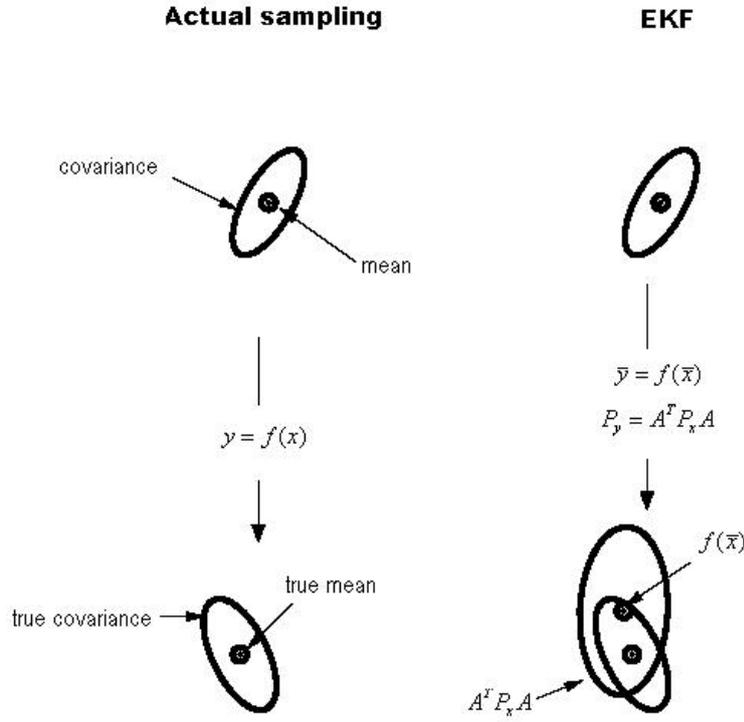


Figure 3-4: The EKF approximates the state distribution by a GRV, which is then propagated analytically through the first order linearization of the nonlinear system.

- \mathcal{W} is the Jacobian matrix of partial derivatives of f with respect to w ,

$$\mathcal{W} = \frac{\partial f_i}{\partial w_j}, \quad (3-42)$$

- \mathcal{H} is the Jacobian matrix of partial derivatives of h with respect to x ,

$$\mathcal{H} = \frac{\partial h_i}{\partial x_j}, \quad (3-43)$$

- \mathcal{V} is the Jacobian matrix of partial derivatives of h with respect to v ,

$$\mathcal{V} = \frac{\partial h_i}{\partial v_j}. \quad (3-44)$$

As such, the EKF can be viewed as providing first-order approximations to the optimal terms. Furthermore, the EKF does not take into account the “uncertainty” in the underlying random variable when the relevant system equations are linearized. This is due to the nature of the first-order Taylor series linearization that expands the nonlinear equations around a single point only, disregarding the spread (uncertainty) of the prior RV.

All KF variants for nonlinear systems calculate an estimate $x_{k+1|k+1}$ and covariance matrix P for a probability density function (p.d.f.) which is non-Gaussian, since, although the prior

distribution might be Gaussian, after undergoing a nonlinear transformation the Gaussianity is lost. The performance of these KFs depends on how representative the Gaussian p.d.f. with mean $x_{k+1|k+1}$ and covariance P is for the (unknown) p.d.f. $p(x_k|\hat{\mathbf{Y}})$.

Figure 3-5 shows a non-Gaussian p.d.f. and three possible Gaussian approximations.

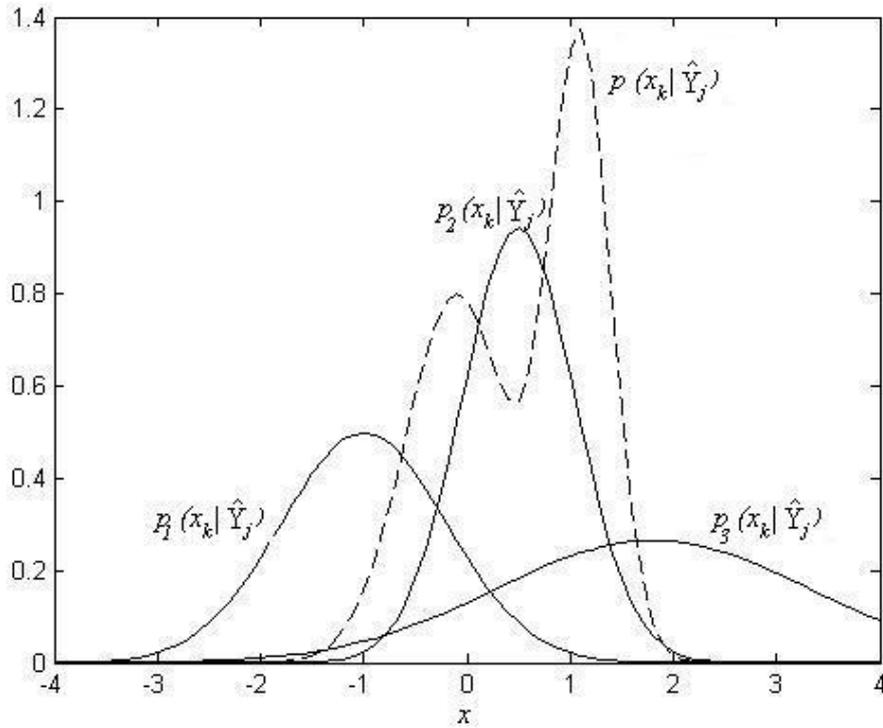


Figure 3-5: Approximation of a non-Gaussian probability density function via a Gaussian probability density function

Intuitive thoughts about which is the most representative Gaussian distribution for any given p.d.f. are formulated in two criteria: the *consistency*, and the *information content* of the estimate. The information content of the state estimate defines an ordering between all consistent filters Lefebvre et al. [21].

- A state x_k with covariance matrix P_k is called consistent if

$$E_{p(x_k|\mathbf{Y})}[(x_k - x_{k+1|k})(x_k - x_{k+1|k})^T] \leq P_k. \quad (3-45)$$

This means that for consistent results, matrix P_k is equal or larger than the expected squared deviation with respect to the estimate x_k under the (unknown) distribution $p(x_k|\hat{\mathbf{Y}})$. Inconsistency is the most encountered problem with the KF variants. In the case presented in Figure 3-6 the covariance matrix is too small and does no longer represent a reliable measure for the uncertainty on the state. Even more, once an inconsistent estimate is met, the subsequent state estimates are also inconsistent Lefebvre et al. [21]. This is because the filter believes the inconsistent state estimate to be more accurate

than it is in reality and hence, it attaches too much weight to this state estimate when processing new measurements. The consistency of the state estimate is a necessary condition for a filter to be acceptable. It has been noted that if inconsistent estimations are calculated, that exacerbates divergence problems.

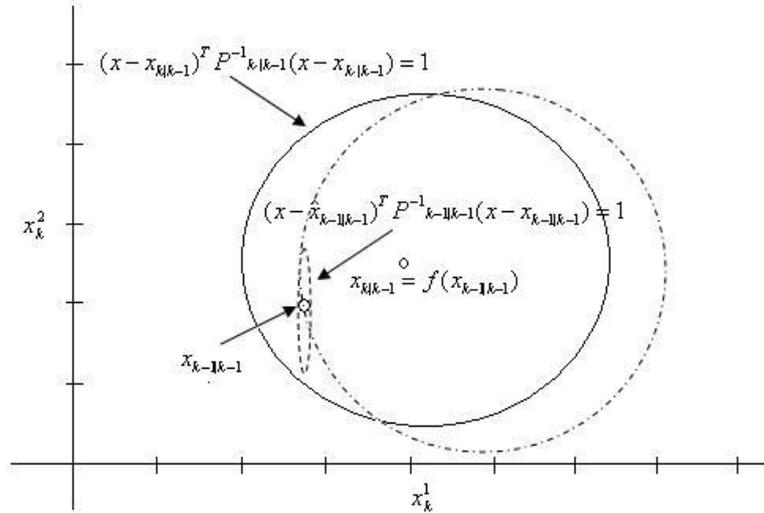


Figure 3-6: Graphical interpretation of the concept of consistency.

- The information content of the state estimate is based on the calculated covariance matrix $P_{k|k}$ since this indicates how uncertain the state is. A large covariance matrix indicates an inaccurate (and little useful) state estimate; the smaller the covariance matrix, the larger the information content of the state estimate.

There is a trade-off between consistent and informative state estimates: inconsistency can be avoided by making P_k artificially larger. Making P_k too large, however (larger than necessary for consistency) corresponds to losing information about the real accuracy of the state estimate.

3-3-2 Unscented Kalman Filter (UKF)

The UKF is part of a set of filters known as linear regression Kalman filters (LRKF). LRKF linearizes the process and measurement functions by statistical linear regression of the functions through a number of regression points.

The LRKF uses the values generated by evaluating the system model in r regression points $x_{k-1|k-1}^j$, $j = 1, \dots, r$, in state space to model the behavior of the process function in the uncertainty region around the updated state estimate $x_{k-1|k-1}$. The regression points are chosen such that their mean and covariance matrix equal the state estimate $x_{k-1|k-1}$ and its

covariance matrix $P_{k-1|k-1}$. Further details of how these points are selected for each variant are given in the following sections.

The main idea behind the UKF according to Julier and Uhlman [18] is that it is easier to approximate a probability distribution than a nonlinear function. Therefore this filter first propagates a number of “points” through the nonlinear system model in order to capture the statistics of the points that went under such transformation. Oppositely, the EKF propagates a point in state space through a linearized version of the original system model. Figure 3-7 illustrates this idea. The precursors of the UKF, defined the filtering algorithm based on the

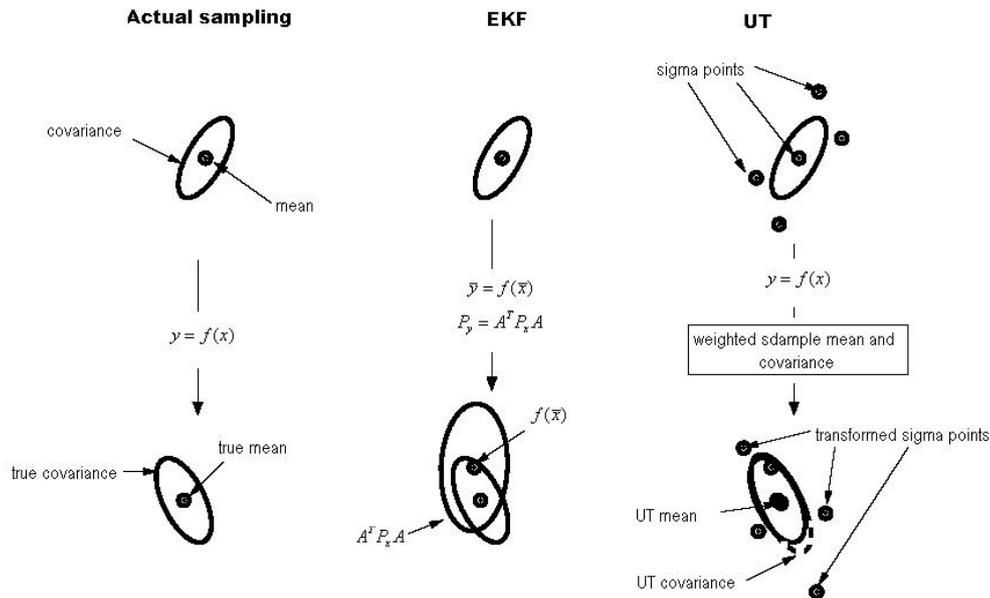


Figure 3-7: Comparison between UKF and EKF of how each filter generates variable approximations.

same steps that the EKF works on:

- Predict the new state of the system $x_{k+1|k}$ and its associated covariance $P_{xx_{k+1|k}}$. This prediction must take account of the effects of process noise.
- Predict the expected observation $y_{k+1|k}$ and the innovation covariance $P_{vv_{k+1|k}}$. This prediction should include the effects of observation noise.
- Finally, predict the cross-correlation matrix $P_{xz_{k+1|k}}$.

A summary of the basic method is as follows Julier and Uhlman [18]:

- Compute the set σ of $2n$ points (where n is the dimension of the state vector) from the rows or columns of the matrices $\pm\sqrt{nP_{xx}}$. This set is zero mean with covariance P . Compute a set of points with the same covariance, but with mean \bar{x} which is the mean of the original distribution, by translating each of the points as $\chi = \sigma + \bar{x}$.

- Transform each point as $\mathcal{Y}_i = h(\chi_i)$.
- Compute \bar{y} and P_{yy} by computing the mean and covariance of the $2n$ points in the set \mathcal{Y}_i . In order to account for the process noise, an augmented state vector with dimension $n + q$ must be employed. It is defined as $x_{k|k_A} = \begin{pmatrix} x_k \\ v_k \end{pmatrix}$, where q is the dimension of the process noise vector. Also an augmented covariance matrix P_A is defined. Both the augmented vector and the augmented covariance matrix are denoted by the subscript A .

Taken this into account, the method can be implemented via the following steps Julier and Uhlman [18]:

- The set of sigma points is computed from the $(n + q) \times (n + q)$ matrix $P_{A_{k|k}}$ as

$$\sigma_{A_{k|k}} \leftarrow 2(n + q) \text{ rows or columns from } \sqrt{(n + q + \kappa)P_{k|k_A}} \quad (3-46)$$

$$\chi_{k|k}^0 = \bar{x}_{k|k_A} \quad (3-47)$$

$$\chi_{k|k}^i = \sigma_{k|k_A}^i + \bar{x}_{k|k_A} \quad (3-48)$$

which assures that $P_{A_{k|k}} = \frac{1}{2(n+q+\kappa)} \sum_{i=1}^{2(n+q)} [\chi_{k|k}^i - \bar{x}_{k|k}] [\chi_{k|k}^i - \bar{x}_{k|k}]^T$.

- The transformed set of sigma points are evaluated for each of the 0 to $2(n + q)$ points by

$$\chi_{k+1|k}^i = f[\chi_{k|k}^i, u_{k+1}, k]. \quad (3-49)$$

- The predicted mean is computed as

$$\bar{x}_{k+1|k} = \frac{1}{n + q + \kappa} \left\{ \kappa \chi_{k+1|k}^0 + \frac{1}{2} \sum_{i=0}^{2(n+q)} \chi_{k+1|k}^i \right\}. \quad (3-50)$$

- And the predicted covariance is computed as

$$P_{k+1|k} = \frac{1}{(n+q+\kappa)} \left\{ \kappa [\chi_{k+1|k}^0 - \bar{x}_{k+1|k}] [\chi_{k+1|k}^0 - \bar{x}_{k+1|k}]^T + \frac{1}{2} \sum_{i=1}^{2(n+q)} [\chi_{k+1|k}^i - \bar{x}_{k+1|k}] [\chi_{k+1|k}^i - \bar{x}_{k+1|k}]^T \right\}. \quad (3-51)$$

Where κ provides an extra degree of freedom to fine tune the higher order moments of the approximation, and can be used to reduce the overall prediction errors. When x_k is assumed Gaussian, a useful heuristic is to select $n + \kappa = 3$. If a different distribution is assumed for x_k then a different choice of κ might be more appropriate.

Using these equations assures that the prediction with uncertainty in the state and process noise yields estimation errors in third order and above.

To complete the description of the UKF filter, the equivalent statistics for the innovation sequence and the cross correlation must be determined. By evaluating each sigma point

through the observation model to yield $\mathcal{Y}_{k+1|k}^i = h[\chi_{k+1|k}, u_{k+1}, k+1]$, the mean observation is found from

$$\bar{y}_{k+1|k} = \frac{1}{n+q+\kappa} \left\{ \kappa \mathcal{Y}_{k+1|k}^0 + \frac{1}{2} \sum_{i=0}^{2(n+q)} \mathcal{Y}_{k+1|k}^i \right\} \quad (3-52)$$

and the covariance is determined from

$$\begin{aligned} P_{yy_{k+1|k}} &= \frac{1}{(n+q+\kappa)} \left\{ \kappa [\mathcal{Y}_{k+1|k}^0 - \bar{y}_{k+1|k}] [\mathcal{Y}_{k+1|k}^0 - \bar{y}_{k+1|k}]^T \right. \\ &\quad \left. + \frac{1}{2} \sum_{i=1}^{2(n+q)} [\mathcal{Y}_{k+1|k}^i - \bar{y}_{k+1|k}] [\mathcal{Y}_{k+1|k}^i - \bar{y}_{k+1|k}]^T \right\} \end{aligned} \quad (3-53)$$

The innovation covariance is equal to the sum of $P_{yy_{k+1|k}}$ and the observation noise covariance matrix:

$$P_{vv_{k+1|k}} = P_{yy_{k+1|k}} + R_{k+1}. \quad (3-54)$$

Finally, the cross correlation matrix is

$$\begin{aligned} P_{xy_{k+1|k}} &= \frac{1}{(n+q+\kappa)} \left\{ \kappa [\chi_{k+1|k}^0 - \bar{x}_{k+1|k}] [\mathcal{Y}_{k+1|k}^0 - \bar{y}_{k+1|k}]^T \right. \\ &\quad \left. + \frac{1}{2} \sum_{i=1}^{2(n+q)} [\chi_{k+1|k}^i - \bar{x}_{k+1|k}] [\mathcal{Y}_{k+1|k}^i - \bar{y}_{k+1|k}]^T \right\}. \end{aligned} \quad (3-55)$$

If the system model presents certain characteristics such as a linear output equation or additive-type noise, that is

$$x_{k+1} = f(x_k) + v_k, \quad (3-56)$$

$$y_{k+1} = g(x_{k+1}) + w_{k+1}, \quad (3-57)$$

or

$$x_{k+1} = f(x_k, v_k), \quad (3-58)$$

$$y_{k+1} = G_{k+1}x_{k+1} + w_{k+1}, \quad (3-59)$$

or

$$x_{k+1} = f(x_k) + v_k, \quad (3-60)$$

$$y_{k+1} = G_{k+1}x_{k+1} + w_{k+1}, \quad (3-61)$$

where G is a matrix of adequate dimensions, it is possible to implement a more computationally efficient unscented type of filter as demonstrated by Hao et al. [15]. The variants of UKF presented there are summarized in Table 3-1.

The interpretation of the columns from Table 3-1 is as follows:

- Dimension - is the size of the augmented state vector of the system, it is, the dimension of the state vector solely plus the dimension of the process noise vector.
- Number - is the number of sigma points used to carry out the algorithm.
- Update - is the number of times the sigma points need to be updated at every iteration.

Table 3-1: Variants of the UKF, Additive UKF, Rao-Blackwellised UKF and Additive Rao-Blackwellised UKF

Filter	Sigma points		
	Dimension	Number	Update
UKF	$2n + q$	$4n + 2q + 1$	1
AUKF	n	$2n + 1$	2
RBUKF	$2n$	$4n + 1$	1
RBAUKF	n	$2n + 1$	1

EKF and UKF with Q-varying matrix

For both filters EKF and UKF a variant was also analyzed during the development of this thesis. The idea behind using a time-varying Q matrix rests on the work by Valappil and Georgakis [43]. “Most of the methods just mentioned assume constant noise characteristics and the availability of data required to obtain a true representation of noise statistics. The assumption of time-invariant process noise is more appropriate for a continuous process that operates at a steady state. But for continuous or batch processes with time-varying process dynamics and operating at a range of process conditions, these noise statistics are time-varying. Using a fixed value of noise statistics can lead to poor filter performance and even result in filter divergence, ... Even in cases where the fixed noise statistics are acceptable, finding the appropriate values can be a tedious task. One is left with the task of selecting a number of parameters, at least equal to the number of states for the case where the matrix is assumed to be diagonal. These realities prompt one to consider alternate schemes to improve the estimate of the process-noise statistics”

The variation of this Q matrix represents the uncertainty of the process model; the methodology presented involves linear approximation of the dependence of the model predictions on the model parameters. In the end, the equation relating the Q matrix with the variation in the parameter is given by:

$$Q = J_p \Gamma_p J_p^T \quad (3-62)$$

where J_p is the partial derivative of the process equation with respect to the parameters, evaluated at the instant the Q matrix is used, and Γ_p is the covariance matrix of the process parameters.

Success in achieving a good model for a batch process requires the knowledge of the model parameters and their variability along the batch, this can be attributed to the transient behavior inherent to batch processes. Uncertainty in the system could then be attributed to the system’s parameters variability along the batch. This approach is followed here, and justifies the use of the method described in the previous paragraphs to calculate the value of the Q matrix.

3-3-3 Ensemble Kalman Filter (EnKF)

The EnKF was proposed by Evensen [9] and describes how a Monte Carlo method is implemented in data assimilation problems; a few years later Burgers et al. [5] published an article

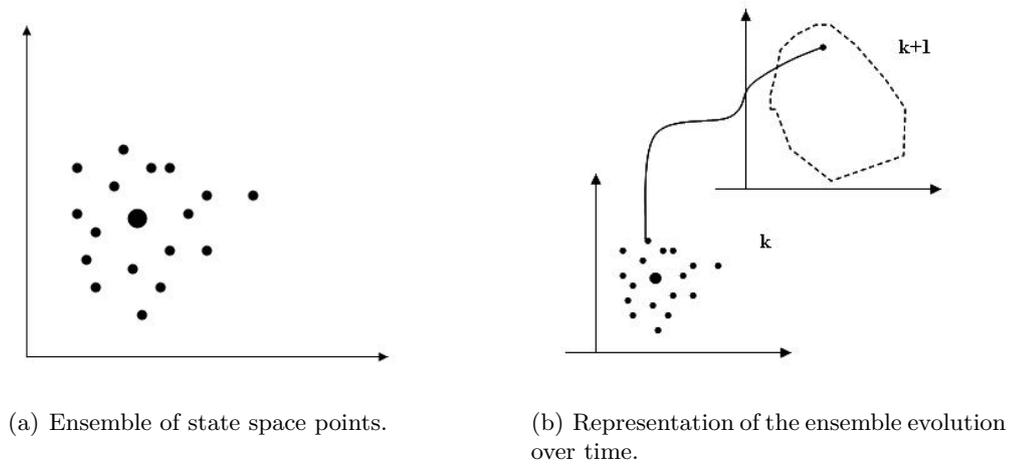


Figure 3-8: A number of points in state space called the ensemble are shown here, this ensemble aims to capture all the statistical properties of the system variables. Integrating and updating each ensemble member allows to propagate the model variables' statistics.

clarifying the mechanism of the filter. Data assimilation is a technique that combines two sources of information: the model and the available measurements. It assumes that both the model and the measurements are subject to errors. So in fact, data assimilation can be regarded as a term used in areas such as hydrology, meteorology, etc. for the filtering problem already discussed in chapter 1.

Evensen [9] states: “In the Monte Carlo method, one first calculates a best guess initial condition based on information available from data and statistics. The model solution based on this initial state is denoted the central forecast. The uncertainty in this best guess initial condition is represented by the initial variance. An ensemble of initial states is then generated in which the mean equals the best guess initial condition and the variance is specified on the basis of knowledge of the uncertainty in the first-guess initial state”. A graphical representation of this is shown in Figure 3-8(a).

Evensen [10] also mentions that “ A large cloud of model states (points in state space) can be used to represent a specific probability density function. By integrating such an ensemble of states forward in time, it is easy to calculate approximate estimates for moments of the probability density function at different time levels. In this context the Monte Carlo method might be considered a particle method in the state space”. This can be visualized in Figure 3-8(b).

The following subsection is completely taken from Evensen [10]. There the error statistics is represented using an ensemble of model states. Then an alternative to the traditional error covariance equation is proposed for the prediction of the error statistics and finally an overall analysis is presented.

Representation of error statistics

The error covariance matrices for the forecast and the analyzed estimate, $P_{k+1|k}$ and $P_{k+1|k+1}$ are defined in the Kalman filter in terms of the true state as

$$P_{k+1|k} = \overline{(x_{k+1|k} - x_k^t)(x_{k+1|k} - x_k^t)^T}, \quad (3-63)$$

$$P_{k+1|k+1} = \overline{(x_{k+1|k+1} - x_k^t)(x_{k+1|k+1} - x_k^t)^T}, \quad (3-64)$$

where the overline denotes an expectation value, x is the model state vector at a particular time and the subscripts $_{k+1|k}$, $_{k+1|k+1}$, represent forecast, and analyzed state, respectively. However the true state is not known, so the ensemble covariance matrices are defined around the ensemble mean \bar{x} ,

$$P_{k+1|k} \approx P_{e_{k+1|k}} = \overline{(x_{k+1|k} - \bar{x}_{k+1|k})(x_{k+1|k} - \bar{x}_{k+1|k})^T}, \quad (3-65)$$

$$P_{k+1|k+1} \approx P_{e_{k+1|k+1}} = \overline{(x_{k+1|k+1} - \bar{x}_{k+1|k+1})(x_{k+1|k+1} - \bar{x}_{k+1|k+1})^T}, \quad (3-66)$$

$\bar{x}_{k+1|k}$ and $\bar{x}_{k+1|k+1}$ are averaged over the ensemble members. Thus it is possible to use an interpretation where the ensemble mean is the best estimate and the spreading of the ensemble around the mean is a natural definition of the error of the ensemble mean.

Since the error covariances as defined in equations (3-65) and (3-66) are expressed in terms of ensemble averages, there will clearly exist infinitely many ensembles with an error covariance equal to $P_{e_{k+1|k}}$ and $P_{e_{k+1|k+1}}$. Thus instead of storing a full covariance matrix, we can present the same error statistics using an appropriate ensemble of model states. However, when the size in the ensemble N increases, the errors in the Monte Carlo Sampling will decrease proportionally to $\frac{1}{\sqrt{N}}$.

An overall analysis scheme

As pinpointed by Burgers et al. [5]; it is essential that the observations are treated as random variables having a distribution with mean equal to the first-guess observations and covariance equal to R as defined in equation (3-10).

It can be shown, see Evensen [9], that the relation between the analyzed and forecast ensemble mean is identical to the relation between the analyzed and forecast state in the standard Kalman filter. Furthermore, by updating each of the ensemble members using the perturbed observations, one also create an ensemble having the correct error statistics for the analysis. The updated ensemble can then be integrated forward in time till the next observation time. Moreover, the error covariance, P of the analyzed ensemble is reduced in the same way as in the standard Kalman filter.

The resulting algorithm can be found in the article from Gillijns et al. [14], and is as follows:

Forecast step

$$x_{k+1|k}^i = f(x_{k|k}^i, u_k) + w_k^i, \quad (3-67)$$

$$\bar{x}_{k+1|k} = \frac{1}{N} \sum_{i=1}^N x_{k+1|k}^i, \quad (3-68)$$

$$E_{k+1|k} = [x_{k+1|k}^1 - \bar{x}_{k+1|k} \cdots x_{k+1|k}^N - \bar{x}_{k+1|k}], \quad (3-69)$$

$$E_{y_{k+1|k}} = [y_{k+1|k}^1 - \bar{y}_{k+1|k} \cdots y_{k+1|k}^N - \bar{y}_{k+1|k}], \quad (3-70)$$

$$\hat{P}_{xy_{k+1|k}} = \frac{1}{N-1} E_{k+1|k} (E_{y_{k+1|k}})^T, \quad (3-71)$$

$$\hat{P}_{yy_{k+1|k}} = \frac{1}{N-1} E_{y_{k+1|k}} (E_{y_{k+1|k}})^T. \quad (3-72)$$

Analysis step

$$\hat{K}_k = \hat{P}_{xy_{k+1|k}} (\hat{P}_{yy_{k+1|k}})^{-1}, \quad (3-73)$$

$$x_{k+1|k+1}^i = x_{k+1|k}^i + \hat{K}_k (y_{k+1} + v_{k+1}^i - h(x_{k+1|k}^i)), \quad (3-74)$$

$$\bar{x}_{k+1|k+1} = \frac{1}{N} \sum_{i=1}^N x_{k+1|k+1}^i. \quad (3-75)$$

3-3-4 Moving Horizon Estimator (MHE)

An optimization based estimator estimates the states of a nonlinear system, described by:

$$x_{k+1} = f(x_k, u_k) + w_k \quad (3-76)$$

$$y_k = g(x_k) + v_k, \quad (3-77)$$

$$x_0 = x_{01} \quad (3-78)$$

the vectors w_k and v_k are, like in the EKF, noise which acts on the system. The objective of the estimator is to find the largest probability that the state of the system (3-76)-(3-77) has estimates $\{x_0, x_1, \dots, x_T\}$, given the measurements $\{y_0, y_1, \dots, y_{T-1}\}$ at time instant $T-1$. Instead of estimating the states only at time instants T or $T-1$, like recursive methods, this method estimates all the states from x_0 until x_T . The other estimates $\{x_0, x_1, \dots, x_{T-1}\}$ are at every instant corrected by new measurements. This correction of past estimates by new measurements is called smoothing. Only the last estimate, x_T will be sent to the controller. Because the process measurements y_k are correlated with the state x_k , their conditional density function can be written as:

$$p(x_0, x_1, \dots, x_T | y_0, y_1, \dots, y_{T-1}) \quad (3-79)$$

The conditional density function describes the probability that the state has estimates (x_0, x_1, \dots, x_T) , if the measurements $(y_0, y_1, \dots, y_{T-1})$ are known. The goal is to find the estimate

that maximizes this density function; that is, the optimal state. This optimal estimate is written as $\hat{x}_{0|T-1}, \hat{x}_{1|T-1}, \dots, \hat{x}_{T|T-1}$ and is the solution of:

$$\{\hat{x}_{0|T-1}, \hat{x}_{1|T-1}, \dots, \hat{x}_{T|T-1}\} \in \underset{\{x_0, x_1, \dots, x_T\}}{\operatorname{argmax}} p(x_0, x_1, \dots, x_T | y_0, y_1, \dots, y_{T-1}) \quad (3-80)$$

the optimal estimate is the peak of the conditional density function (CDF). The maximization of the CDF can be written as an optimization problem. In order to do this one assumes that w_k and v_k are independent, normally distributed variables with zero mean and covariances Q and R ; x_0 is normal with mean \hat{x}_0 and covariance Π_0 . In fact this are the same assumptions made for the EKF. The transformation of the CDF to an optimization problem can be found in Rao and Rawlings [34]. The state estimation problem (3-80) can be formulated as:

$$\min_{x_0, \{w_k\}_{k=0}^{T-1}} \|x_0 - \hat{x}_0\|_{\Pi_0^{-1}}^2 + \sum_{k=0}^{T-1} \|v_k\|_{R^{-1}}^2 + \sum_{k=0}^{T-1} \|w_k\|_{Q^{-1}}^2 \quad (3-81)$$

subject to:

$$x_{k+1} = f(x_k, u_k) + w_k \quad (3-82)$$

$$y_k = g(x_k) + v_k, \quad (3-83)$$

$$(3-84)$$

$$x_k \in \mathcal{X} \quad w_k \in \mathcal{W} \quad (3-85)$$

the problem computes the sequence $\{x_0, \{w_k\}_{k=0}^{T-1}\}$ such that the cost function is minimized in (3-81). When the variables $\{w_k\}_{k=0}^{T-1}$ and x_0 are known, the optimal estimates, denoted by $\hat{x}_{0|T-1}, \hat{x}_{1|T-1}, \dots, \hat{x}_{T|T-1}$ can be computed via the model equation (3-82) and the measured inputs. The estimation problem described above is often referred to as the *batch least squares* estimation problem.

If one interprets v_k and w_k as normally distributed noise sequences then Q and R represent the covariances of the process and measurement noise.

Solving the latter problem on-line is impossible because the size of the problem grows without bound when more measurements come available, therefore, the estimation problem will be formulated over a finite horizon and is called the moving horizon approach, or moving horizon estimation (MHE).

To bound the size of the optimization, the number of measurements on which the estimation is based must be limited. Suppose that the number of measurements on which the estimation is based is equal to N . The estimation can then be divided into two parts, $t_1 = \{0 \leq k \leq T - N - 1\}$ and $t_2 = \{T - N \leq k \leq T - 1\}$. the first part t_1 , summarizes the effect of the data until $T - N - 1$. The second part is estimated via the optimization from $T - N$ until $T - 1$. The data window (or horizon size) in which the optimization takes place moves with every new measurement. We illustrate this in figure 3-9.

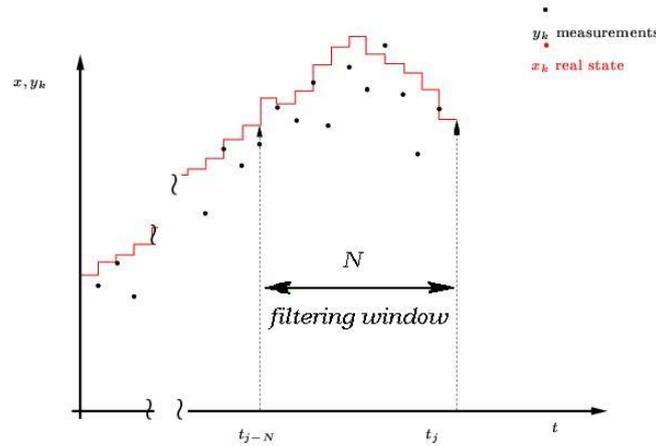


Figure 3-9: moving horizon

For MHE the horizon size is N (the width between both broken lines). The BLS estimation problem encompasses from time instant 0 till the second broken line, as mentioned before, the BLS problem is a larger one.

The MHE problem can be written as:

$$\min_{x_{T-N}, \{w_k\}_{k=T-N}^{T-1}} \|x_{T-N} - \hat{x}_{T-N}\|_{\Pi_{T-N}^{-1}}^2 + \sum_{k=T-N}^{T-1} \|v_k\|_{R^{-1}}^2 + \sum_{k=T-N}^{T-1} \|w_k\|_{Q^{-1}}^2 \quad (3-86)$$

subject to (3-82)-(3-85).

The initial value \hat{x}_{T-N} with covariance Π_{T-N} summarizes the prior information at time $T-N$, and is called the arrival cost. Note that when $T \leq N$ the MHE problem is the same as the BLS problem, and the initial values \hat{x}_0 , Π_0 are used. The calculation of the arrival cost can be done in different ways. If the underlying system is linear, one could use the Kalman filter. All the necessary variables, like covariances Q and R , measurements u_{T-N-1} and y_{T-N-1} , etc. are available. The computation of the new initial values (\hat{x}_{T-N} and Π_{T-N}) is equal to the KF scheme. If one uses the KF to update the arrival cost, it is shown by Roberson et al. [39], that the MHE reduces to a Kalman filter, regardless of horizon size N . As said before the process must be linear and no constraints must be added. The similarity can be understood by looking at the objectives of both. In the KF a estimate with minimum error variance is reconstructed. The objective of MHE is to estimate the mode of the conditional density function. Because in linear systems the conditional density is symmetric and has only one mode, the optimal estimates in the KF and MHE are equivalent. For linear systems, the only advantages of MHE over KF is the ability to incorporate constraints.

For nonlinear systems do not exist algebraic expressions for the arrival cost. The EKF can than be used to approximate the arrival cost. MHE is equivalent to the EKF if a horizon length N of 1 is used Roberson et al. [39]. If $N > 1$ then the EKF is used only for the calculation of the arrival cost, while the rest of the estimates are computed via the nonlinear optimization.

The online operation of the MHE can be divided in two steps:

- calculate the initial estimate and the estimated disturbances via the optimization problem 3-86, after which the estimated states can be calculated,
- estimate the new arrival costs. The arrival costs are used as initial conditions for the optimization problem at the next time instant.

A problem of using an EKF to compute the arrival costs is that it may weight the past too heavily. This can be critical if the initial guess \hat{x}_0 is poor and the horizon is too short. In literature there exist two ways to manipulate the arrival costs if the EKF updates limit performance, this can be the use of a forgetting factor or make use of smoothed estimates.

Addition of a forgetting factor

A simple strategy is to premultiply the arrival costs, estimated via the EKF, by a scalar $\alpha \in (0, 1)$. This α is called a forgetting factor. The corresponding expression for the arrival costs is:

$$\alpha \|x_{T-N} - \hat{x}_{T-N}\|_{\Pi_{T-N}^{-1}}^2 \quad (3-87)$$

by doing this, the part of the estimation problem based on the real measurements is considered more important than the part of the “guessed” arrival costs, by assigning an appropriate value for α .

Making use of smoothed estimates

In the EKF update one uses $\hat{x}_{T-N|T-N-1}$ and $\Pi_{T-N|T-N-1}$ to calculate the arrival costs at time instant $T-1$. However, at time instant $T-1$, one has already knowledge of measurement at time $T-1$. A more accurate estimate of the initial value is given by the smoothed estimate $\hat{x}_{T-N|T-1}$, calculated during the optimization at the previous time instant. An advantage of this smoothed update is that no information gained by the last smoothed estimations is lost. There does not exist a probabilistic motivation for using the smoothed estimates, because it uses future estimates to estimate the arrival costs (which summarizes effects that happened in the past). However, the smoothed update scheme provides the most reliable estimation. It considers the smoothed estimates based on measurements and better estimates for the arrival cost than the possibly guessed initial values, which are used in the EKF update.

3-3-5 Extended Luenberger Observer (ELO)

The Luenberger observer is a deterministic estimator, and thus assumes that no noise is acting on the system. In 1964 David Luenberger [22] described a system that estimates which are not available for direct measurement. The Luenberger observer estimates the states of an linear time invariant (LTI) system in state-space.

Suppose one has an LTI state-space representation of any system,

$$x_{k+1} = Ax_k + Bu_k, \quad (3-88)$$

$$y_k = Cx_k, \quad (3-89)$$

$$x_0 = x_{01} \quad (3-90)$$

where x_k is the state vector, u_k the input vector and y_k the output vector.

The goal of the Luenberger observer (LO) is to produce an estimate, \hat{x}_k , of the state vector x_k , such that the estimation error

$$e_k = x_k - \hat{x}_k \quad (3-91)$$

is minimized.

In the LO the dynamics of the estimator are described by:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + K(y_k - C\hat{x}_k), \quad \hat{x}_0 = \hat{x}_{01} \quad (3-92)$$

The observer calculates estimates for the next time instant, by correcting the estimate \hat{x}_k by the difference between the measured and estimated output multiplied by a gain. This gain, K , is called the observer gain. It is easily verified that $\hat{x}_0 = x_0$, implies $\hat{x}_{k+1} = x_{k+1}$. This approach, correction of an estimated state by the difference of the measured and estimated output, is used in most linear techniques.

The goal of the LO is to minimize the estimation error. The dynamics of the estimation error are given by

$$e_{k+1} = x_{k+1} - \hat{x}_{k+1} = (A - KC)e_k. \quad (3-93)$$

If all eigenvalues of the matrix $A - KC$ lie inside the unit circle, the estimation error will converge to zero. The eigenvalues of $A - KC$ can be placed arbitrarily by choosing K accordingly if the system under consideration is observable. The observer gain is often selected such that the eigenvalues of $A - KC$ are somewhat faster than those of the observed system. In this way, the convergence of the estimates is faster than other effects.

The version of this type of observer for nonlinear systems receives the name, extended Luenberger observer (ELO). The derivation of the ELO that follows is based on the work of Dochain [8].

A general nonlinear model may be described by:

$$x_{k+1} = f(x_k, u_k) \quad (3-94)$$

$$y_k = h(x_k) \quad (3-95)$$

$$x_0 = x_{01} \quad (3-96)$$

In a similar way as in the linear case, the dynamics of the observer can be written as,

$$\hat{x}_{k+1} = f(\hat{x}_k, u_k) + K_k(y_k - h(\hat{x}_k)), \quad (3-97)$$

if the estimation error is given by

$$e_{k+1} = x_{k+1} - \hat{x}_{k+1} = f(\hat{x}_k + e_k, u_k) - f(\hat{x}_k, u_k) - K_k(h(\hat{x}_k + e_k) - h(\hat{x}_k)). \quad (3-98)$$

The error dynamics do not immediately allow a condition on which the error converges to zero. If one assumes that the initial error, e_0 , is small, a linearization around $e = 0$ can be made. Then (3-98) becomes,

$$e_{k+1} = (A_k - K_k C_k) e_k \quad (3-99)$$

where

$$A_k = \left[\frac{\partial f(x_k, u_k)}{\partial x} \right]_{x_k = \hat{x}_k, u_k = u_k} \quad \text{and} \quad C_k = \left[\frac{\partial h(x_k)}{\partial x} \right]_{x_k = \hat{x}_k}. \quad (3-100)$$

A_k and C_k are the linearization of the system and output equation (3-94), (3-95) respectively, around the estimate \hat{x}_k and are time varying matrices, which means that the observer gain K_k is time varying.

Once again, the design of the state observer consists of choosing an appropriate gain. The objective is to choose K_k such that the linearized error dynamics is asymptotically stable, which implies that the eigenvalues of the matrix $A_k - K_k C_k$ should lie inside the unit circle. Nevertheless, the eigenvalues of the linearized system are dependent on the working point, and it is not possible to give a general approach to calculate the desired gain.

3-4 Observer Design

Design of nonlinear versions of KF (EKF, UKF and EnKF)

An observer uses both a copy of the model of the system of interest, and measurements taken from the real system. These two information sources together with some extra knowledge such as the statistics of the variables involved in the system, and about the system itself, are exploited by the observer in order to deliver accurate state estimates. Making appropriate use of all that knowledge is achieved by proper tuning.

The state estimates given by the observer are generated by ponderating the estimates given by the internal model as well as the measurements obtained from the system, a successful estimation is achieved if an adequate trade-off is found between the model predictions and the measurements available. Why?, the reasoning is as follows:

- the observer is, in the best scenario, fed by the same input that the system (there are no disturbances whatsoever), this allows the internal model of the observer to generate state estimates whose accuracy depends on how well the model reproduces reality. On the other hand the observer also receives information in the form of measurements from the system, this information may be corrupted due to any of the error source already mentioned. Therefore, in general, there will be a discrepancy between estimates given by the observer's internal model and the measurements. Both information sources bear erroneous information and thus are not 100 % reliable. Therefore, both sources must be combined such that the output is as close to an uncorrupted value as possible.

In the case of the nonlinear versions of the KF the tuning knobs are constituted by the covariance matrices belonging to the system and measurement noise Q and R , respectively, and the initial error covariance matrix, P ; all as defined as in section 3-1.

The meaning of the components in the Q and R matrices can be translated in practical terms as how much we rely on the information given by the model of the plant embedded in the observer, and the information given by the measurements performed on the system. For instance, large values in the elements of Q indicate a large uncertainty in the veracity of the variable values delivered by the observer, the opposite applies, small values indicate certainty. This explanation is applicable to the elements of R , nevertheless, the elements of this matrix indicate the veracity of the measurements.

The P matrix assesses how certain we are about the value of the errors, and the smaller it is, the better the results.

It must be noted that “*the ratio between the elements of the matrices indicates how truthful the information given by the observer’s model is with respect to the measurements*”. Appendices B and D show two analyzed cases where larger weights are assigned to either the internal model estimates or the system measurements in order to achieve reasonable state estimates estimates.

For simplicity, we may use equations (3-14) and (3-31) in the case x is a scalar to clarify the latter paragraph. If we make $Q \gg P$ we see that influence of the model-predicted state estimates on the observer estimates is negligible. Contrary, if $P \gg Q$ then the influence of the measurements is negligible and the observer estimates highly rely on the estimates given by its internal model.

To tune an observer it is necessary to make an initial guess of the actual values of these matrices. So how to choose the initial values? this question does not have a straight and convincing answer as it is not possible to know the values of the covariance of the state variables beforehand, in case we could, the stochasticity would be null and therefore this study would be totally unnecessary.

Some guidelines can be followed to achieve a satisfactory tuning,

- Infer variable statistics from available data: In case that history of several previous experiments were at hand, it would be helpful to try to determine the probability distribution of the state variables as well as the output variables and thereby define the required covariance matrices. This may lead to difficulties as most of the times not all state variables are measured. In the case of EKF and UKF, an attempt to describe the statistics of the state variables in terms of the parameters of the system was employed. This lead to the application of a Q -varying matrix filter, the reasoning behind this variant is explained in Chapter 5
- Automated optimization of the components of the Q and R matrices: Despite the importance of having a better tool or approach to tune a KF, very few information has been found in literature. Nevertheless some examples can be found. For instance, as the final goal of designing an observer is perhaps that of minimizing the estimation error of the state variables, it is possible to set an optimization problem in order to find optimal values for the covariance matrices. Rapp [35] presents an application using genetic algorithms. Powell [31] instead uses a downhill simplex algorithm to solve the optimization problem. No more details about these examples are given here as they were not employed in the present work, however, it is recommended to read the latter articles as it may be helpful in the case a system with a large number of variables is treated.

- Trial and error: another option is to randomly pick an initial value for the matrices. Once the initial guess has led the observer to converge, the rest of the tuning can be done by trial and error varying the components of the covariance matrices in order to increase or decrease the role each variable plays in the system final result. All in order to improve the estimation results according to a prestablished criterion. In our case study, the goal was to minimize the RMSE of the estimation error over a batch production.

In the case of KF variants this methodology is usually quite tedious, and the difficulty increases with the number of variables. Part of the complexity relies on the fact that it is hard to interpret what every component of the covariance matrices mean in terms of what happens in the system.

In the present work tuning was made by trial and error as the process model did not present a large number of variables, and fair results were quickly found.

The possibilities to tune all the KF extensions are basically the same that for EKF. Indeed UKF and EnKF also require finding appropriate matrices Q, R and P for the filters to work properly, however, these filters offer new possibilities to determine those matrices as will be shown next.

Highlights on tuning UKF

As presented in the description of this filter, in order to generate the sigma points it is necessary to specify the value of 3 parameters, namely n , q and κ . Values for n and q are set by the shape of the problem, as they are the state dimension, and the size of the process noise vector. However, the designer may still choose κ according to the insight or the information available concerning the system statistics. This parameter offers an extra knob to tune the filter. In our case, we followed the recommendation of choosing $\kappa = n - 3$ as we counted on the fact that measurement noise was Gaussian. However, the initial matrices still have to be chosen and again, as with EKF, they have the largest impact on the evolution of the filter, being this part still the most difficult task when tuning UKF.

Highlights on tuning EnKF

In the same way the other two KF variants, EnKF also needs an initial guess for the Q and R matrices, however, EnKF calculates these matrices based on the ensemble members as defined in a previous section. An important characteristic of EnKF is that the ensemble members can be chosen from an arbitrary distribution. Therefore, knowledge of the probability distribution of the state variables would be more accurately represented. In the present case study, the selection of the ensemble was chosen such that their covariance matrices were equivalent to those used to tune EKF and UKF, in fact the ensemble members were withdrawn from a Gaussian distribution, such that they could be comparable to the sigma-points from UKF. The obvious reason is because UKF and EKF were tuned first, and we bet on EnKF having a similar behavior like any of the other two filters if its matrices looked alike. Despite the fact that EnKF still had to be retuned, we had at least a reference point on UKF and EKF.

EnKF also offers the possibility to enlarge the ensemble to any dimension. Enlarging it gives dividends of improved estimation accuracy, nevertheless, there seems to be a threshold that

once it is reached, improvement in the estimation accuracy is very small when compared to the extra computational burden generated by evaluating the system in a large number of points.

In the present case, an ensemble of 20 members sufficed.

Tuning ELO

As seen from the description of the algorithm for this observer, it is possible to make use of pole placement, however, this procedure needs to be repeated every time the observer is iterated. There is also the possibility of determining a single useful observer gain, such tuning procedure was used in Mesbah et al. [23]. The approach was to find a gain that minimized the tracking error. Details can be found in the last reference.

Even though in Mesbah et al. [23] a deterministic system model was used, this tuning approach still worked for our application where an stochastic model was used instead. However the performance was heavily degraded due to the uncertainty introduced.

Tuning MHE

From the algorithm description we may say there are two components that may be used to tune this observer.

First an adequate calculation of the arrival costs. We skipped this part since due to the inherent variability of a batch process there is no need to evaluate the arrival costs described in the section of MHE.

Secondly, we have the horizon length N . As a general rule, the larger the horizon length, the more accurate the estimation results will be, however, this comes at the expense of an increase of the computational burden. A practical rule of thumb is that the length of the horizon should not be less than the number of the system states, Qu and Hahn [32].

Tuning for this observer was very straight forward as it gave good results without presenting any issues.

3-5 Summary

We described the observation problem at the beginning of this chapter and it was addressed under a Bayesian framework when the system under consideration is stochastic. Concepts such as smoothing, filtering and prediction which are commonly encountered in literature related to observers were defined and clearly differentiated. On this basis, we introduced the concept of observer and highlighted its importance within an NMPC framework.

A description of the error sources that will perturb the system under investigation was also given in this chapter. We presented the observers that will be used in our case study, we gave a general overview of each of them and presented their algorithms. The concept and ideas behind a varying system covariance matrix for UKF and EKF was also introduced.

From a control point of view we clearly established that we are mainly concerned with input-output behavior, therefore estimation of all state variables might not be necessary.

Finally we listed some considerations made when tuning the observers as they were implemented for computer simulation.

3-6 Remarks

The EKF, UKF and EnKF fall in the category of Gaussian Bayesian filters because they approximate the estimated variables probability distribution through a Gaussian probability distribution. Nevertheless, EnKF may distribute its ensemble members based on an arbitrary probability distribution. All these filters fall under the category of maximum likelihood.

Since the EKF algorithm requires differentiation it may face difficulties when dealing with non-differentiable and/or severely nonlinear systems. So, in this regard UKF and EnKF offer an advantage of not requiring differentiation.

Selection of the σ -points, as well as the number of points, for UKF is made under an established and deterministic procedure defined in its algorithm. This contrasts with the fact that ensemble members for the EnKF are randomly selected and their number is arbitrary. This characteristic places EnKF as a Monte Carlo method.

Since the MHE is an optimization based estimator it can handle constraints.

Estimation of all state variables may not be required since input-output behavior is what matters under a control point of view, and it is possible to use different models to describe a problem.

Determining adequate values for the elements of the covariance matrices when tuning observers such as EKF UKF and EnKF is complicated because it is usually hard to give an interpretation of every element. As such, tuning of observer involving a large number of variables may become tedious and very complicated. This gives an advantage to MHE because only a few parameters such as the estimation horizon need to be adjusted.

The algorithm to calculate the system time-varying matrices for EKF and UKF allowed for an easier tuning of those observers.

Chapter 4

Case study

In this chapter we present some concepts in crystallization. Afterwards, we present the case study introducing the model describing the system under investigation and present in a following section an observability analysis which is needed prior to the design of an observer.

4-1 Crystallization

Crystallization is the (natural or artificial) process of formation of solid crystals precipitating from a solution, melt or more rarely deposited directly from a gas. Crystallization is also a chemical solid-liquid separation technique, in which mass transfer of a solute from the liquid solution to a pure solid crystalline phase occurs.

Crystallization from the melt

A melt generally refers to a multicomponent liquid mixture that solidifies upon cooling. Melt crystallization is the common term applied to the crystallization of such systems to achieve ultra purification of an often organic compound to purities of 99.9% to 99.999%. Melt crystallization is merely used as a purification technique, [1].

Crystallization from solution

Crystallization is mostly applied as a single step solid-liquid separation technique and in those cases the compound to be crystallized is already dissolved in some solvent. This type is also applied to obtain a product in a particular dispersed solid form (e.g. very fine or coarse particles), [1].

Before explaining crystallization as a process, we introduce the concept of supersaturation.

- supersaturation can be understood as a solution that contains more solute material that could be dissolved by the solvent under normal circumstances.

The concept normal circumstances means that given a system under certain temperature, pressure and volume, all must remain unchanged in order to evaluate a property of the system. For instance, solubility of salt in water has a determined value given specific pressure and temperature values.

In other words, supersaturation refers to a state in which the liquid (solvent) contains more dissolved solids (solute) than can ordinarily be accommodated at that temperature.

Crystallization as a process

Once supersaturation has been reached, two main processes by which crystals form may take place: nucleation and crystal growth.

According to [1], nucleation can be split in two mechanisms, that is to say,

- primary nucleation, which is a step where solute molecules start to gather into clusters, roughly speaking this means there is an elevation of solute concentration in a small region that becomes stable under the current operating conditions. These stable clusters constitute the nuclei. However when the clusters are not stable, they dissolve. Therefore, the clusters need to reach a critical size in order to become stable nuclei. Such critical size is dictated by the operating conditions (temperature, supersaturation, etc.). It is at the stage of nucleation that the atoms arrange in a defined and periodic manner that defines the crystal structure.
- secondary nucleation, on the other hand, refers to the birth of nuclei at the interface of the parent crystals. Contrary to the relatively high supersaturations required for primary nucleation, secondary nucleation already occurs at low to moderate values of the supersaturation. For medium to well soluble salts it is considered to be the main source of nuclei. The different types of secondary nucleation are named after their origin; however, we will not present all of them as they are not of interest for our case study.

The crystal growth is the subsequent growth of the nuclei that succeed in achieving the critical cluster size. Nucleation and growth continue to occur simultaneously while the supersaturation exists. Supersaturation is the driving force of the crystallization, hence the rate of nucleation and growth is driven by the existing supersaturation in the solution. Depending upon the conditions, either nucleation or growth may be predominant over the other, and as a result, crystals with different sizes and shapes are obtained (control of crystal size and shape constitutes one of the main challenges in industrial manufacturing, such as for pharmaceuticals). Once the supersaturation is exhausted, the solid-liquid system reaches equilibrium and the crystallization is complete, unless the operating conditions are modified from equilibrium so as to supersaturate the solution again.

Crystallization methods

In this work we will focus on crystallization from solution, there we can distinguish four crystallization methods, [1]:

- cooling crystallization, which is carried out based on the fact that most chemical compounds, dissolved in most solvents, show the so-called direct solubility, that is, the solubility threshold increases with temperature. So, whenever the conditions are favorable, crystal formation results from simply cooling the solution.
- evaporative crystallization, the solid phase is formed by evaporation of the solvent. As crystallization proceeds from a boiling solution, at a chosen temperature, the pressure and the mother-liquor concentration are fixed.
- precipitation, two soluble reactants are mixed to form a sparingly soluble product. Since the reactant streams are generally concentrated, high supersaturations are created. Therefore, conversion of the solutes into solid particles is a fast process.
- addition of an anti-solvent, crystallization occurs because the solute has a low solubility in the mixed solvent. Since the addition of the anti-solvent also dilutes the mixture, the decrease in solubility should largely exceed this dilution effect.

Besides can take place in two so-called operation modes, either batchwise or in continuous fashion.

According to [1], in the continuous mode, an undersaturated solution or melt containing the solute to be crystallized is continuously fed to the crystallizer. Here the solid is formed as a dispersed phase, which is kept in suspension by some mixing device. The suspension containing the dispersed solids and a supersaturated solution leaves the reactor continuously, for both cooling and evaporative crystallization. The slurry produced in the crystallizer is generally transported to a centrifuge or filter, where the crystals are separated from the mother liquor and washed. In batch crystallization, the crystallizer is initially filled with an undersaturated solution. After a prestablished batch time the crystallizer contents are discharged. Batch crystallization is in a few cases applied without agitation. In batch crystallization, the crystallizer is initially filled with an undersaturated solution. After a pre-established batch time the crystallizer contents are discharged. Batch crystallization is in a few cases applied without agitation. Both modes are depicted in Figure 4-1.

Characterization

Crystals' specifications define crystals' quality on the basis of characteristics like crystal size distribution of the product, shape, polymorphism, mother liquor inclusions, impurities in the crystal lattice or degree of agglomeration, but to name a few. As we will be interested in obtaining a crystal size distribution with certain characteristics, we define it next.

Crystal size distribution (CSD)

One of the main characteristics of a product is its CSD or in case of agglomerated or noncrystalline particles its particle size distribution. The CSD is important for the product quality, but also influences the performance of the process, the separation of the crystals from the mother liquor, and the subsequent drying of the crystals.

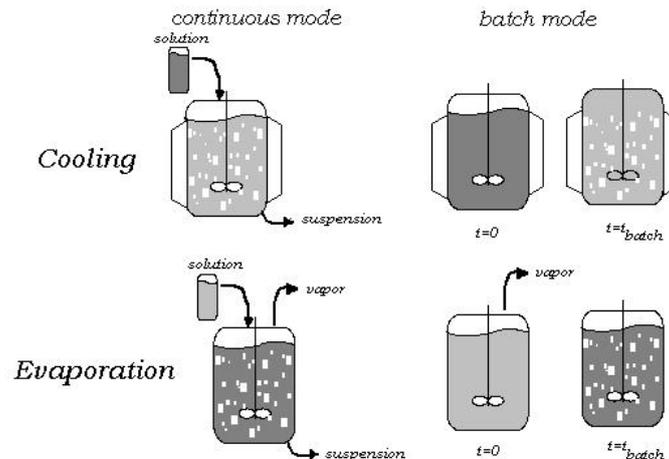


Figure 4-1: Modes of operation in crystallization. The tone of grey indicates the solution concentration, and the dots the presence of crystals. Adapted from [1]

The dominant properties of the product will usually be the average crystal size and the width of the distribution, but the zeroth to third moments of the CSD are also linked to important physical parameters of the product, since they are related to the total number, total length, total surface area and total mass of the crystals.

Crystal size

How to measure crystals size can be achieved in different ways, it also receives different names accordingly. We now give a list with their names followed by the methodology to determine the crystals' size

- length, maximal visible length
- sieve diameter, width of the minimum square aperture through which the particle will pass
- volume diameter, diameter of a sphere having the same volume as the crystal
- surface diameter, diameter of a sphere having the same surface area as the crystal
- projected area diameter, diameter of a sphere having the same projected area as the crystal viewed from a fixed direction

Directly correlated with the definition of the crystal size are the surface and volumetric shape factors k_a and k_v . These factors appear to compensate for inaccuracies due to considering crystals to have a perfect geometrical shape. Their values depend on the criterion used to define the crystal size.

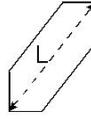


Figure 4-2: hypothetical length of a crystal, L

Representations of the CSD

According to [1], two types of CSD representations are normally considered,

- the cumulative oversize number distribution $N(L)$ $\{\#/m^3\}$ and the mass distribution $M(L)$ $\{kg/m^3\}$

$$N(L) = \int_0^L n(L)dL \quad (4-1)$$

where N represents the number of crystals and L their length.

$$M(L) = \int_0^L m(L)dL = \rho k_v \int_0^L L^3 n(L)dL, \quad (4-2)$$

- the number (or population) density distribution $n(L)$ and the mass density distribution $m(L)$

$$n(L) = \frac{dN(L)}{dL} \quad (4-3)$$

and

$$m(L) = \frac{dM(L)}{dL} \quad (4-4)$$

- the moments of the CSD

$$m_j = \int_0^\infty L^j n(L)dL \quad (4-5)$$

Table 4-1: Statistical moments and their physical interpretation

Physical interpretation	nomenclature
total number of crystals	m_0
total length	m_1
total surface area	$k_a m_2$
total volume	$k_v m_3$
total mass	$\rho k_v m_3$

The shape of cumulative and density distributions of a crystal population can be observed in Figure 4-3, note the resemblance with a probability density function. As such, the CSD curve gives information of how the crystals are dispersed around certain lengths, it is desirable to have a narrow curve and the value determining the location of the peak (on the size axis)

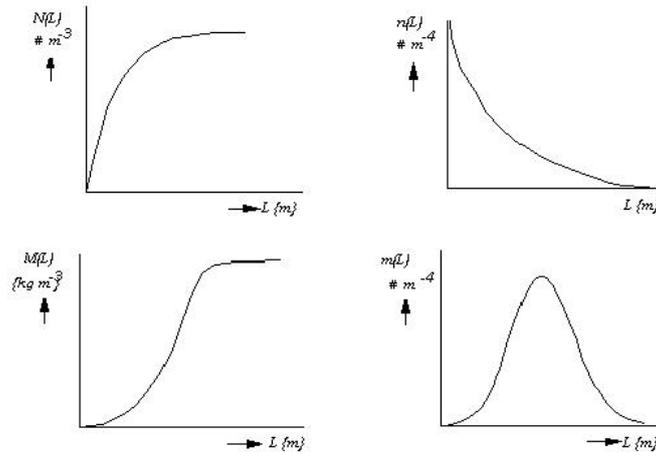


Figure 4-3: Cumulative and density distributions of a crystal population. Taken from [1]

close to the desired crystal size. A curve like that would indicate a majority of the crystals being very close to a desired crystal size with a small, defined somehow, length variability around it.

Knowing the CSD is desirable, nevertheless, valuable information can be inferred from the leading statistical moments of a CSD, in fact, once we know them we should be able to draw some conclusions about the characteristics of the product.

4-2 Description of the case study

We introduce in Figure 4-4 a sketch of the setup that gave rise to our case study.

There we distinguish the following elements,

- crystallizer, is the vessel where the crystallization process takes place, it contains an impeller used for stirring, and on the outside, it is surrounded by a cooling/heating jacket.
- cooling/heating jacket, it surrounds the crystallizer. Water running inside the jacket allows the operator to supply or withdraw heat from the system. This is the system input.
- seeding vessel, it contains the crystal seeds used to initialize the crystallization process.
- liquid sonic probe, it is used to acquire measurements while the system is running.

The process can be described as follows: crystals known as seeds lie on the seeding vessel. In the meantime, the crystallizer is filled up with a solution containing the solvent, and the solute which is the material to be crystallized. The cooling/heating jacket allows to control the temperature in the system and brings it to a boiling point such that solvent evaporates

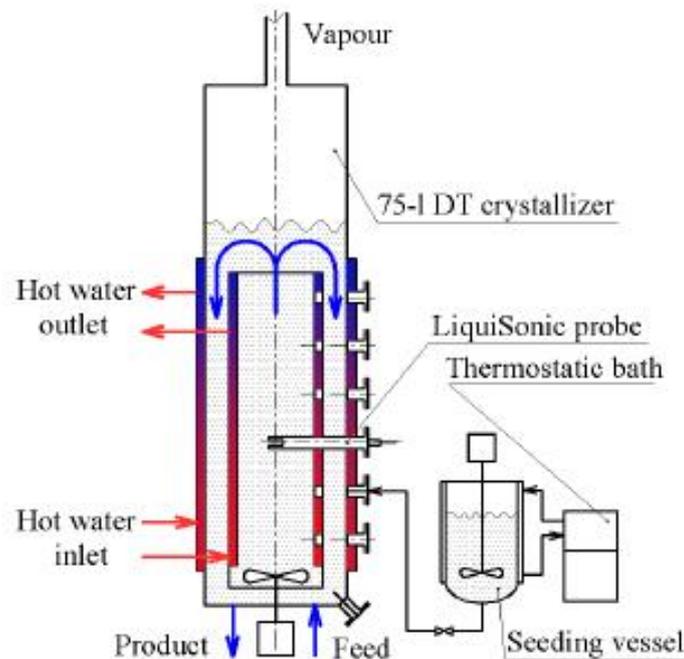


Figure 4-4: 75-liter evaporative fed-batch crystallizer setup. From Mesbah et al. [24]

leaving the crystallizer through an exhaust pipe. This brings the saturation level up to a desired supersaturation value. When the supersaturation has reached a specific value the seeds are fed into the crystallizer, and the crystallization process begins. Due to the system running at boiling temperature it is an isothermal process. The solvent lost by evaporation is replaced by liquid fed into the crystallizer via a feeding line to maintain constant volume. Measurements are taken via the probe inside the crystallizer, and once the batch time is reached the process is shut down.

It is important to notice that,

- this is a batch process.
- heat is the control input to the system and is acquired via the cooling/heating jacket.
- controlling the supersaturation value is vital as it is the driving force for crystal growth.
- measurements can be carried on online, but not all variables of interest can be measured.

After describing the process, and based on the theory about crystallization presented in previous sections, we can state that *our case study is an evaporative fed-batch crystallization process, it takes place in a 75-liter vessel, and the working substance is ammonium-sulphate dissolved in water.*

For our case study, secondary nucleation gives a complete picture of the crystallization process since it is the dominant nucleation mechanism occurring in seeded batch crystallizers, Mesbah et al. [24]. As we already described, *seeds* are placed in the crystallizer at the beginning of the batch-time in order to start up the nucleation process, and larger crystals start building up when solute particles bond onto the surface of the seeds.

Process model

As we presented in Chapter 2, our analysis will take place in an NMPC framework, hence we should count with a model describing our case study.

The mathematical models of solution crystallization processes are typically obtained through the application of a population balance equation, mass balance equations for solvent and solute, energy balance equation, and expressions describing the variation of the equilibrium concentration. The population balance equation accounts for the evolution of crystal particles along temporal and size domains. Under the assumptions of mixed suspension, constant crystallizer volume, nucleation of crystals of infinitesimal size, and negligible breakage and agglomeration, the dynamic population balance equation for a semi-batch crystallization process, according to Randolph and Larson [33], simplifies to:

$$V \frac{\partial n(L, t)}{\partial t} + V \frac{\partial (n(L, t)G)}{\partial L} = VB(L, t) - Q_p n(L, t) \quad (4-6)$$

where n is the number density ($\# \cdot m^{-3} \cdot m^{-1}$), G is the growth rate ($m \cdot s^{-1}$), B is the nucleation rate ($\# \cdot m^{-3} \cdot m^{-1} \cdot s^{-1}$), L is the characteristic crystal size (m), V is the crystallizer volume (m^3), and Q_p is the sample stream flow rate ($m^3 \cdot s^{-1}$). Numerical solution of the population balance equation often requires considerable computational effort that might render the real-time implementation of model based control strategies infeasible. The method of moments is, therefore, applied to equation (4-6) in order to convert the population balance equation into a set of computationally affordable Ordinary Differential Equations (ODEs). Defining the i^{th} moment of $n(L, t)$ as:

$$m_i = \int_0^\infty L^i n(L, t) dL \quad i = 0, \dots, 4 \quad (4-7)$$

multiplying equation (4-6) by $L^i dL$ and, subsequently, integrating over the entire crystal size domain result in the following set of ODEs that describes the evolution of the leading statistical moments of the (CSD) in time:

$$\frac{dm_0}{dt} = B_0 - \frac{m_0 Q_p}{V} \quad (4-8)$$

$$\frac{dm_i}{dt} = iGm_{i-1} - \frac{m_i Q_p}{V} \quad i = 0, \dots, 4 \quad (4-9)$$

Here B_0 represents the total rate of nucleation ($\# \cdot m^{-3} \cdot m^{-1} \cdot s^{-1}$). The choice of the moment model in this study is justified by the relatively low supersaturation levels in batch runs due to a large seed loading, so that the effect of secondary nucleation is minimized and, therefore, the CSD dynamics is mainly governed by the crystal growth, Mesbah et al. [24]. The empirical expressions realized for the total nucleation rate, and the size independent crystal growth rate are as follows:

$$B_0 = k_b m_3 G \quad (4-10)$$

$$G = k_g (C - C^*)^g \quad (4-11)$$

The nucleation rate constant k_b , the growth rate constant k_g , and the growth rate exponent g are the kinetic parameters corresponding to the ammonium sulphate-water system. Furthermore C and C^* are the solute concentration and the equilibrium concentration, respectively; their difference determines the driving force of the crystallization process, known as the supersaturation. In the face of isothermal operation of the evaporative crystallizer, a single expression is derived for the solute concentration using the mass and energy balance equations:

$$\frac{dC}{dt} = \frac{\frac{Q_p(C^* - C)}{V} + 3k_v G m_2 (k_1 + C)}{1 - k_v m_3} + \frac{k_2 H_{in}}{1 - k_v m_3} \quad (4-12)$$

with constant coefficients given by:

$$k_1 = \frac{H_v C^*}{H_v - H_L} \left(\frac{\rho_c}{\rho_L} - 1 + \frac{\rho_L H_L - \rho_c H_c}{\rho_L H_v} \right) - \frac{\rho_c}{\rho_L} \quad (4-13)$$

$$k_2 = \frac{C^*}{V \rho_L (H_v - H_L)} \quad (4-14)$$

where k_v is the crystal volumetric shape factor, H_{in} is the heat input to the crystallizer (kW), ρ_L is the saturated solution density ($kg \cdot m^{-3}$), and ρ_c is the density of crystals ($kg \cdot m^{-3}$). H_L , H_c and H_v are the solution, crystals and vapor specific enthalpies ($kJ \cdot kg^{-1}$), respectively.

It follows that the dynamic behavior of the system under investigation is governed by a set of differential algebraic equations (DAEs), equations (4-8) to (4-14). Hence, the five leading moments of the CSD, and the solute concentration are the state variables determining the dynamics of the system. *We mentioned before that measurements are only available for the moments of the CSD. As the concentration is not available, it must be estimated, we will do this by means of an observer.*

4-3 Observability analysis

As we will employ an observer, part of its design requires what is known as observability analysis.

The observability problem consists of investigating whether there exists relations binding the state variables to the inputs, outputs and their time derivatives and thus locally defining them uniquely in terms of measurable quantities. If no such relations exist, the initial state of the system cannot be deduced from observing its input-output behavior.

Short mathematical background

In practical terms observability can be interpreted as the possibility of reconstructing the state vector from the information contained in measurements taken from the system.

The concept of observability is useful in solving the problem of reconstructing unmeasurable variables from measurable ones in the minimum length of time.

The concept of observability is very important because, in practice, the difficulty encountered with state feedback control, and some other control variants, is that some of the state variables are not accessible for direct measurement, with the result that it becomes necessary to estimate the unmeasurable state variables in order to construct control signals. Such estimates are possible only if the system is observable.

Determining observability of a nonlinear system requires some basic knowledge in systems analysis, all the required theory and definitions can be found in Nijmeijer and van der Schaft [29], Jeltsema and Scherpen [17], and Hermann and Krener [16]. We mainly follow Nijmeijer and van der Schaft [29].

According to the terminology used in differential geometry, we shall denote a vector function $f : \mathbb{D} \rightarrow \mathbb{R}^n$ as a vector field on the domain $\mathbb{D} \subset \mathbb{R}^n$. The intuitive reason for this term is that every vector function f corresponds with a field of vectors in an n -dimensional space. A vector field is an n -dimensional column. Furthermore, the transpose of a vector field is said to be a covector field. Hence, a covector field is an n -dimensional row. An example of a covector is: let $h : \mathbb{D} \rightarrow \mathbb{R}$, then the differential of h is a covector field, defined by

$$dh = \frac{\partial h}{\partial x} = \left(\frac{\partial h}{\partial x_1}, \dots, \frac{\partial h}{\partial x_n} \right). \quad (4-15)$$

We will limit ourselves to smooth vector fields, by smooth we mean that the function $f(x)$ has continuous partial derivatives of any required order.

Lie derivatives

Given a scalar function $h(x)$ and a vector field $f(x)$, we define a new scalar function $L_f h(x)$, called the Lie derivative of $h(x)$ with respect to $f(x)$.

Definition 1. *Let $h : \mathbb{D} \rightarrow \mathbb{R}$ be a smooth function, and $f : \mathbb{D} \rightarrow \mathbb{R}^n$ be a smooth vector field on the domain $\mathbb{D} \subset \mathbb{R}^n$. Then the Lie derivative of $h(x)$ with respect to $f(x)$ is a scalar function defined by*

$$L_f h(x) = \frac{\partial h}{\partial x} f(x). \quad (4-16)$$

Repeated Lie derivatives can be defined recursively as

$$\begin{aligned} L_f^0 h(x) &= h(x) \\ L_f^1 h(x) &= L_f h(x) \\ &\vdots \\ L_f^i h(x) &= L_f(L_f^{i-1} h(x)), \quad i = 1, 2, 3, \dots \end{aligned} \quad (4-17)$$

Similarly, if $g(x)$ is another vector field, then the scalar function $L_g L_f h(x)$ is defined as

$$L_g L_f h(x) = \frac{\partial L_f h}{\partial x} g(x). \quad (4-18)$$

Let us consider a smooth control-affine system together with an output map

$$\begin{aligned} \dot{x} &= f(x) + \sum_{j=1}^m g_j u_j, & u &= (u_1, \dots, u_m) \in U \subset \mathbb{R}^m, \\ y_i &= h_i(x), & i &= 1, 2, \dots, p, \end{aligned} \quad (4-19)$$

where $h = (h_1, h_2, \dots, h_p)^T : \mathbb{R}^n \rightarrow \mathbb{R}^p$ is the smooth output map of the system. It is defined that $y(t, 0, x_0, u) = h(x(t, 0, x_0, u))$ denotes the output of (4-19) for u and initial state $x(0) = x_0$.

Some definitions are required to state observability,

Definition 2. Two states $x_1, x_2 \in \mathbb{R}^n$ are said to be indistinguishable, denoted $x_1 I x_2$, if for every admissible input function u , the output function $t \rightarrow y(t, 0, x_1, u)$, $t \geq 0$, of the system for initial state $x(0) = x_2$, are identical on their common domain of definition. The system is called observable if $x_1 I x_2$ implies $x_1 = x_2$.

Notice that this definition of observability does not imply that every input function distinguishes points of \mathbb{R}^n . However, if the output is the sum of a function of the initial state and a function of the input (as it is for linear systems) then it is easily seen that if some input distinguishes between two initial states then every input will do.

Definition 3. An input u is universal on $[0, t]$ if for every pair of distinct states $x_1 \neq x_2$, there exists $\tau \in [0, t]$ such that $h(x(t, 0, x_1, \tau)) \neq h(x(t, 0, x_2, \tau))$.

Definition 4. A non universal input is called singular

Definition 5. Consider the nonlinear system (4-19). The observation space \mathbf{O} of (4-19) is the linear space (over \mathbb{R}) of functions on \mathbb{R}^n containing h_1, \dots, h_p , and all repeated Lie derivatives

$$L_{z_1} L_{z_2} \dots L_{z_k} h_j, \quad j \in p, \quad k = 1, 2, \dots \quad (4-20)$$

with $z_i, i \in k$, in the set $\{f, g_1, \dots, g_m\}$.

The observation space \mathbf{O} defines the observability codistribution, denoted as \mathbf{O}_c , by setting $\mathbf{O}_c = d\mathbf{O}$, ($d(\cdot)$ denotes the differential operator as defined in (4-15)) i.e.,

$$\mathbf{O}_c = \text{span}\{dh_1(x_0), \dots, dh_p(x_0), dL_{z_1} L_{z_2} \dots L_{z_k} h_j(x_0)\}, \quad j \in p, k = 1, 2, \dots \quad (4-21)$$

where $z_i, i \in k$, is a vector field in the set $\{f, g_1, \dots, g_m\}$. The distribution \mathbf{O}_c is invariant with respect to f, g_1, \dots, g_m and it is contained in the kernel of $\text{span}\{dh_1, \dots, dh_p\}$. If \mathbf{O}_c is nonsingular, it is also involutive. The main theorem concerning local observability, Nijmeijer and van der Schaft [29], can be stated now.

Theorem 1. *Consider the system (4-19) on \mathbb{R}^n . If $\mathbf{O}_c(x_0)$ is of constant dimension, with*

$$\dim(\mathbf{O}_c(x_0)) = n, \quad (4-22)$$

then the system (4-19) is locally observable at x_0 .

Observability analysis for the moment model of the 75-liter fed-batch evaporative crystallizer

The state space representation of the analyzed system is given by

$$\begin{pmatrix} \dot{m}_0 \\ \dot{m}_1 \\ \dot{m}_2 \\ \dot{m}_3 \\ \dot{m}_4 \\ \dot{C} \end{pmatrix} = \begin{pmatrix} B_0 - \frac{m_0 Q_p}{V} \\ Gm_0 - \frac{m_1 Q_p}{V} \\ 2Gm_1 - \frac{m_2 Q_p}{V} \\ 3Gm_2 - \frac{m_3 Q_p}{V} \\ 4Gm_3 - \frac{m_4 Q_p}{V} \\ \frac{Q_p(C^* - C)}{V} + \frac{3k_v Gm_2(k_1 + C)}{1 - k_v m_3} + \frac{k_2 H_{in}}{1 - k_v m_3} \end{pmatrix}, \quad (4-23)$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \begin{pmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \\ C \end{pmatrix}, \quad (4-24)$$

$$B_0 = k_b m_3 G \quad (4-25)$$

$$G = k_g (C - C^*)^g, \quad (4-26)$$

with the variables defined as in previous sections. It can be recasted in the form of equation (4-19) and is as follows

$$f = \begin{pmatrix} k_1 m_3 k_g (C - C^*)^g - \frac{m_0 Q_p}{V} \\ k_g (C - C^*)^g m_0 - \frac{m_1 Q_p}{V} \\ 2k_g (C - C^*)^g m_1 - \frac{m_2 Q_p}{V} \\ 3k_g (C - C^*)^g m_2 - \frac{m_3 Q_p}{V} \\ 4k_g (C - C^*)^g m_3 - \frac{m_4 Q_p}{V} \\ \frac{Q_p(C - C^*)}{V} + \frac{3k_v k_g (C - C^*)^g m_2 (k_1 + C)}{1 - k_v m_3} \end{pmatrix} \quad (4-27)$$

$$g = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{k_2}{1-k_v m_3} \end{pmatrix} \quad (4-28)$$

$$h = \begin{pmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \\ C \end{pmatrix} \quad (4-29)$$

where $m_0, m_1, m_2, m_3, m_4,$ and C conform the state variables, therefore the dimension n of the system is 6.

The observability of the system will be evaluated in two cases; first when the full state variables are available to be measured, and second when the sixth state (the concentration C) is not available.

Case 1: full state available

In this case f, g and h remain as defined by (4-27), (4-28) and (4-29) respectively. Therefore $h_i = m_{i-1}$ for $i = \{1, 2, 3, 4, 5\}$, and $h_6 = C$.

Based on the theory of the previous section, the observability codistribution can be obtained by only calculating dh_i for $i = \{1, 2, 3, 4, 5, 6\}$, since this gives us the following codistribution:

$$\mathbf{O}_c = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad (4-30)$$

which as can be seen has rank=6. Thus the system is locally observable at any point x_0 since there is no dependence of the rank condition on any of the state variables.

Case 2: sixth state (concentration C) not available

In this case the state space representation of the system suffers a slight change, f and g remain the same, but h is now defined by only five components $h_i = m_{i-1}$ for $i = \{1, 2, 3, 4, 5\}$, this is, we can only measure five state variables while the one missing is the concentration.

On this basis, we proceed as before, the observability codistribution can be obtained by first calculating dh_i for $i = \{1, 2, 3, 4, 5\}$, and also $dL_f(h_1)$. This gives,

$$\mathbf{O}_c = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -\frac{Q_p}{V} & 0 & 0 & k_1 k_g (C - C^*)^g & 0 & k_1 k_g m_3 g (C - C^*)^{g-1} \end{pmatrix}, \quad (4-31)$$

the rank of the codistribution above has rank=6 and the system will be locally observable as long as the product flow rate, and the difference between the actual concentration and the saturation concentration do not become zero simultaneously.

Given a dynamical system, the observer aims at obtaining an estimate of the current state by only using available measurements. For linear systems, the property of observability, characterized by the Kalman rank condition, guarantees the possibility to indeed design an observer. In the case of nonlinear systems, observability is not enough, basically because this property in general depends on the input of the system. In other words, observability of a nonlinear system does not exclude the existence of inputs for which two distinct initial states cannot be distinguished by using the knowledge of the measured output. The moment model for the 75-liter crystallizer is observable in the two cases analyzed here, therefore constructing an observer for the system is possible, as long as also properties like the absence of singular inputs are fulfilled.

4-4 Summary

We began this chapter introducing elemental concepts in crystallization, then presented some crystallization methods and finally gave an overview of how the product may be typically characterized. There we introduced the concept of crystal size distribution, useful in this case study.

A rather detailed overview of the process under investigation was given, and wrote the equations describing the system model.

Based on the latter model, we put into effect an observability analysis.

4-5 Remarks

The system model used in the present work is known as a moment model. It is derived from the more complex population balance equation, which is a partial differential equation, and some energy and mass balance equations. The more complex model can be transformed into a set of nonlinear differential and algebraic equations (DAEs) thanks to the fulfillment of certain characteristics already explained in the corresponding section. This set of DAEs can be recasted as a nonlinear control-affine system consisting of 6 state variables. This means the system can be put in the form of,

$$\begin{aligned}\dot{x} &= f(x) + \sum_{j=1}^m g_j u_j, \quad u = (u_1, \dots, u_m) \in U \subset \mathbb{R}^m, \\ y_i &= h_i(x), \quad i = 1, 2, \dots, p,\end{aligned}\tag{4-32}$$

that is, the control input or the control input times a another relation is additive with respect to the time derivatives of the state variables.

One of the states is the solute concentration and given the fact it cannot be measured in real-time, it needs to be estimated.

Once the model was in the latter form, we put into effect an observability analysis.

Indeed the system is observable and then we can proceed with the implementation of the observers.

Simulation Results

Here we analyze and interpret, on the basis of the theory presented in previous chapters, data generated by computer simulations. We make use of the observers introduced before in a plant simulator of our case study. We first describe the scenarios from which data was acquired, establish the evaluation criteria and finally we show and explain the results of such simulations.

5-1 Analyzed Scenarios

The mathematical model, used by the plant simulator and the observer, is that given by equations (4-8) to (4-14), the state variables x_1 to x_5 represent the first five leading statistical moments of the crystal size distribution of the system (m_0 to m_4), and x_6 represents the solute concentration (C). It should be noted that measurements are available only for the first 5 variables, not for the concentration, so the latter can only be estimated, here strives the importance of an observer in our case study. Nevertheless as all the results come from computer simulations, we are able to determine the actual value of the solute concentration for the process and then compare it to the estimated value given by the observers.

The nonlinear observers employed were:

- moving horizon estimator (MHE)
- extended Luenberger observer (ELO)
- extended Kalman filter (EKF) with two variants, the classical implementation and that where a time-varying Q matrix is used. Please bear in mind that this is the covariance matrix of the noise acting on the plant and the notation is standard for all the observers used here. Support to apply this variant is explained in Chapter 2
- unscented Kalman filter (UKF), also with two variants, the classical one and that with a time-varying Q matrix

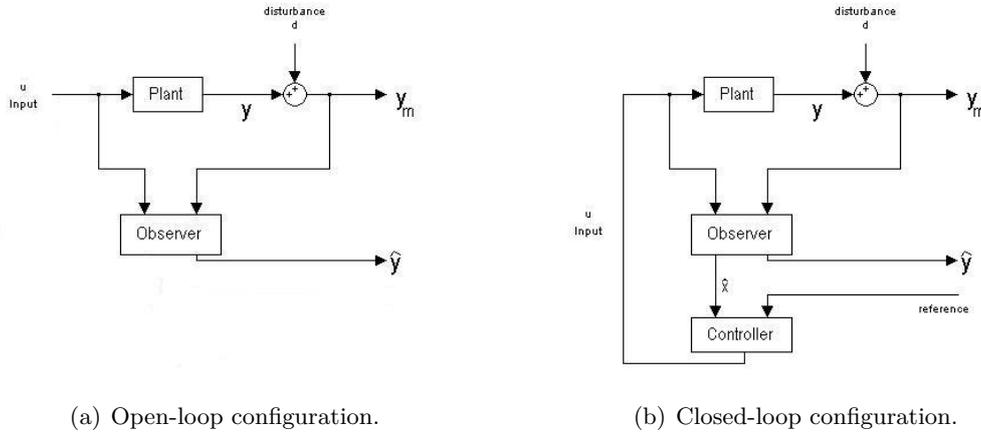


Figure 5-1: open- and closed-loop configurations

- and the ensemble Kalman filter (EnKF)

We can now distinguish two main branches for the simulations, namely open- and closed-loop scenarios.

A simple open-loop implementation may be used to compare computational time and estimation accuracy between observers, once this is done, it is of interest to determine if there is a correlation between estimation accuracy and control performance. However, an open-loop implementation with an optimal actuator profile computed off-line can lead to unexpected results due to a mismatch between the process and the model, irreproducible start-up or/and process disturbances. These limitations underline the need for an optimal model-based control strategy implemented on-line in a closed-loop mode. The closed-loop implementation will also be tested under the influence of the different error sources already described in Chapter 4. It is important to determine which error source is more detrimental to the control performance and whether it is possible to counteract the effects of such error source.

Open-loop

The nonlinear observers are first analyzed in an open-loop framework where no controller is present in the setup, and the input profile is determined beforehand. Figure 5-1(a) shows the arrangement of the plant and observer in open-loop. The plant and the observer are fed by the same inputs, the plant outputs correspond to vector y . Every 100 s it is possible to measure the system outputs, however, our measurements are corrupted by noise and we obtain vector y_m , these values are then fed into the observer, which must filter out the noise and deliver the output-estimate vector \hat{y} .

For the open-loop case, two scenarios are analyzed, namely the nominal case, where only measurement noise is present on the plant as an error source, and the uncertain initial conditions case.

Closed-loop

Under this scenario the observer uses measurements from the plant to update the values of the states, the controller also makes use of the updated states, which closes the loop, and based on this, the built-in optimizer determines the input to the system as can be understood from Figure 5-1(b).

For the analysis we performed, yet we can divide each branch in subcases where different sources of error are introduced,

- nominal case- the only error source is due to stochastic measurement error. This error randomly pushes the values of measurements up or down and in general it is expected not to shift the mean value of a measurement.
- uncertain initial conditions- plant, and the pair observer-controller are initialized at different points in state-space. In our case study, uncertain initial conditions are due to improper seeding.
- model mismatch- the mathematical models used to simulate the plant, and by both, observer and controller, are not exactly equal. We may think of this error source as erroneous values of certain parameters used to model the dynamics of a system because they cannot be precisely determined.
- systematic measurement error- systematic errors tend to be consistently either positive or negative therefore biasing the average value of a measurement. In practice an uncalibrated device may give rise to this error.
- general case- in this case all the error sources mentioned before are introduced simultaneously in the simulation. This scenario is closer to a real situation.

As a complementary note, stochastic measurement error is always present in every scenario.

For the closed-loop case all the scenarios mentioned before were simulated.

Under the closed-loop setup the observers are coupled to a nonlinear controller described by Mesbah et al. [24]. The closed-loop setup is shown in Figure 5-1(b). Important differences with respect to the open-loop case are the presence of the reference trajectory, that must be followed by the plant, and the feedback introduced by feeding the controller with information from the observer. This reference is used by the controller which aims to follow it as close as possible. Another outstanding point is the interaction of the observer, as it determines the values of the state variables, it determines the position of the system in state-space, and therefore, provides the controller with valuable information so as to where the system should be driven to.

Coupling a dynamic optimizer to an stochastic observer is accomplished by following the structure and implementation found in Mesbah et al. [24]. Simulation of the different scenarios require the solution of algebraic differential equations that describe the dynamics of the elements in the setup, that is to say plant, observer and controller. Different solvers were used for integrating each set of differential equations, they are listed below

- Plant: MATLABTM ode23s solver
- MHE: DASPK solver
- ELO: MATLABTM ode15s solver
- For the remaining filters a discretized version of the continuous-time state-space model is used using an explicit Euler scheme.

The MATLABTM solvers we chose allow for numerically stable simulation results. In the case of explicit Euler, a previous analysis of the step size was done to avoid numerical stability issues.

As we are evaluating the usefulness of different observers, they must be evaluated under equal circumstances, therefore, the type and magnitude of errors introduced to the system are kept equal for every observer. Even more, as stochastic measurement error plays an important role in the simulations, the set of values used for every simulation were identical for every observer as well. All this precautions should allow for a fair comparison.

Finally, every simulation exemplifies a real-life batch run, which is equivalent to 10800 s.

5-2 Evaluation Criteria

Results evaluation is divided in two parts, the first one is to evaluate the performance of the observers from a systems and control point of view, where the estimation accuracy of every observer, its computational demand, and finally the fulfillment of a control criterion is revised. It must be noted that evaluating the control performance is relevant only in the closed-loop simulations.

The second evaluation part takes on a process oriented approach, which on the one hand is interested in the fulfillment of quality specifications of the product, of vital importance in this application, and on the other hand, maximization of the production yield.

The evaluation of each scenario, described previously, is based on data generated by 50 computer simulations of the system under investigation, that is, the plant simulator, observer and controller under the NMPC framework.

Systems and control evaluation

First, evaluating the observer is based on two main features, namely

- estimation accuracy- it is determined based on the normalized root mean squared error (NRMSE) between each component of the uncorrupted plant states x and the corresponding component of the estimate given by the observer \hat{x} , the formula to calculate this quantity is

$$NRMSE = \sqrt{\sum_{i=0}^n \left(\frac{x_i - \hat{x}_i}{x_i} \right)^2} \quad (5-1)$$

where the index i the i^{th} data point, and n represents the number of data points used for the calculation. In this case, $n = 50$, the number of simulations we set in the previous section.

- CPU time- it is taken as a measure of an observer computational demand and it is determined as the average CPU time required to run the algorithm of an observer from the moment that a measurement is available until the observer delivers the variables' estimates.

The criterion to evaluate the control performance is given by

- the tracking error- defined as the difference between the value of a desired trajectory and the actual value of the system. It is evaluated on the basis of a NRMSE as well,

$$NRMSE = \sqrt{\sum_{i=0}^n \left(\frac{r_i - G_{r_i}}{r_i} \right)^2} \quad (5-2)$$

where r represents the value of the reference trajectory and G_{r_i} ¹ is the actual value of the system.

For our case study $r = 2.5 \cdot 10^{-8} \text{ m/s}$ which represents the desired crystal growth rate.

Two data subsets can be identified, see Figure 5-2.

- case 1- the NRMSE is calculated at the same time instant, k , that a measurement is available and along the 50 simulations. This is equivalent to calculate the NRMSE over the projections of the estimation error curves over the plane (VV-t) at every time k as depicted in Figure 5-2. These results will be referred as the NRMSE per time step, or vs. time.
- case 2- the NRMSE is calculated over the estimation errors generated along each of the 108 times a measurement is performed in one batch run. This gives us a picture of how the model-based control strategy behaves to changes in the noise applied to the system in each simulation. This is equivalent to calculate the NRMSE over the projections of every simulation on the plane (VV-S#). These results will be referred to as NRMSE vs. simulation.

To summarize, in the open-loop case we only evaluate the performance of the observer while in the closed-loop case, both the controller and observer are evaluated. The performance of the observers is evaluated based on their estimation accuracy and their computational demand (the latter measured by their required CPU time). Furthermore, in the closed-loop case the control performance is also analyzed, the output of the system is ought to follow a reference trajectory corresponding to a predetermined optimal crystal growth rate and the evaluation criteria is based on how closely the system follows the growth rate trajectory (tracking error) as well as how much variation there will be in the tracking error when several batches are run. Large variability in the process behavior due to its stochastic nature is ought to be counteracted by the application of our model-based control strategy.

¹ G_{r_i} is the crystals growth rate

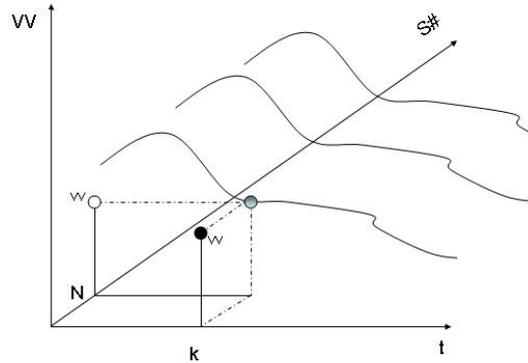


Figure 5-2: data sets used for performance evaluation

Process evaluation

It is assumed that by maintaining the crystal growth rate constrained to a maximum value of $2.5 \cdot 10^{-8} m/s$ (reference trajectory), the quality specifications of the product are met. We will also achieve a mean crystal size of around $500 \mu m$, and a crystal content of roughly 30%.

Maintaining the crystal growth rate close to the reference trajectory value is important when it comes to fulfilling quality specifications of the product. If the crystal growth rate upper limit is exceeded, then unwanted phenomena will occur in the process, for instance undesired agglomeration and nucleation will take place. This will lead to non-acceptable quality defects such as mother liquor inclusion, impurity inclusion and impurity uptake. Low quality product means lower process efficiency and therefore lower profits.

Therefore the evaluation of the control strategy from a process point of view is mainly determined by the ability to maintain the plant's crystal growth rate as close as its upper limit as possible, and keep it preferably smaller than the upper value.

The production yield is obviously determined by the crystal content. But it is second in importance.

5-3 Open-loop analysis

The main task of the observer in the open-loop scenario is to deliver accurate estimates of the system state variables. Plant variables estimates can be used for process monitoring, as each have a physical interpretation in terms of the crystals, thereby their importance. Coming up we will present some plots showing the evolution of the estimation error along a batch run, the simulation starts at time $t_0 = 0$ and finishes at $t_f = 10800 s$. For instance, Figure 5-3 shows the NRMSE of the estimates of the plant variables given by each observer. Even though the estimation error is zero at the beginning of the plot as both the observer and plant have the same initial conditions, the next time the observer gives an estimate there is already an estimation error. We also comment that fact the error becomes smaller as time goes by, meaning the estimates are converging to actual values of the variables. Figures 5-3 to 5-7 present the NRMSE for every single state variable. Therefore, the only difference between

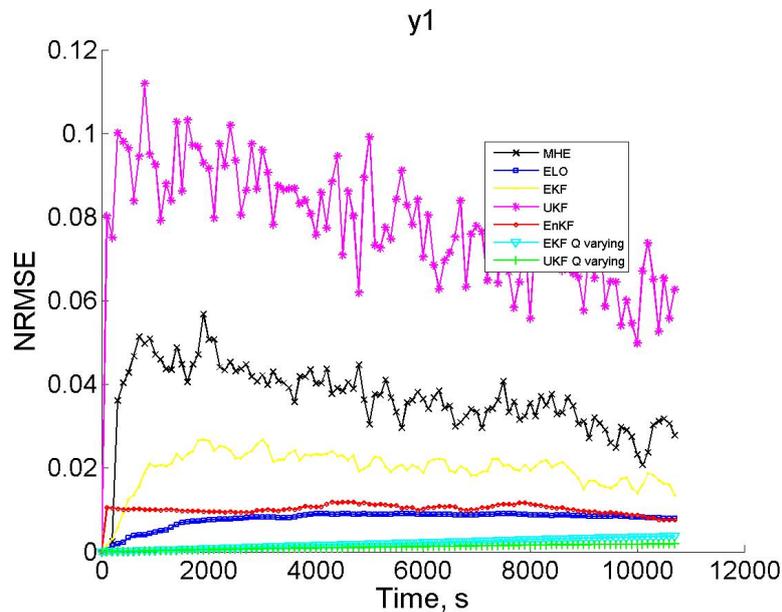


Figure 5-3: NRMSE y_1 along time, open-loop case.

each plot is the fact that they belong to different variables, but they all describe the same feature, the evolution of the estimation error along a batch run.

In general, all the stochastic observers performed better than the ELO and even better than MHE, this was expected due to the presence of stochastic measurement noise corrupting the information we acquire from the system. In Appendix A we include the graphs when no stochastic measurement error is used, the performance of all the stochastic observers is hardly degraded, but an improvement is noticeable in the ELO and MHE.

Finally, the implementations of the EKF and UKF with time-varying Q matrix perform better overall. Values of the covariance time-varying Q matrix according to the proposed algorithm indicate that uncertainty is low due to parameters variations, therefore when this algorithm is used we highly rely on the prediction capabilities of plant model, consequently we observe low variation in the trajectories of the estimated variables. On the other hand, ELO and MHE show a worse estimation error since the noise present in the system is not filtered out. MHE, however performed better than ELO.

The results concerning the CPU time required by every observer are shown in Figures 5-8 and 5-9.

Computational Demand

Figure 5-8 shows the time required by each observer every time a measurement is available. The values are consistent through the whole simulation, and no large peaks, meaning the algorithm took too long to provide an estimate at a certain instant, are present. The interval between measurements is 100s, then from the values given in the graphs, we realize the observer leaves enough margin for other tasks, such as calculating input signals, which is important in the case of a real-time implementation where time is a hard constraint.

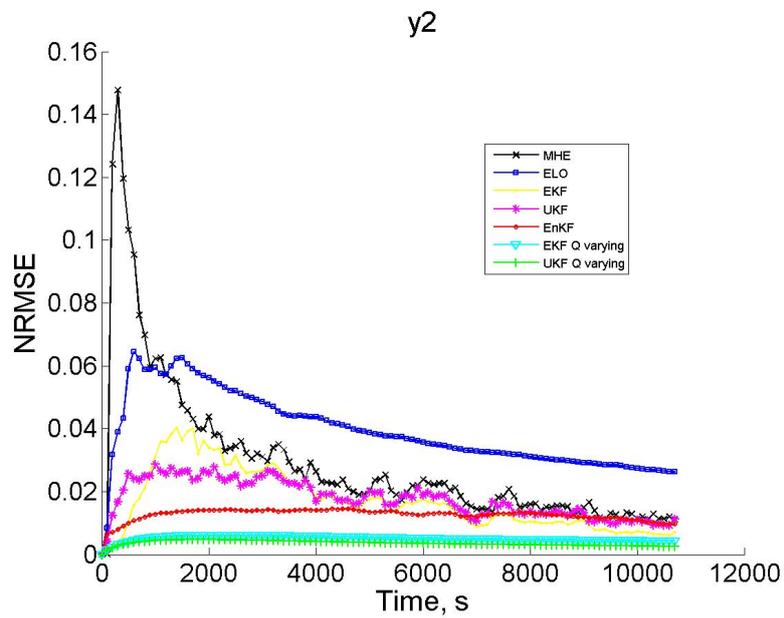


Figure 5-4: NRMSE y_2 along time, open-loop case.

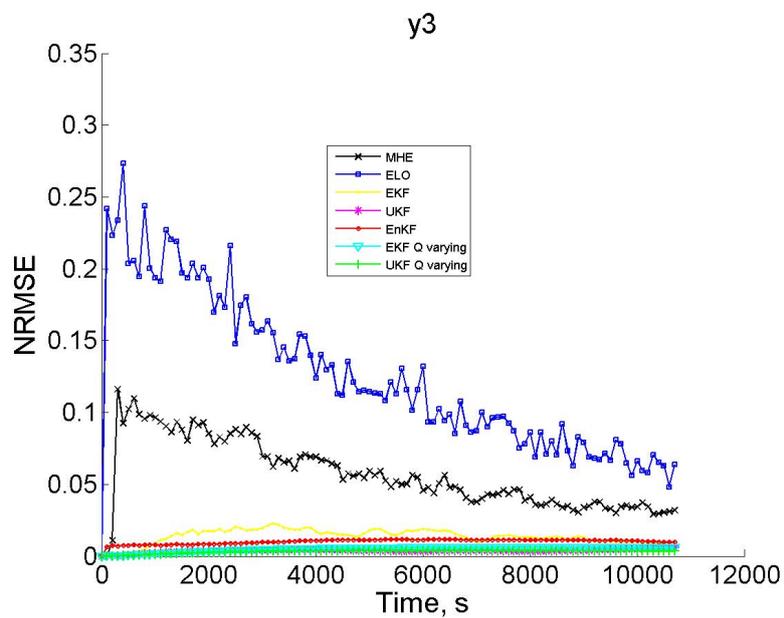


Figure 5-5: NRMSE y_3 along time, open-loop case.

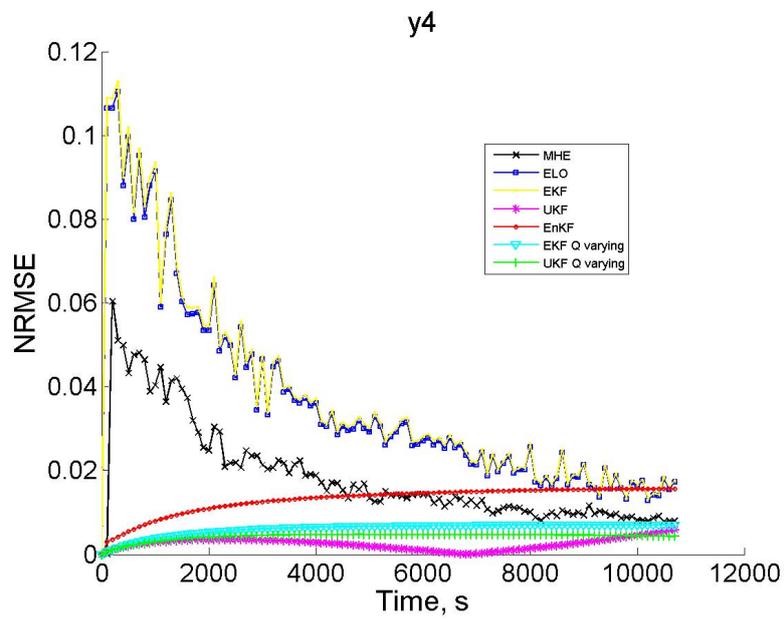


Figure 5-6: NRMSE y_4 along time, open-loop case.

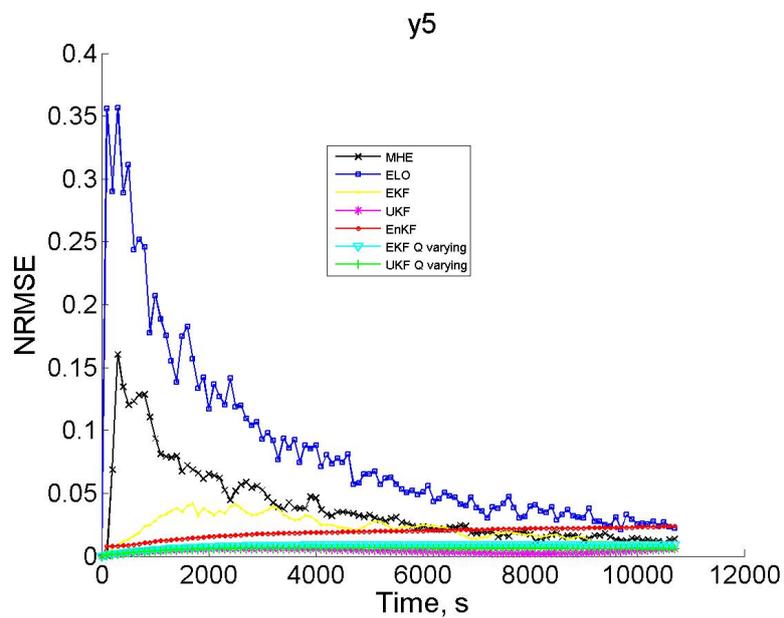


Figure 5-7: NRMSE y_5 along time, open-loop case.

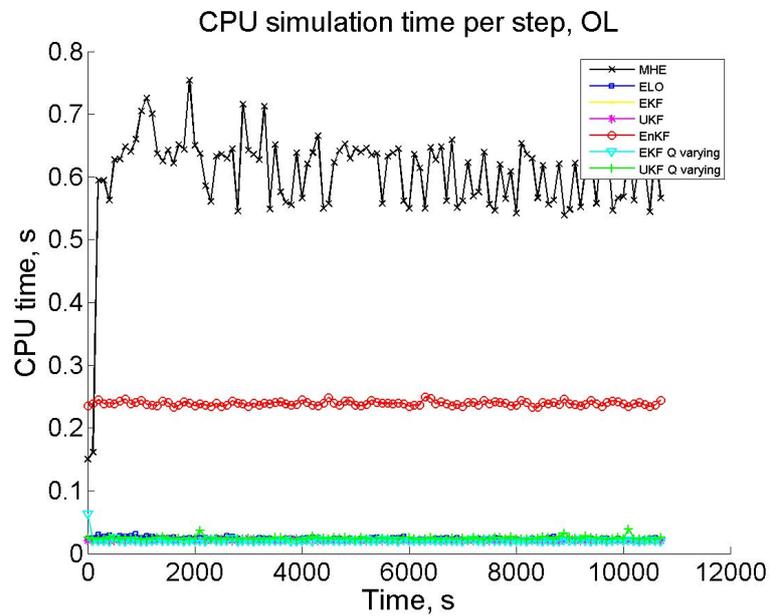


Figure 5-8: CPU time required by each observer at each iteration, open-loop case

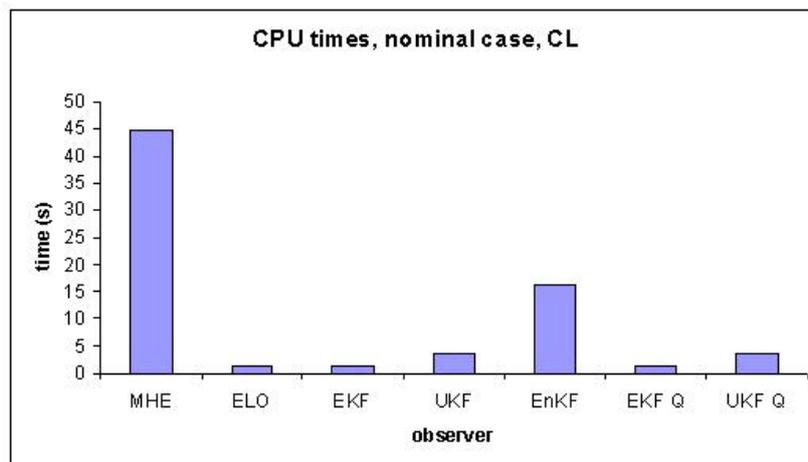


Figure 5-9: CPU time required by each observer during the simulation of one batch, open-loop case

Figure 5-9 shows the total time consumed by the observer during the simulation of one batch. EKF, UKF and ELO rank as the least time consuming observers, while MHE and EnKF are the slowest ones. In the case of UKF and EKF when a time-varying Q matrix is used, it slightly increases computational time, however, the overall performance of the control strategy is largely improved, at least in terms of estimation accuracy.

5-4 Closed-loop analysis

This section is divided in two parts, first we analyze the nominal case and then we move onto the case where all the error sources were applied simultaneously.

The tuning of the observers is kept constant with respect to the open-loop case, otherwise it will be explicitly said. For instance, ELO was the only observer that was retuned for the closed-loop nominal case.

Closed-loop nominal case

All the graphs showing the estimation errors for the nominal case in closed-loop configuration can be found in Appendix B. They are not reproduced here because they do not present more relevant features than those highlighted in the open-loop scenario. That is, we observe convergence to zero in the estimation errors; the performance of the observers remains unchanged, being the deterministic observers those with poorest behavior, while EKF and UKF seem to be the best suited for the task.

In the next section we will concentrate on the control performance analysis, there we will discuss the control performance for the closed-loop configuration.

Before going further, we take on how the ensemble size, in the case of EnKF, and the estimation horizon, concerning the MHE, influence their performance.

Effects of the ensemble size

We have four cases,

- ensemble size of 10 members,
- ensemble size of 20 members,
- ensemble size of 40 members,
- ensemble size of 80 members.

Generating the ensemble members was made by sampling numbers from a normal distribution. Selecting this distribution was rather arbitrary as we did not have access to prior information which would give us a clue on how the probability distribution of each state variable looked like, but we knew the noise perturbing the measurements was from a Gaussian distribution. In fact determining the covariance matrices of the system and measurement noise is what complicates tuning of any filter, otherwise smarter options could be made from the beginning.

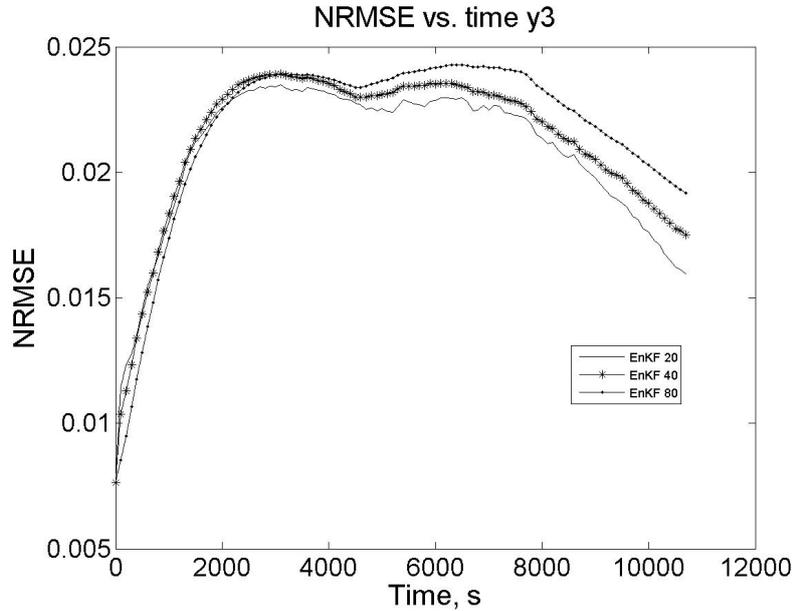


Figure 5-10: EnKF NRMSE y_3 , at every instant for different ensemble sizes

To generate the numbers from a normal distribution it is necessary to specify the mean and the variance of the distribution, in our case, the mean must have been equaled to the desired initial value of the state variables. The variance is up to the person tuning the observer. In the present work the variance was chosen close to that given by the Q matrix of the EKF. As already explained in previous sections, this matrix represents how much the estimates given by the observer model are trusted, as the values chosen worked for the EKF, it was expected the required values for another filter to work were similar. Eventually, the values had to be adjusted for the EnKF to deliver accurate results in the order of the other filters. This is a straight consequence as the EKF and EnKF algorithms are not equivalent.

In case there were no access to this information, that is the Q matrix of the EKF, the values might be estimated by analyzing available experimental data, if any, otherwise, tuning would need to be done by trial and error.

We can see in Figure 5-10 that an ensemble size $N = 20$ members leads to a worse estimation accuracy than for $N = 20$. In Figure 5-11 we see that augmenting the ensemble size proportionally increases the computational time. Furthermore, increasing N does not improve the accuracy of the filter in the same proportion it increases the computational time, so in our case, it is not worth to further increase the ensemble size to more than 20 members.

The position in state-space of the ensemble members is related to the probability distribution the members are sampled from. This can be explained in the context of our example. As the members are sampled from a normal distribution, it is very likely to sample values which are close to the mean of the distribution, especially if the variance is small, somehow. Increasing the number of members will definitely increase the chances to sample values away from the mean, but the major ensemble members will be not far from one another.

At the end of the day, the location of the ensemble members determines how well the statistics from the state variables are captured. Because of this, increasing the ensemble size from a dis-

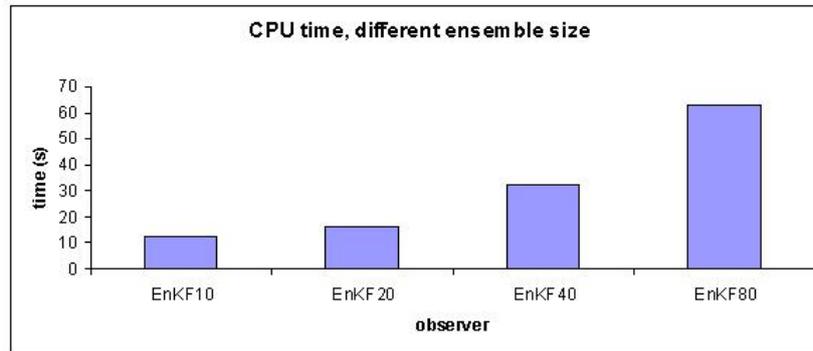


Figure 5-11: CPU time for different ensemble sizes

tribution with the same parameters will not lead to large improvements in the latter task once a threshold has been reached. Different sampling techniques exist and could be used to further improve the filter performance as the ensemble size grows. In our case, the aforementioned threshold seems to be an ensemble size of 20.

Effects of the estimation horizon

The MHE estimation horizon values were set to 300, 500 and 700 s.

The estimation horizon as described in Chapter 2 allows to increase/decrease (if the estimation horizon is increased/decreased) the amount of information this observer uses back in time to generate the variables' estimates. Therefore, if more information about the system is available, in terms of measurements, it is expected the observer to generate better estimates, because the observer should “better understand” the behavior of the system. This principle is applied in the smoothers. Predictions are better when the estimation horizon is increased since the NRMSE values decrease. This is shown in Figure 5-12.

Nevertheless, this performance improvement comes to the price of an increased computational time, which in general is not desirable. We present computation times for the MHE in Figure 5-13.

Control performance

Crystal growth rate is what we aim to control in this case study. This quantity must be kept as close as possible to a prestablished value of $2.5 \cdot 10^{-8} m/s$ as long as possible during the batch. Fluctuations may exist but those above the target value are not desirable due to quality issues introduced in Chapter 4.

After comparing the growth rate data with the estimation error of all the state variables we realized that control of the growth rate is closely related to variables y_3 and y_4 , representing second and third moment respectively, but mainly to y_3 . This may not be a surprise as the second moment is related to the total surface of crystals in the system, which in turn favors crystal growth. Besides, concentration, which is the driving force for crystal growth, is mainly

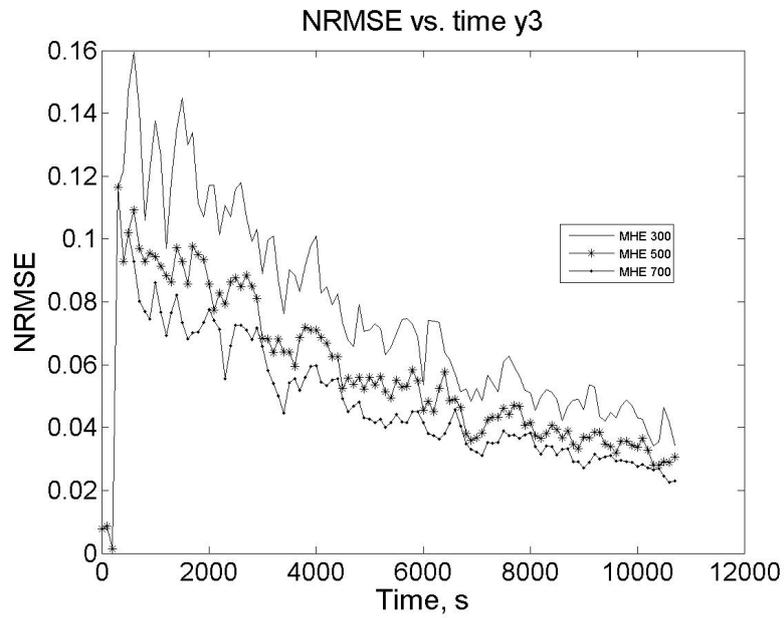


Figure 5-12: NRMSE y_3 different estimation horizons

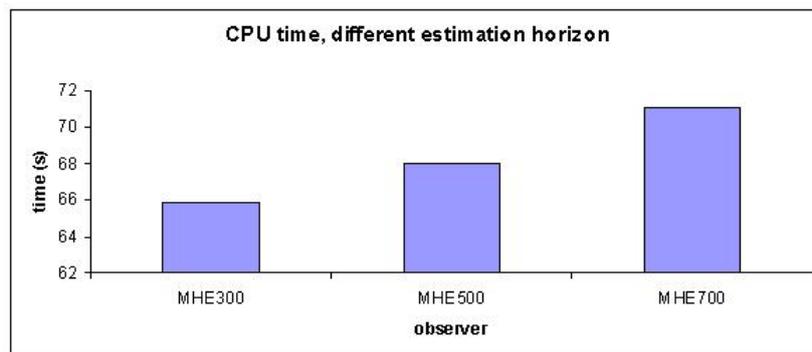


Figure 5-13: CPU time for different estimation horizons

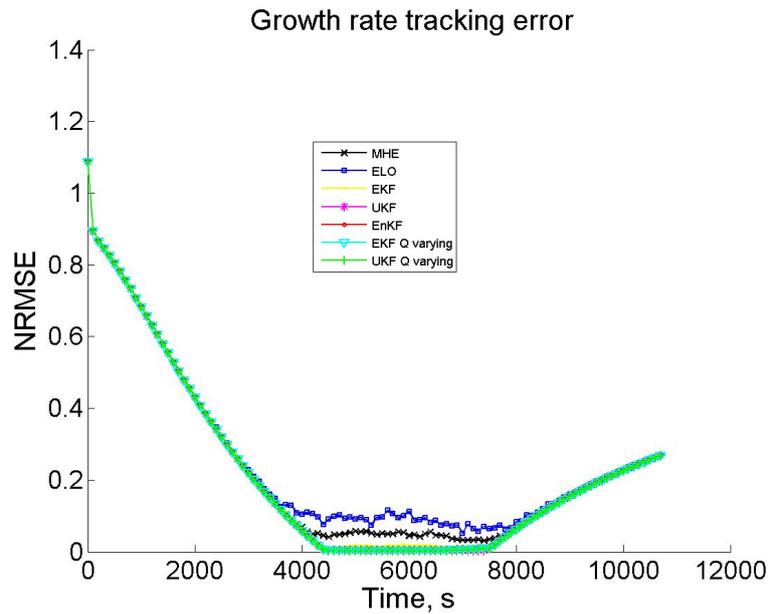


Figure 5-14: NRMSE crystal growth rate, at every instant, nominal case

Table 5-1: Observers listed in ascendant order by the magnitude of their tracking error

UKF Q-varying
EKF Q-varying
UKF
EKF
EnKF
MHE
ELO

related to the second moment. Therefore, good estimation of system's second moment is of high importance in order to achieve good control performance.

Crystal's growth rate tracking error through a batch run is presented in Figure 5-14. There we have plot the tracking NRMSE for every observer versus time. We also show in Figure 5-15 a close-up of the growth rate curves in the interval from roughly 4000 to 8000 s. We show this interval since it is there where we can exert control action over the crystal growth, this due to actuation limitations. We see in these figures that ELO and MHE present the largest offset with respect to the reference trajectory. Indication that noise disturbs the control scheme performance when a deterministic observer is used.

If the tracking error is quantified over the time interval where we aim to control the growth rate, and the observers are listed such that the first in the list presents the smallest tracking error, we obtain Table 5-1

The same can be done with respect to the estimation error of all variables; however, it turns

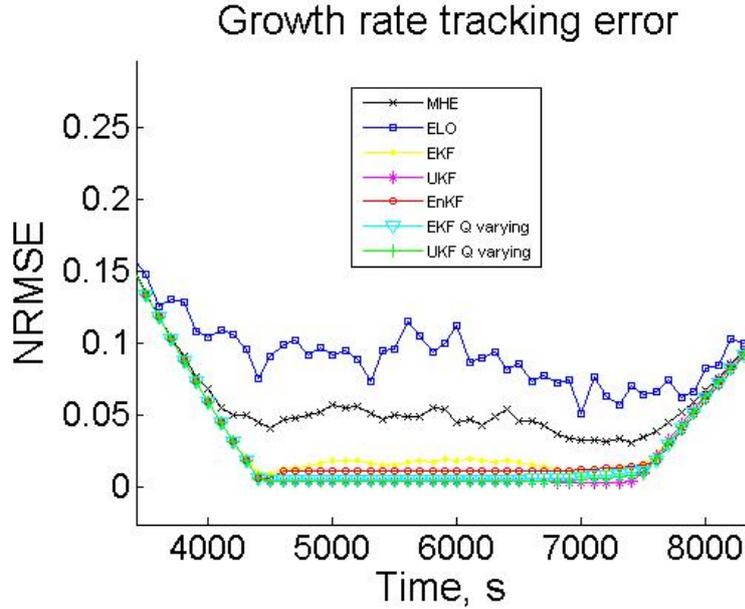


Figure 5-15: zoomed-in NRMSE crystal growth rate, at every instant

out that the ranks for y_3 and y_4 , which as already discussed represent the total crystals surface and volume respectively, list the observers in exactly the same order that when listed to respect to the tracking error. This brings to light a relationship between the control performance and the observer performance. We may now say that better estimation accuracy will lead to better tracking.

The link between the estimation accuracy and the tracking error can be traced back to the model of the plant. The equation to calculate the growth rate is given by,

$$G_r = k_g(C - C^*) \quad (5-3)$$

furthermore, according to (5-4) the solute concentration C is a function of y_3 and y_4 ,

$$\frac{dC}{dt} = \frac{Q_p \frac{C - C^*}{V} + 3k_v k_g (C - C^*)^g y_3 (k_1 + C) + k_2 H_{in}}{1 - k_v y_4} \quad (5-4)$$

from (5-3) and (5-4) we see the Growth rate directly depends on y_3 and y_4 , so as long as we estimate these variables adequately we will be able to correctly drive the system to obtain the desired crystal growth rate. Somebody may argue that even if the growth rate depends on y_3 and y_4 they still depend on y_1 , y_2 , etc., and might state that all the variables should be accurately estimated. Even if that is intuitively appealing, it is not necessarily true, as the relations linking one variable to the others may combine to generate an estimate of a variable with a smaller estimation error than that found in the variables building the relationship. In fact this is the case in our system and it will be shown in the next section.

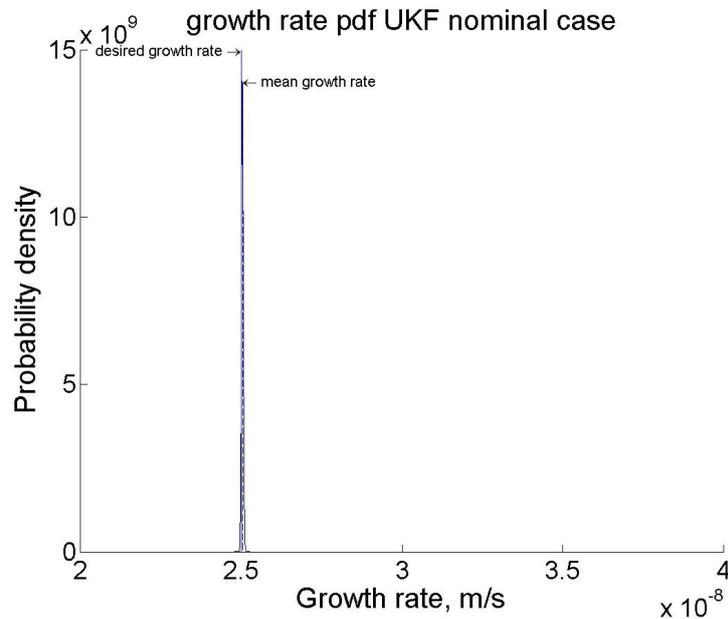


Figure 5-16: expected distribution of the crystal growth rate when the UKF is used

Process evaluation

Process evaluation takes on how close the plant can follow the growth rate reference trajectory. Therefore, we calculate the mean value and variance, over 50 simulations, of the crystal growth rate achieved when different observers are used in the process. These quantities will tell us what growth rate we can expect and how much it will vary around this value. Furthermore, we plot this quantities as a normal probability density function together with the desired crystal growth rate, Figure 5-16 shows the growth rate reference value, the mean value achieved by the plant and how growth rate values spread over the simulations. The observer used in Figure 5-16 is UKF. We showed only this observer as it performed better than the rest. The mean value of the growth rate is very close to the one desired, and the process will closely stick around as the shape of the curve is very narrow. Then we can expect good quality properties of the product from those batches.

We also show in Figures 5-17, 5-18 that production's mean crystal size and production's crystal content also stayed close to both, the desired mean crystal size and the crystal content of the batch. Finally, for the nominal case we can say that, both quality and production specifications were met.

Closed-loop, general case

We start off by making clear that observers were tuned for the open-loop scenario. For the closed-loop nominal case no changes were made, only the ELO had to be retuned in order to achieve reasonable estimation accuracy. For the closed-loop general case, the latest tuning for the ELO was not changed, neither was changed for the rest of the observers, except for the EnKF and the Q-varying matrix versions of the EKF and UKF. This had to be done because introduction of model mismatch caused the latter filters to diverge.

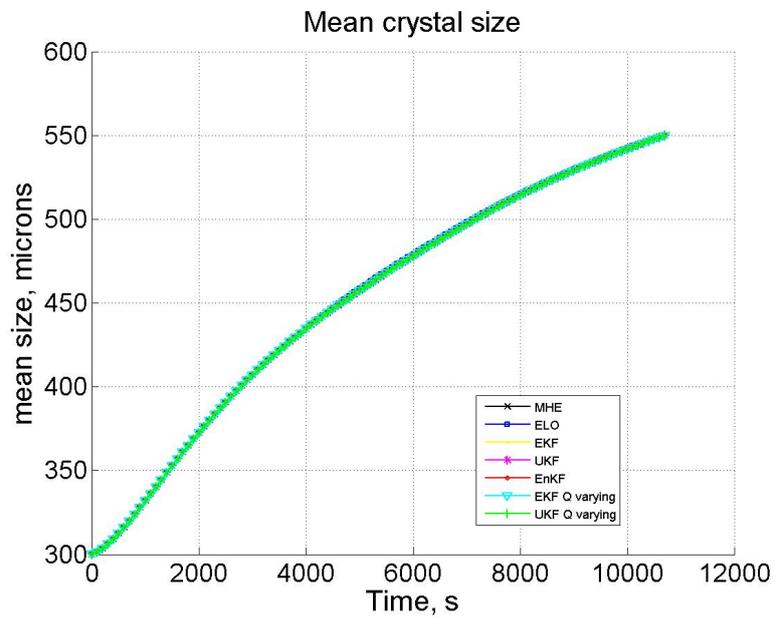


Figure 5-17: mean crystal size, nominal case.

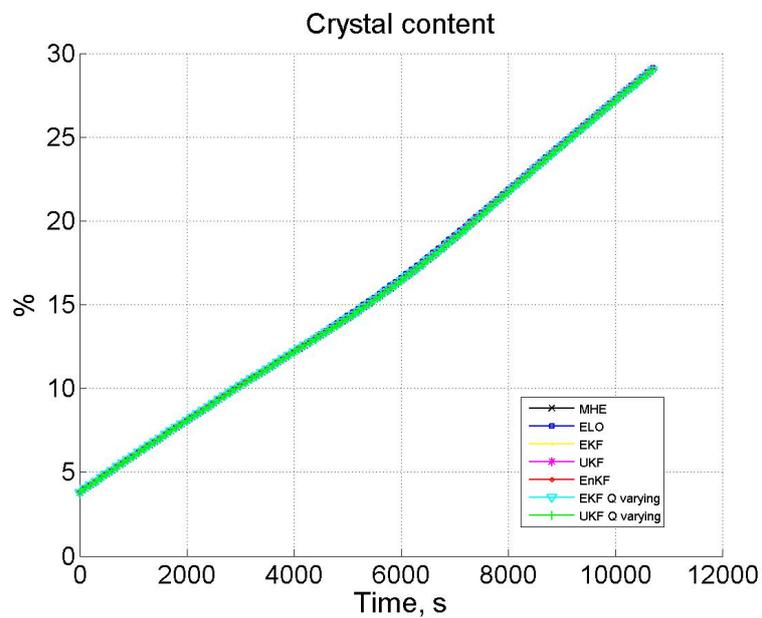


Figure 5-18: crystal content, nominal case.

Table 5-2: Initial conditions for the plant, observer and controller, general case

Variable	Plant	Observer & controller
m_0	$1.08 \cdot 10^{10}$	$1.13 \cdot 10^{10}$
m_1	$3.42 \cdot 10^{05}$	$3.59 \cdot 10^{05}$
m_2	$3.69 \cdot 10^{02}$	$3.87 \cdot 10^{02}$
m_3	$8.30 \cdot 10^{-02}$	$8.72 \cdot 10^{-02}$
m_4	$2.49 \cdot 10^{-05}$	$2.61 \cdot 10^{-05}$
C	$4.57 \cdot 10^{-01}$	$4.66 \cdot 10^{-01}$

If we look back on Figures 5-3 to 5-7 we see that EnKF, and the Q-varying matrix versions of EKF and UKF achieve a smaller estimation error than the rest of the observers. As mentioned before, in order to filter out the stochastic measurement noise, which was the only error source in the nominal case, the observers must make a trade-off between the importance given to the predictions delivered by their process model, and the measurements. If the noise in the measurements is too high, then it is preferable to attach more weight to the model. If the model can faithfully reproduce the dynamics of the system to control, then we are on the safe side and we will move in the right direction, that is, we will be able to make good estimates of the variables. We say the latter observers were heavily weighting the predictions done by the model, that is why their predictions look so smooth, besides, their estimation error is very small because the predictions of their model fit quite well the values from the reality. However, if the model and the real counterpart exhibit different output behavior, we will not be able to compensate for those differences since we are barely weighting the information obtained from the real system, and therefore losing information. As this loss of information builds up the observer will diverge. Model mismatch will, in general, cause different output behavior between the real plant and the observer model, and therefore the observers that heavily weight the predictions from their model might show divergence. That was the case for EnKF and the Q-varying matrix versions of EKF and UKF.

Now we can proceed to analyze Figures 5-19 to 5-23 which correspond to the general case where all the error sources are introduced into the model based control system. All figures start at a point different from zero, this is due to the presence of uncertain initial conditions. The initial conditions in the observer-controller pair were perturbed, with +5% for the first five variables, namely the statistical moments of the crystal size distribution, and +2% for the sixth variable, corresponding to the concentration, with respect to the values used in the process.

Table 5-2 shows the values of the initial conditions for both the observer-controller pair and the plant.

The fact that the plant and the observer start at different points enlarges the initial estimation errors with respect to the nominal case. However, since the plant is observable, the observer is able to compensate for the initial estimation errors. It is not possible to determine the speed of convergence of the observer, as it would be the case for a linear system, but the observability of the system allows for the initial larger errors to vanish and converge to comparable values like in the nominal case as the system evolves in time. Our statement can be verified by the

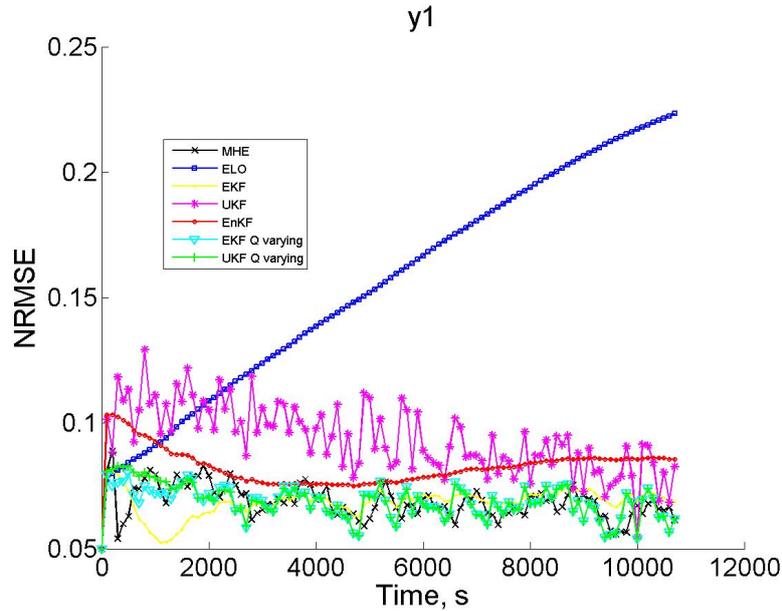


Figure 5-19: closed-loop general case, NRMSE y_1 , at every instant

plots in Appendix C, where the effects of solely uncertain initial conditions as error source are shown.

Yet we argue the observer is capable of compensating by the error of uncertain initial conditions, we see in the plots for the general case that the estimation error seems to stabilize around a value larger than in the nominal case. We explain this difference by the presence of systematic measurement error. We again refer to the fact that model predictions and measurements have to be weighted in order to deliver a variable estimate. If we do not count with extra information, such that in no way we can realize that our measurements are biased from the actual values, then we will consider the measured values as our reality, and we will try to get as close to them as possible. As we cannot tell the difference between reality and that obscured by our measurements we will not be able to compensate for that error. In fact, if somehow we retrieve the true information from the system and compare it to our predictions we will find an offset between them. This explains the offset in our results. We do have access to the real data once we conclude a simulation and we see that without extra information about the systematic measurement error the observers are not able to compensate for it. Furthermore, we would need to augment the state of the system if we wanted to compensate for systematic measurement error, or implement integrating action in the controller to eliminate any offset. Results in Appendix D support the former argument, and the latter finds support in Appendix F.

Control performance

The dependence between estimation accuracy and control performance found in the open-loop case also holds in our most general case. As the estimation error for y_3 and y_4 is larger than for the nominal case so is the tracking error. This claim is presented in Figure 5-24. We are

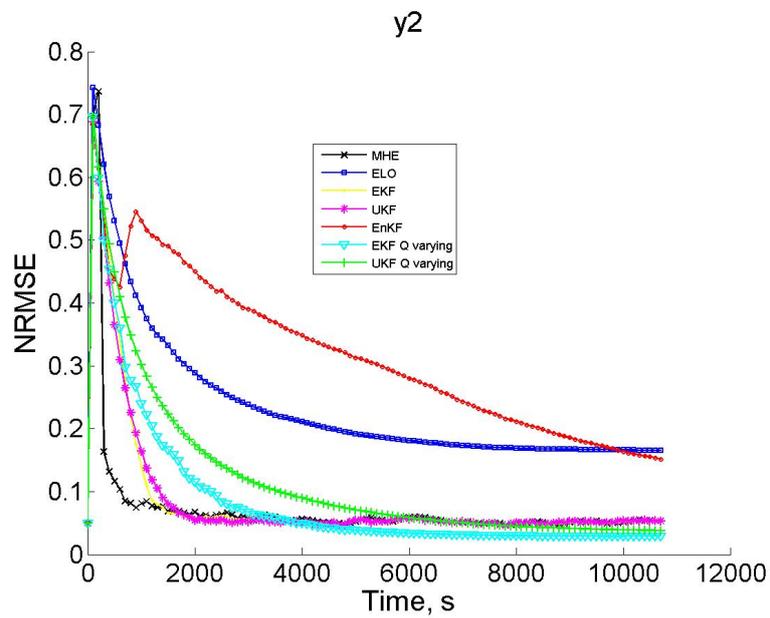


Figure 5-20: closed-loop general case, NRMSE y_2 , at every instant

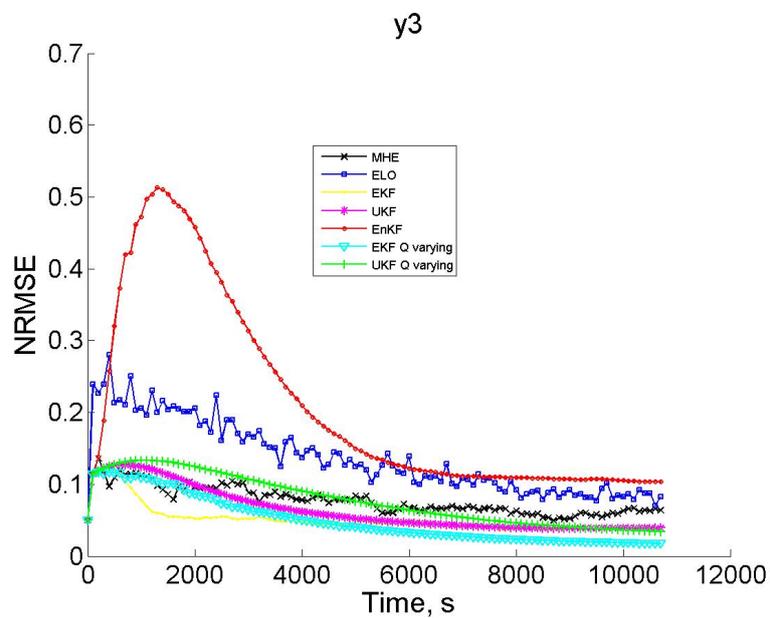


Figure 5-21: closed-loop general case, NRMSE y_3 , at every instant

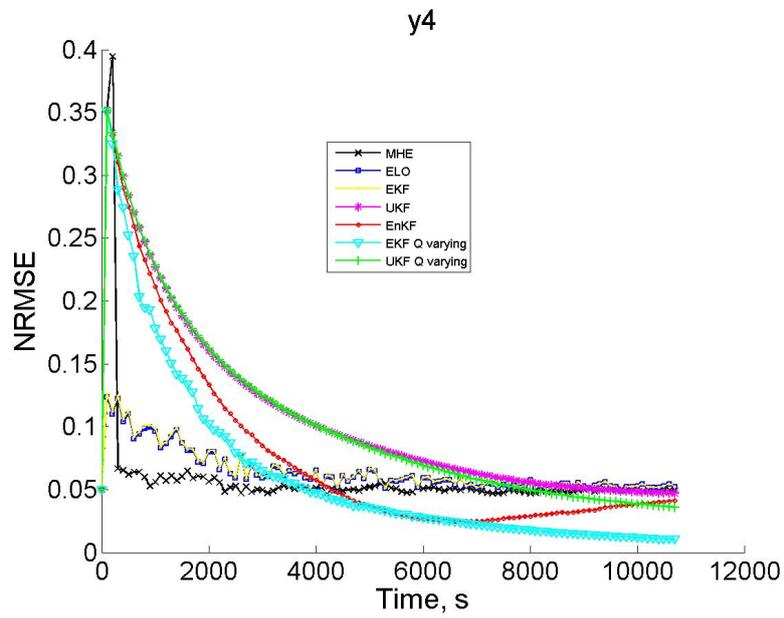


Figure 5-22: closed-loop general case, NRMSE y_4 , at every instant

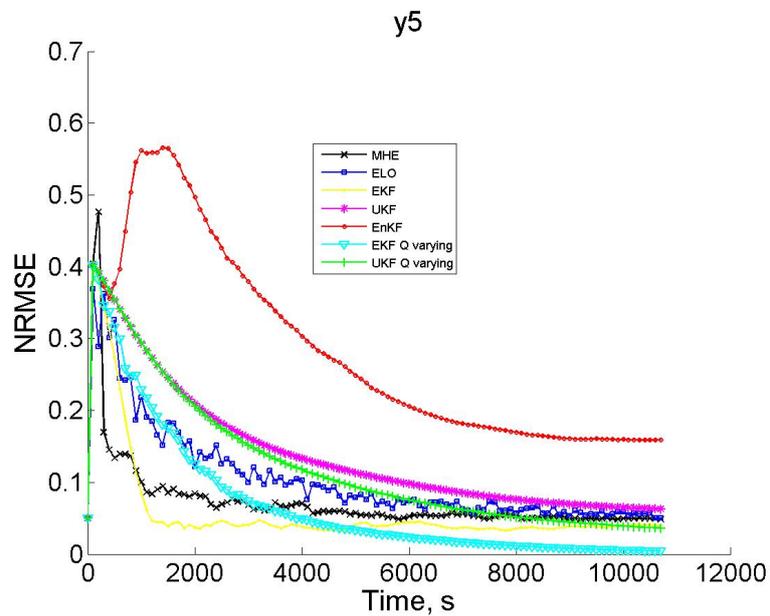


Figure 5-23: closed-loop general case, NRMSE y_5 , at every instant

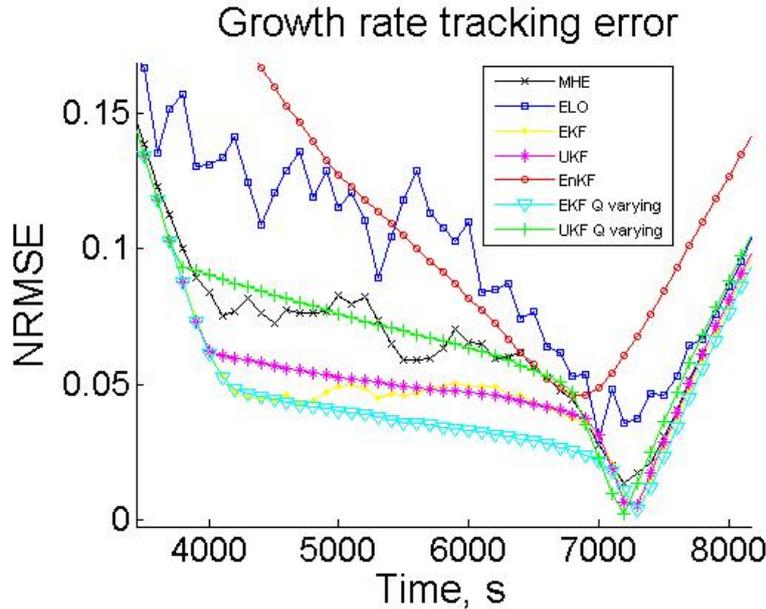


Figure 5-24: NRMSE crystal growth rate, at every instant, closed-loop general case.

only interested in the interval from about 4000 to 8000 s since actuation limitations are not an issue therein. When Figures 5-15 and 5-24 are compared we see the latter is shifted with respect to zero in the vertical axis (indicating the NRMSE), this is the result of introducing errors in the system, while in Figure 5-15, growth rate in the nominal case, the bias is smaller.

Process evaluation

For the general case we plotted how the tracking error mean value is distributed along 50 simulations per observer. All the plots corresponding to each observer can be found in Appendix G. Here we only show the plot for three cases, UKF in Figure 5-25 which presented the smallest tracking error, ELO in Figure 5-26 and EnKF in Figure 5-27, with the worst performance. Results for ELO were expected as it is a deterministic observer, but not for EnKF. The bad performance achieved by EnKF can be improved by retuning the observer, though. Coming back to the graphs, for EnKF and ELO we see the average value of the mean growth rate is above the upper limit, then we can expect product not meeting the quality specifications. However, there is no way to determine how low the quality will be, unless the production is inspected. Even more, as the mean value for the crystal growth rate is larger than $2.5 \cdot 10^{-8} m/s$, we expect a larger mean crystal size, and also a larger product yield at the end of the batch than in the nominal case. Figures 5-28 and 5-29 show the prediction is true. However, even if the productivity has been increased, we must highlight that it has been achieved by reducing the quality of the product, so this is not an acceptable trade-off. Quality must be at the top of the requirements to be fulfilled by the process. More clearly, we show the case of EnKF, we compare the crystal content in Figure 5-30 and the mean crystal growth in Figure 5-31. Finally, we compare in Figure 5-32 how the presence of all the error sources shift the expected value of the growth rate with respect to the nominal case. The mean value is shifted to a higher value, this is not desirable since it will degrade product

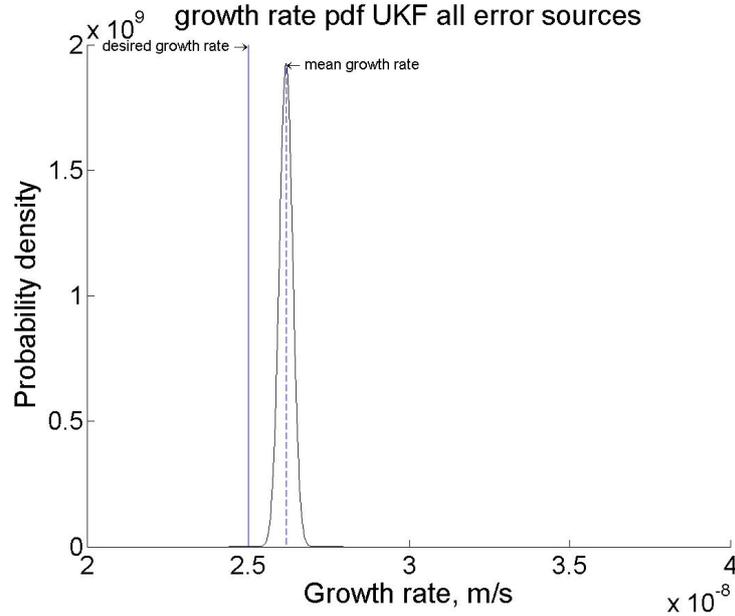


Figure 5-25: expected distribution of the crystal growth rate when the UKF is used.

quality. All the errors also increase variability in the process, that is, values of state variables and properties of the product will vary over a larger range giving rise to more irreproducible batches. Both consequences are undesirable.

5-5 Disturbance suppression

The presence of a disturbance will drive the system away from the desired growth rate set point. Once transient effects vanish, input or output disturbances will create a similar effect on the process, as they will drift the system away from its desired path. Therefore, from a practical point of view in terms of effects, the presence of disturbances on either side of the system does not really make a difference. We will however, perturb the system on the output to perform a disturbance rejection analysis. The immediate consequence is that it will lead to erroneous measurements and therefore erroneous information feeding our control strategy. We can include an output disturbance in the mathematical model of the system as follows,

$$y = h(x, u) + d \quad (5-5)$$

where y is the output vector of the system, x and u are the states and inputs to the system, respectively, and d is the disturbance.

For our case study the perturbation appears 3000 s after the batch was initialized and its value is proportional to that of the corresponding output where it is acting, so we have,

$$d = ky. \quad (5-6)$$

We now take on the task of suppressing the effects of such disturbance. To do so, we will make use of a technique where we try to estimate the value of the disturbance.

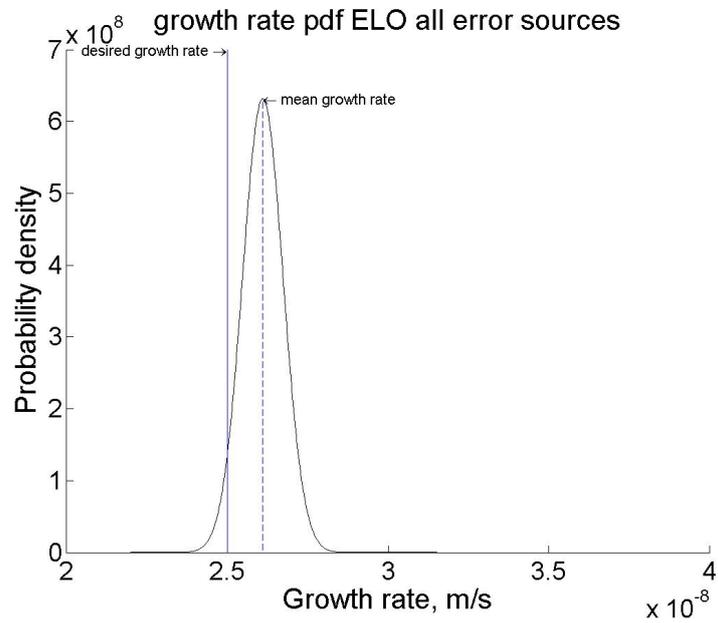


Figure 5-26: expected distribution of the crystal growth rate when the ELO is used

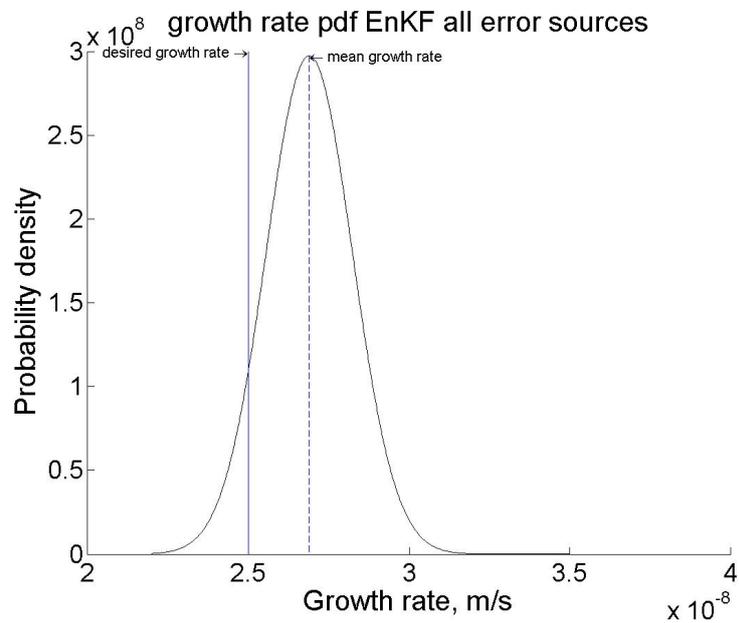


Figure 5-27: expected distribution of the crystal growth rate when the EnKF is used

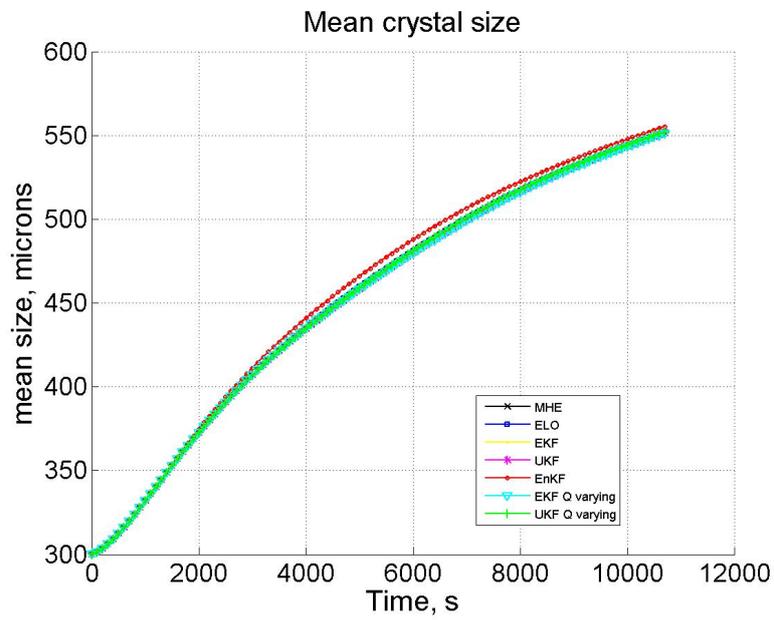


Figure 5-28: mean crystal size, general case

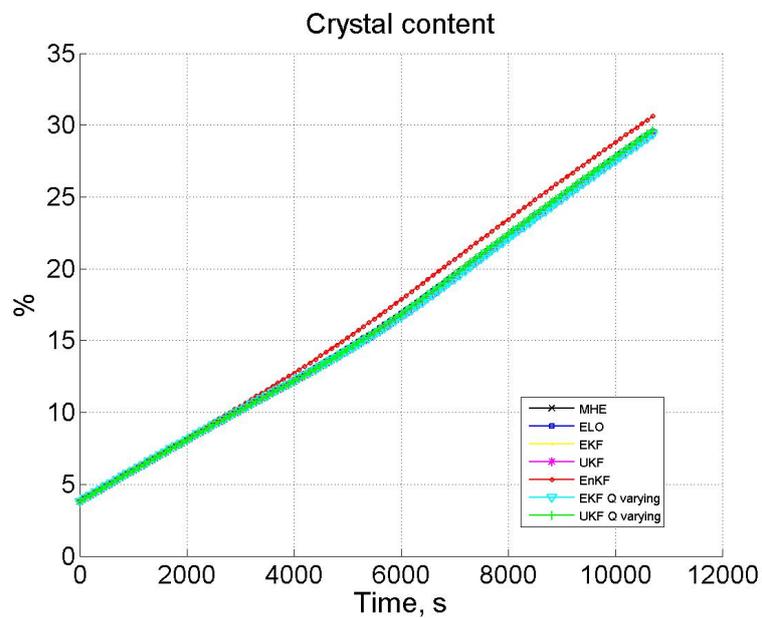


Figure 5-29: crystal content, general case

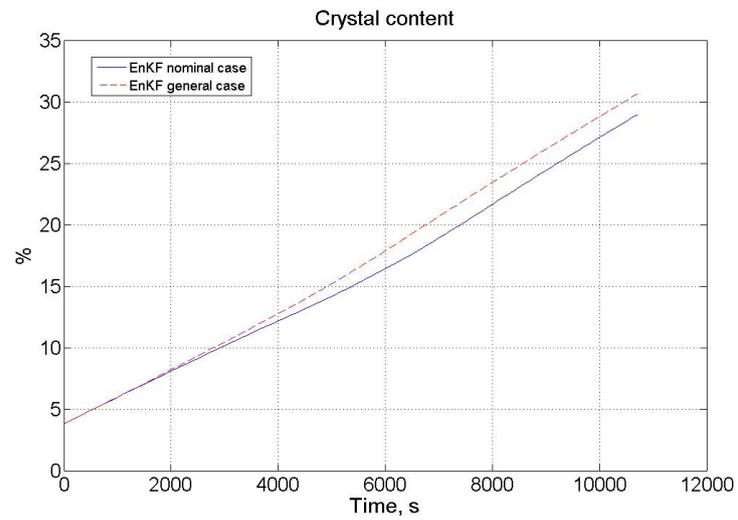


Figure 5-30: crystal content, general case vs. nominal case when EnKF is employed.

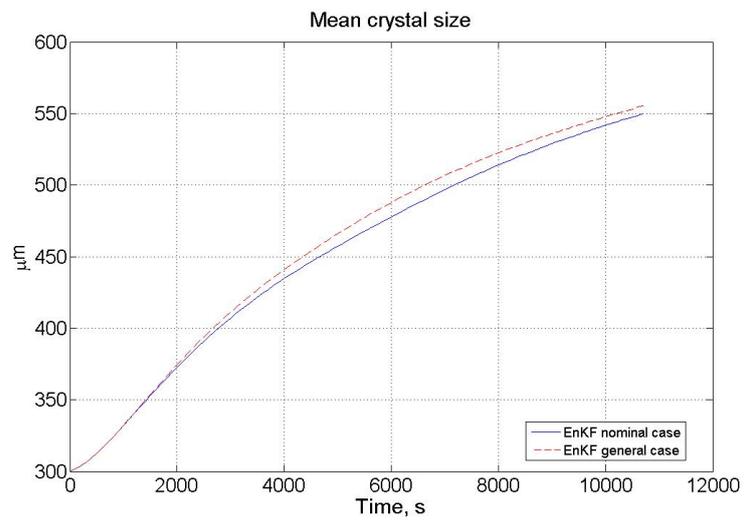


Figure 5-31: mean crystal size, general case vs. nominal case when EnKF is employed.

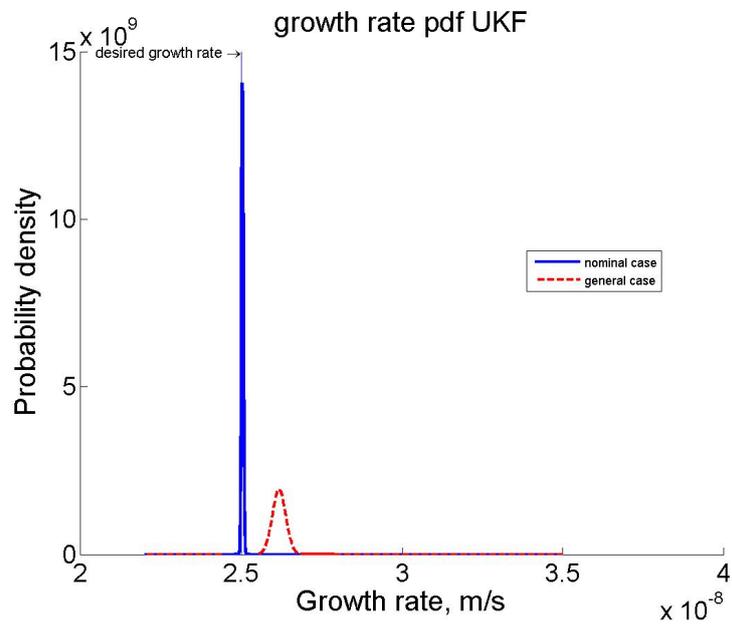


Figure 5-32: expected distribution of the crystal growth rate when the UKF is used

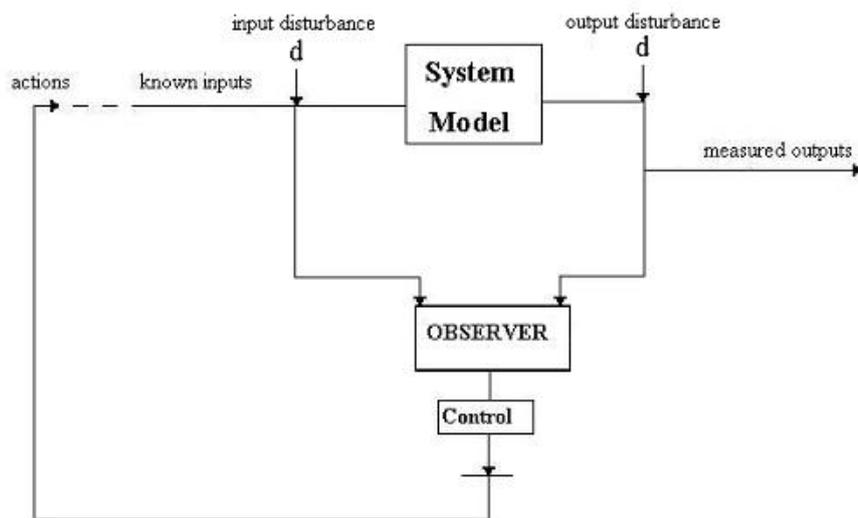


Figure 5-33: disturbances perturbing the system at input and output.

In order to estimate the disturbance we need to propose a model for it, that is, we try to describe the dynamics of the disturbance acting on the system, then we incorporate the disturbance model in the observer such that the observer will estimate the system states and the disturbance, simultaneously. The observer is then augmented with as many states as perturbations we want to estimate. We can write this as,

$$\begin{pmatrix} \hat{x} \\ \hat{d} \end{pmatrix} = \begin{pmatrix} f(\hat{x}, u) \\ f_d(\hat{d}) \end{pmatrix} \quad (5-7)$$

where f represents the system dynamics and f_d represents the disturbance dynamics. Adding f_d to the original system function is what causes the number of states to increase as we need to assign a state variable per disturbance. Both functions together give rise to a new composed function which is in turn used in the observer. Also, \hat{x} and \hat{d} are the estimates of the states and disturbance vectors respectively, and u the input vector. It must be clear that when a system is augmented, the augmented system must be observable as well, otherwise we would not be able to withdraw any information from the measurements in order to reconstruct the states.

Since in previous analyses we saw that improving the estimation accuracy of the second and third moments, that is x_3 and x_4 , leads to a better control performance we will mainly focus on suppressing the effect of the disturbance on x_3 . Therefore only one extra state will be added to the observer in order to account for the disturbance in x_3 .

We proceed by stating that when we measure x_3 in the system we obtain the value x_{3c} (subscript c stands for corrupted), thus we should be able to isolate x_3 out of the latter measurement. To do so, we see that x_{3c} is composed by,

- x_3 which corresponds to the state variable free from errors,
- d , the value of a non stochastic disturbance,
- n , a stochastic disturbance.

Therefore, we can express $x_{3c} = x_3 + d + n$.

The observer should allow us to,

- filter out noise,
- estimate d , and
- estimate x_3 .

Augmented system

In this case we propose a model that indicates that the disturbance has a constant value. We assign variable x_7 to the estimate of the disturbance. In discrete-time we have the disturbance dynamics given by

$$x_{7_{k+1}} = x_{7_k}. \quad (5-8)$$

Using the latter disturbance model we incorporate it to our system model consisting of equations (4-8) to (4-14) and apply our control strategy to the system.

The augmented system is given by the set of equations,

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \end{pmatrix} = \begin{pmatrix} B_0 - \frac{x_1 Q_p}{V} \\ Gx_1 - \frac{x_2 Q_p}{V} \\ 2Gx_2 - \frac{x_3 Q_p}{V} \\ 3Gx_3 - \frac{x_4 Q_p}{V} \\ 4Gx_4 - \frac{x_5 Q_p}{V} \\ \frac{Q_p(C^* - x_6) + 3K_v Gx_3(k_1 + C)}{1 - K_v x_4} + \frac{k_2 H_{in}}{1 - K_v x_4} \\ 0 \end{pmatrix}, \quad (5-9)$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} = \begin{pmatrix} x_1 + \nu_1 + d_1 \\ x_2 + \nu_2 + d_2 \\ x_3 + \nu_3 + d_3 \\ x_4 + \nu_4 + d_4 \\ x_5 + \nu_5 + d_5 \end{pmatrix}, \quad (5-10)$$

$$B_0 = k_b x_4 G \quad (5-11)$$

$$G = k_g (x_6 - C^*)^g. \quad (5-12)$$

All the variables are defined as in chapter 4, plus x_7 used to estimate the disturbance.

We again emphasize that we are able to estimate both the states and the disturbances because we have included a larger model in the observer which describes the dynamics of both types of variables.

We can now make use of the state and disturbance estimates in the controller in order to suppress the effects due to the perturbations.

Disturbance rejection results

To generate the results that follow we used EKF and UKF as observers. They were chosen as they presented good features throughout all our analysis presented in previous sections.

Figures 5-34 and 5-35 explicitly show the components perturbing the system. The erratic signal drawn with a thin solid line is the stochastic disturbance acting on the system, the dash-dotted line is the non-stochastic disturbance, while the broken line is the sum of the former and the latter signals. Finally, the thick solid line is the estimate given by the observer, when using the disturbance model given by (5-8), of the non stochastic disturbance. EKF and UKF were used in the control strategy in each figure, respectively.

We also plotted the second moment $m_2(x_3)$. For the perturbed system we plotted two cases, first when no action was taken to suppress the disturbance, and second, when a model of the

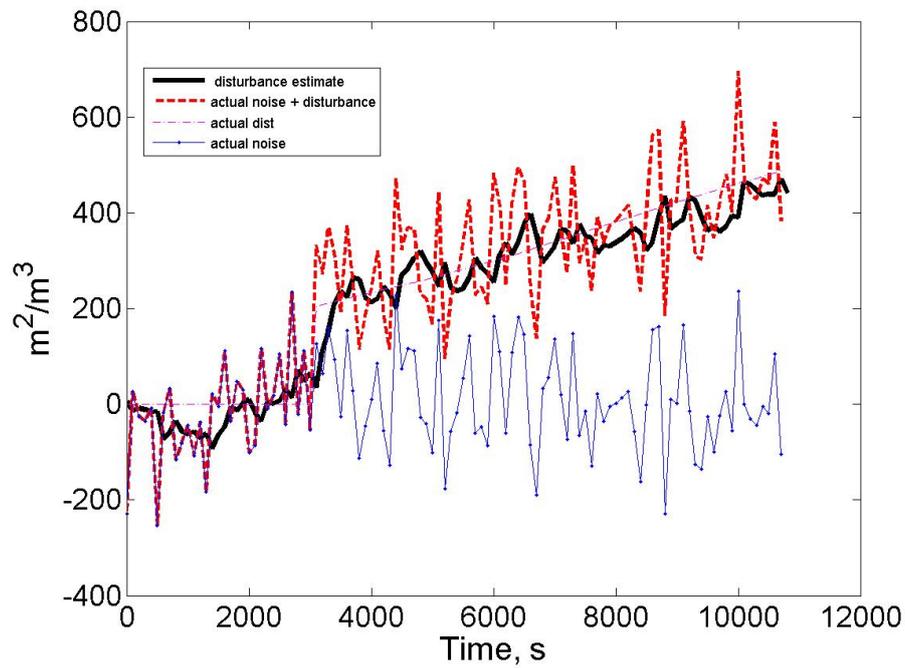


Figure 5-34: composition of the perturbation and its estimate using EKF.

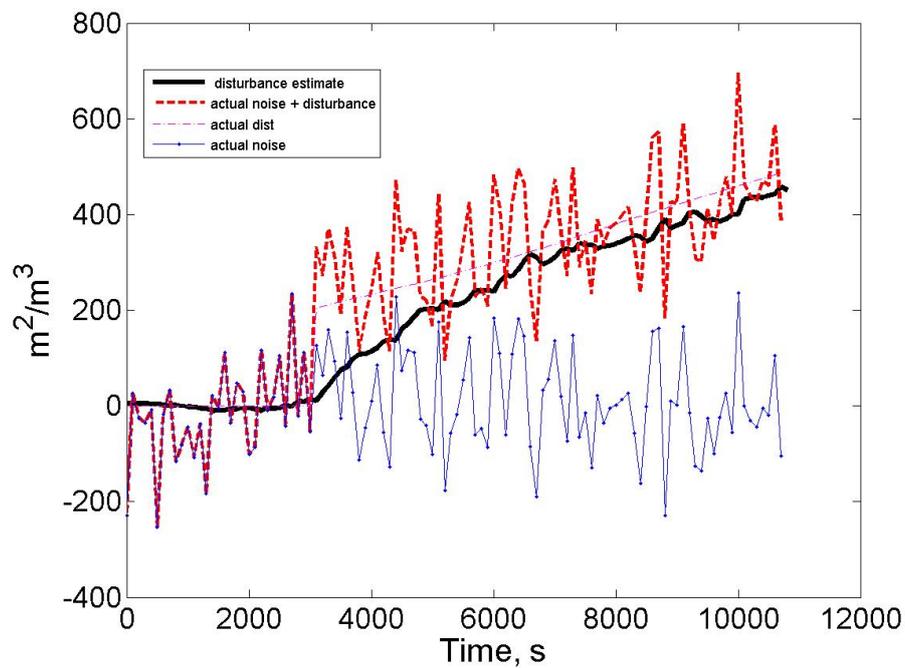


Figure 5-35: composition of the perturbation and its estimate using UKF.

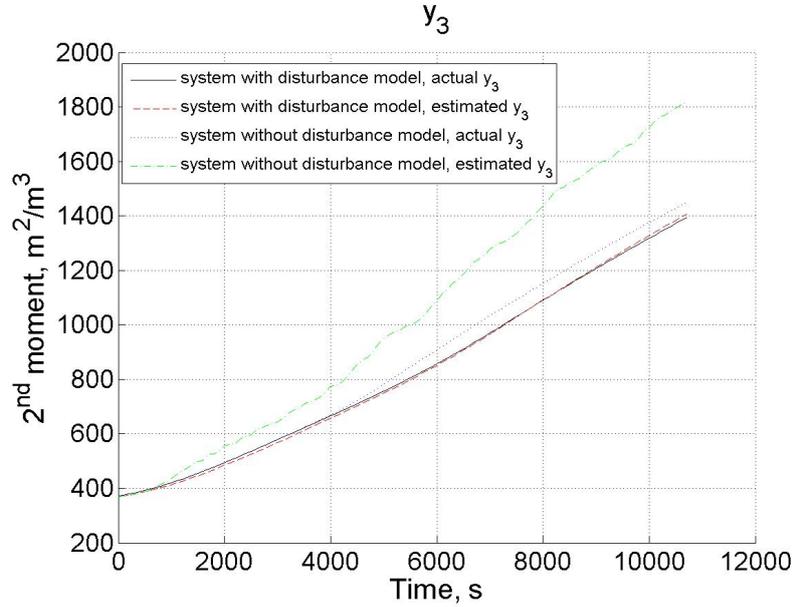


Figure 5-36: second moment and its estimate. EKF was employed here.

disturbance was included in the observer. Figures 5-36, 5-37 show the second moment when EKF and UKF are used in the control strategy, respectively. We see how the second moment of the system without a disturbance model drifts away to a larger extent from the true state than that of the system including a disturbance model.

The estimation error of the second moment is shown in Figures 5-38 and 5-39 for each filter. We see how the estimation error decreases by using an augmented observer which includes a disturbance model.

As a consequence of a smaller estimation error we expect improvement when tracking the reference trajectory for the crystals' growth rate. The tracking error is shown in Figures 5-40, where EKF was employed, and 5-41 where we used UKF. Indeed, we see how the tracking error is diminished when we are able to estimate the value of the disturbance perturbing the system, we are able to drive the system to the setpoint in the interval from $t \approx 4000$ to $t \approx 8000s$ where control is feasible. We see that with the strategy of estimating the value of a disturbance we may improve the performance of the system.

Summarizing our findings from this chapter we have,

- Concerning observers,
 - EKF and UKF showed a more balanced performance than the rest of the observers. These two filters presented a low computational demand only above ELO. Their estimation accuracy was on top places under all the scenarios where different error sources were introduced.
 - The inclusion of a Q -varying matrix in the algorithm of EKF and UKF highly improved their estimation capabilities, while barely impacted their CPU demand, meaning that the description of the Q matrix in terms of the uncertainty attributed to the system parameters is a powerful approach.

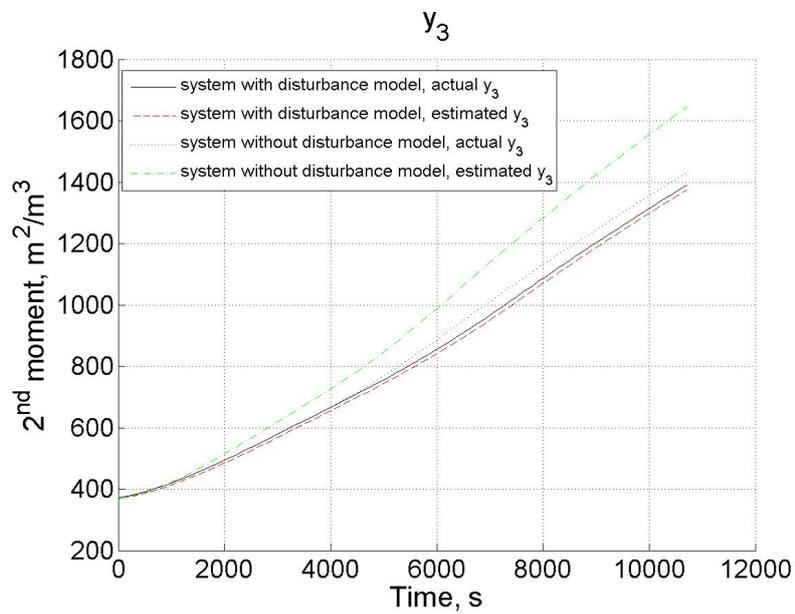


Figure 5-37: second moment and its estimate. UKF was employed here.

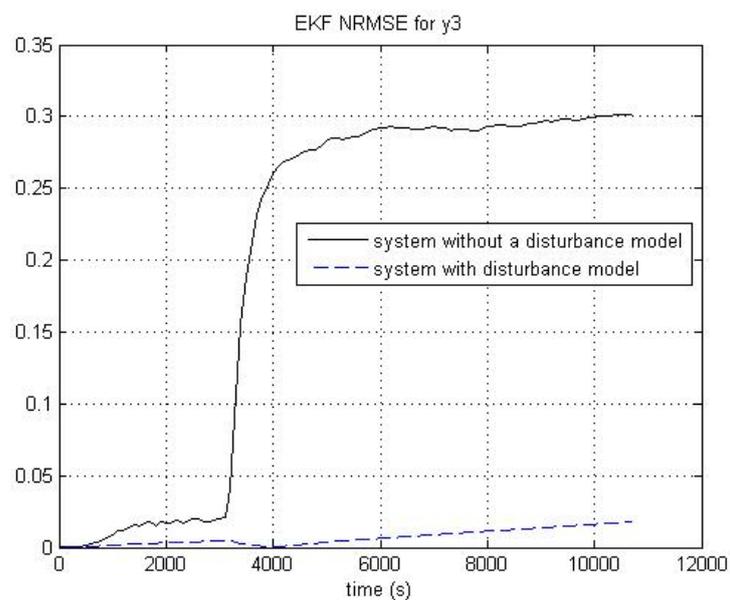


Figure 5-38: normalized estimation error of the second moment, EKF was used here.

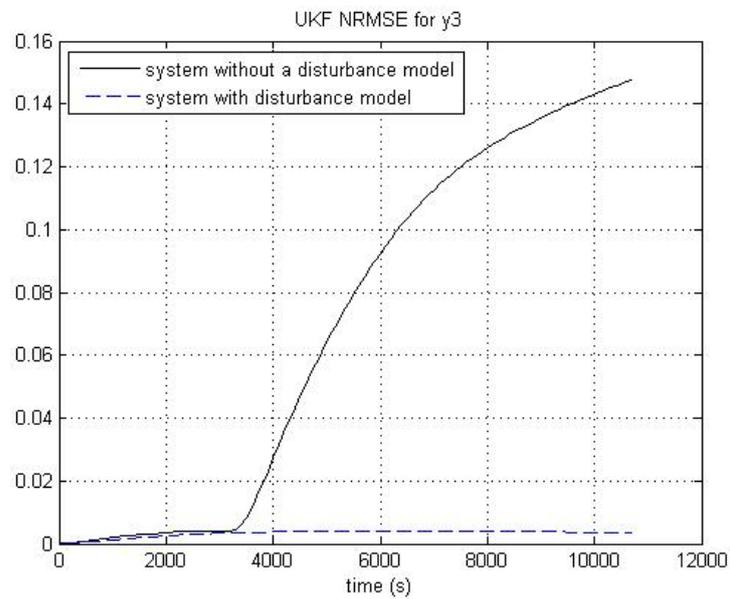


Figure 5-39: normalized estimation error of the second moment, UKF was used here.

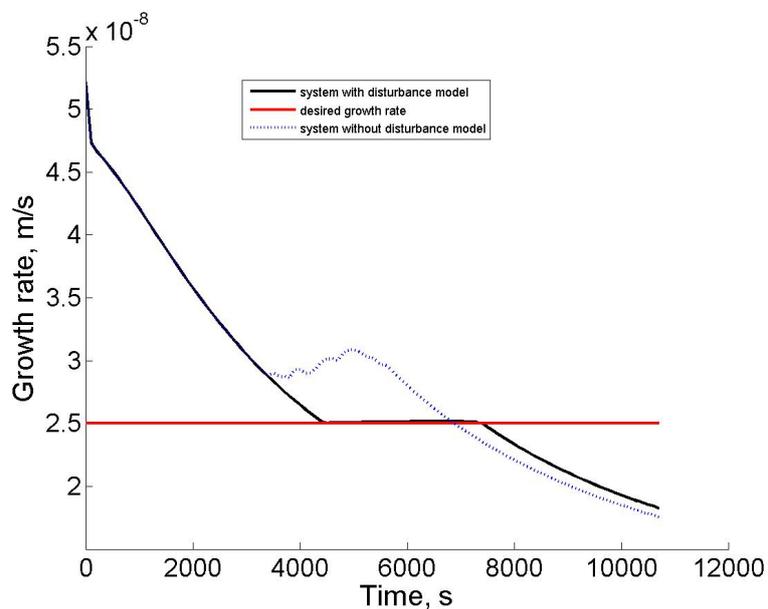


Figure 5-40: comparison of tracking error when a disturbance perturbs the system and EKF is the observer.

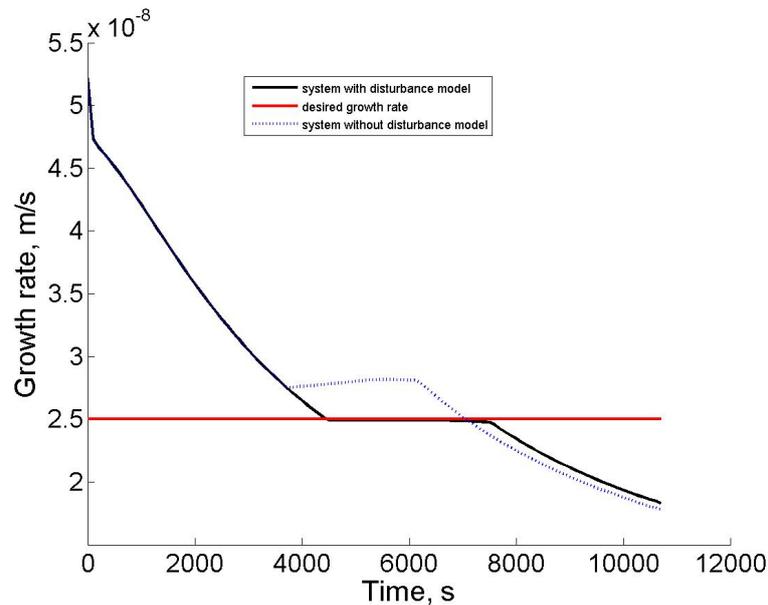


Figure 5-41: comparison of tracking error when a disturbance perturbs the system and UKF is the observer.

- MHE and ELO presented the largest estimation errors for any error source. However; MHE due to its optimization-based estimation scheme showed good robustness in the sense that no matter what error source was introduced in the system, the estimation error did not show large variations, nevertheless, its estimation accuracy was not as good as for EKF and UKF. Another shortcoming for MHE was its computational demand, it was several times higher than that for EKF and UKF.
 - EnKF presented larger estimation errors when compared to EKF and UKF. The accuracy of the filter can be improved by reselecting the ensemble members. However, the trial and error approach used in this work did not lead to a good ensemble selection. Another drawback was the computational demand, as the ensemble size was augmented, the increment in CPU time required to run the observer could not be justified by the improvement in its estimation accuracy.
 - The control strategy failed to deal with the most general scenario where all error sources were introduced in the system. Basically because the estimation accuracy of the observers was worsened with respect to the nominal case. As all error sources contributed to deteriorating the estimation and control performance, it was shown that the presence of disturbances had the largest impact. None of the observers was able to cope with this error source unless the observer was augmented with another state variable to estimate the disturbance.
- Concerning the process control evaluation,
 - Good estimation accuracy of the second statistical moment leads to good tracking performance of the controlled system.

- Estimating the value of a deterministic disturbance by including a model of its dynamics in the observer certainly improved tracking performance and helped us eliminate tracking offset.
- Implementing an integrator to suppress disturbances in a system has the advantage, when compared to disturbance estimation, that no a priori knowledge of the disturbance is required. Nevertheless, its implementation may face some difficulties when the magnitude of control inputs is constrained. Instead, for the option of disturbance estimation used in combination with an optimization-based controller, the state variable assigned to estimate the disturbance, can be included in the optimizer, which due to its ability to handle constraints in the system inputs, should automatically lead to feasible control inputs.

5-6 Summary

We described the setup for open-loop and closed-loop scenarios and stated the relevance of both cases.

We also established evaluation criteria for observers and control performance being the estimation error and computational burden for the former ones, and tracking error for the latter.

In the remaining sections of this chapter we showed results from computer simulations corresponding to the nominal case of the open-loop scenario, where only stochastic noise perturbed the system. We continued by presenting results of the general case for the closed-loop scenario, where all error sources perturbed our system. The results for the remaining cases, where error sources were introduced into the system one at a time, can be found in appendices B, C, D, E.

One section was completely devoted to disturbance rejection as it represents an important feature in terms of control. There we presented how disturbances degraded control performance and did not allow our system to reach its process goals. We demonstrate this issue by plotting the expected and actual CSD, crystal content and mean crystal size for selected observers while unmeasured disturbances affected the process.

We closed the chapter after offering a solution on how to eliminate disturbances. The methodology is based on estimating actual disturbances. To accomplish that, the same observer used for state estimation is then augmented with a number of variables assigned to disturbance estimation. We described this methodology and also compared results when no action is taken to suppress disturbances. A more complete discussion of the topic can be found in appendix F.

5-7 Remarks

An open-loop analysis allows us to get a grip on the estimation accuracy of every observer. It also serves as a reference to evaluate computational burden.

Under the closed-loop setup all error sources were studied, being unmeasured disturbance the error source which more drastically affected, in a negative way, estimation accuracy and

also control performance. Therefore estimation NRMSE and tracking NRMSE were larger when an unmeasured disturbance perturbed our system than when any other error source was investigated.

In the present case it was possible to show a relationship where better state estimation lead to better control performance. In fact accurate estimation of second and third leading moments is credited to improve control performance. When these variables were more accurately estimated the tracking error with respect to the preset control trajectory was reduced.

From observers NRMSE plots, it is shown that ELO and MHE present larger estimation errors than, for instance EKF and UKF. This is attributed to the fact that the former observers are deterministic, while the latter ones are conceived as stochastic observers and seem to perform better under a stochastic scenario. Furthermore, usage of the algorithm to calculate a time-varying system covariance matrix certainly improved the capabilities of EKF and UKF. This allows us to state that describing system uncertainty under the hypothesis made by Valappil and Georgakis [43] is adequate for our case study.

EnKF with 20 ensemble members ranked barely above deterministic observers and certainly below the other stochastic ones in terms of estimation accuracy. Although increasing the ensemble size helped improve estimation, it definitely had a negative impact in computational burden.

ELO, EKF and UKF were on top in terms of low computational burden. Addition of a time-varying matrix in the case of EKF and UKF increased the computational burden, but it still remained far below MHE and EnKF.

Finally, it was not possible to suppress the effects of an unmeasured disturbance unless the observer in turn was modified to account for this error source. There was certainly a large improvement between the cases where no action was taken and that with an augmented observer. This indicates that estimation of disturbances is a useful technique to tackle this problem.

Conclusions and Recommendations

Conclusions

Evaluation of every observer was made by analyzing the results of computer simulations of the system under investigation. Every observer was coupled to the same optimizer, this observer-optimizer pair gave rise to an NMPC framework which had proved to be useful in similar applications according to previous studies.

The system was perturbed with different error sources, namely unmeasured disturbances, uncertain initial conditions, model mismatch and systematic error. The following aspects were evaluated:

- in terms of the observer
 - Estimation accuracy, quantified as NRMSE.
Concerning estimation accuracy stochastic observers worked better than deterministic ones. This was expected as the case study involved stochastic variables. EKF and UKF came on top as best options since their NRMSEs were the smallest throughout all the tests. Specially the variants including a time-varying system covariance matrix. This indicates that quantifying uncertainty of a batch process as described by the algorithm to calculate the time-varying matrix was adequate.
 - Computational burden, quantified as CPU time
In terms of computational burden the observers ranked, from best to worst: ELO, EKF, UKF, EKF with time-varying matrix, UKF with time-varying matrix, EnKF and MHE. Best in this case indicates a shorter CPU time.

We should mention here that EnKF was worse than the rest of stochastic observers, but roughly better than MHE. This remains valid for estimation accuracy and CPU time when an ensemble of 20 members was used. Increasing the ensemble size did help improve estimation accuracy but severely increased its required CPU time. Another path to improve EnKF estimation accuracy is to play with the probability distribution from

which the ensemble members are withdrawn. If the states distributions can be properly represented by the ensemble, the estimation accuracy should improve accordingly.

- in terms of control performance

- Control performance, quantified as the tracking NRMSE

Minimizing the error when tracking a preset growth rate is the main control goal. This allows production of crystals with a narrow CSD and centered at a desirable mean crystal size. It also allows to achieve a required quantity of product. Therefore in terms of control performance, smaller tracking NRMSE equals better crystals quality and enough product per batch.

In this regard UKF and EKF both with time-varying matrix gave the best results as they presented the smallest tracking NRMSE throughout all tests.

Unmeasured disturbances demonstrated to be the most detrimental error source affecting the system. This error severely increased the estimation errors and also caused a large deviation with respect to the reference to be tracked by the controller. When measurements are affected by this type of error an observer interprets erroneous measurements as correct ones. This shifts the standing point of an observer and forces it to converge to values biased from the correct ones needed.

None observer, as initially implemented, was able to compensate for the latter error source, however, the procedure of augmenting the observers in order to estimate the disturbance and include this information in the optimizer to suppress its effect was successful. Even in the presence of an unmeasured disturbance we were able to improve estimation accuracy and also tracking error. We must warn the user that this methodology requires a priori knowledge of the disturbances.

A relation between estimation accuracy and control performance was certainly spotted. It was shown that good estimation of the second and third leading moments translates into a smaller tracking error. This relies on the fact that second and third moments are related to total area and total volume of the crystals. Both variables are explicitly used to determine the concentration, that in turn is necessary to determine crystal growth rate, which is our controlled quantity.

Based on our findings we state that UKF with time-varying matrix allowed to achieve the best control performance. It edged EKF with time-varying matrix in this regard, but its CPU time was slightly higher, nevertheless the difference was so small that should not matter in a practical implementation.

We conclude that since UKF with time-varying matrix performed better than all remaining observers in terms of estimation accuracy and control performance, while its computational demand was higher but still very close to the one required by EKF and ELO, then UKF should be the choice for a future real-time implementation.

Recommendations

- EnKF has built a reputation as an effective estimator in engineering applications where highly nonlinear models containing hundreds or thousands of state variables are used.

Therefore, it should not be discarded as an option when a more detailed model describing the crystallization process is required. Generating the ensemble under a different criteria from that used inhere should be implemented to reevaluate the filter. Information on sampling techniques may be found in literature cited in the work of Prof. Geir Evensen, Evensen [10], Evensen [11] and also in his website, <http://enkf.nersc.no/>.

- Optional to the observers presented here, there exist some other variants that do not require linearization of the system as EKF and ELO do. They should be borne in mind for further applications where more complex or non-differentiable models are employed, see for instance, Nørgaard [38], Shi and Han [40], Chen [6], Rawlings and Bakshi [37].

Bibliography

- [1] PAON course, Industrial Crystallization and Precipitation, course notes.
- [2] G. Besançon. *Advanced Topics in Control Systems Theory, chapter 2*. Springer, 2006.
- [3] R. R. Bitmead, M. Gevers, and V. Wertz. *Adaptive Optimal Control The Thinking Man's GPC*. Prentice Hall, chapter 3, 1990.
- [4] R.R. Bitmead, M. Gevers, and V. Wertz. Tutorial overview of model predictive control. *Prentice Hall*, 1990.
- [5] G. Burgers, P.J. van Leeuwen, and G. Evensen. On the analysis scheme in the ensemble kalman filter. *Monthly Weather Review*, 1998.
- [6] Z. Chen. Bayesian filtering: From kalman filters to particle filters , and beyond. *online document*, <http://www.mendeley.com/research/bayesian-filtering-from-kalman-filters-to-particle-filters-and-beyond/>.
- [7] E.J. Davison and H.W. Smith. Pole assignment in linear time-invariant multivariable systems with constant disturbances. *Automatica*, 1971.
- [8] D. Dochain. State and parameter estimation in chemical and biochemical processes: A tutorial. *Journal of Process Control*, 13(8):801-818, 2003.
- [9] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research*, 1994.
- [10] G. Evensen. The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean dynamics*, 2003.
- [11] G. Evensen. *Data Assimilation: the Ensemble Kalman Filter*. Springer, 2006.
- [12] P. K. Findeisen. *Moving horizon state estimators for discrete time systems*. PhD thesis, University of Winsconsin-Madison, 1997.

- [13] B.A. Francis and W.M. Wonham. The internal model principle of control theory. *Automatica*, 1976.
- [14] S. Gillijns, O.B. Mendoza, J. Chandrasekar, B.L.R. De Moor, D.S. Bernstein, and A. Ridley. What is the ensemble kalman filter and how well does it work. *American Control Conference*, 2006.
- [15] Y. Hao, Z. Xiong, F. Sun, and X. Wang. Comparison of unscented kalman filters. *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation*, 2007.
- [16] R. Hermann and A.J. Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 1977.
- [17] D. Jeltsema and M.A. Scherpen. Modeling and systems analysis. *TU Delft publications, Lecture notes, chapter 5*, 2005.
- [18] S.J. Julier and J.K. Uhlman. A general method for approximating nonlinear transformations of probability distributions. *Technical Report, University of Oxford, Department of Engineering Science*, 1996.
- [19] Uhlman J. K. Julier S. J. A new extension of the kalman filter to nonlinear systems. Technical report, University of Oxford, Dept. of Engineering Science.
- [20] R. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 1960.
- [21] T. Lefebvre, B. Herman, and J. De Schutter. Kalman filters for nonlinear systems: a comparison of performance. *International Journal of Control*, 2001.
- [22] D. Luenberger. Observing the state of a linear system. *IEEE Transactions on Military Electronics*, 1964.
- [23] A. Mesbah, A.N.N. Kalbasenka, A.E.M. Huesman, H.J.M. Kramer, P.J. Jansens, and P.M.J. Van den Hof. Real-time dynamic optimization of crystal yield in fed-batch evaporative crystallization of ammonium sulphate. *Proc. 14th International Workshop on Industrial Crystallization*, 2007.
- [24] A. Mesbah, A. N. Kalbasenka, A. E. M. Huesman H. J. M. Kramer, and P. M. J. Van den Hof. Real-time dynamic optimization of batch crystallization processes. *In Proc. of the 17th IFAC World Congress, pages 3246–3251, Seoul Korea.*, 2008.
- [25] I. Mora Moreno. *Literature survey: nonlinear observers*. 2008.
- [26] J.W. Mullin. *Ullman’s Encyclopedia of Industrial Chemistry, vol. B2 Unit Operations*. Wiley-VCH, 1988.
- [27] K.R. Muske and T.A. Badgwell. Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 2001.

- [28] Z.K. Nagy. Strategic feedback control of pharmaceutical crystallization processes. *online document, summary of grant EP/E022294/1 from the Engineering and Physical Sciences Research Council (EPSRC) valid from May 2007 to October 2010.* <http://gow.epsrc.ac.uk/ViewGrant.aspx?GrantRef=EP/E022294/1>.
- [29] H. Nijmeijer and A.J. van der Schaft. Nonlinear dynamical control systems. *Springer, chapter 3*, 1990.
- [30] G. Pannocchia and J.B. Rawlings. Disturbance models for offset-free model-predictive control. *AIChE Journal*, 2003.
- [31] T.D. Powell. Automated tuning of an extended kalman filter using the downhill simplex algorithm. *Paper AAS 99-368, AASIAIAA Astrodynamics Specialist Conference, Girdwood, Alaska*, 16-19 August 1999.
- [32] C.C. Qu and J. Hahn. Computation of arrival cost for moving horizon estimation via unscented kalman filter. *J. Process Control*, 2008.
- [33] A. Randolph and M. A. Larson. *Theory of Particulate Process*. Academic Press, 1988.
- [34] C.V. Rao and J.B. Rawlings. Constrained process monitoring: Moving-horizon approach. *AIChE Journal*, 48(1):97-109, 2002.
- [35] K. Rapp. *Nonlinear estimation and control in the iron ore pelletizing process*. PhD thesis, Norwegian University of Science and Technology, Faculty of Information Technology, Mathematics and Electrical Engineering, 2004.
- [36] J. Rawlings and B. Bakshi. Particle filtering and moving horizon estimation. *Computers & Chemical Engineering*, 2006.
- [37] J. Rawlings and B. Bakshi. Particle filtering and moving horizon estimation. *Computers & Chemical Engineering*, 2006.
- [38] M. Nørgaard, N.K. Poulsen, and O. Ravn. Advances in derivative-free state estimation for nonlinear systems. *Technical Report, Technical University of Denmark, Dept. of Mathematical Modeling*, 1998.
- [39] D.G. Robsертson, J.H. Lee, and J.B. Rawlings. A moving horizon-based approach for least-squares estimation. *AIChE Journal*, 42(8):2209-222, 1996.
- [40] Y. Shi and C. Han. The divided difference particle filter. *10th International Conference on Information Fusion*, 2007.
- [41] F.G. Shinskey. *Feedback Controllers for Process Industries*. McGraw-Hill, 1994.
- [42] K.J. Åström and B. Wittenmark. Computer controlled systems. *Prentice Hall, 2nd edition, chapter 9, 11*, 1990.
- [43] J. Valappil and C. Georgakis. Systematic estimation of state noise statistics for extended kalman filters. *AIChE Journal*, 2000.
- [44] T. J. J. van den Boom and T. C. P. M. Backx. *Model Predictive Control, lecture notes*. TU Delft, 2007.

- [45] R. van der Merwe and E. Wan. Sigma-point kalman filters for probabilistic inference in dynamic state-space models. *In Proceedings of the Workshop on Advances in Machine Learning, Montreal Canada, 2003.*
- [46] G. Welch and G. Bishop. An introduction to the kalman filter. *Technical Report, TR95-041, University of North Carolina - Chapel Hill, 1995.*

Appendix A

Open-loop case: complementary figures

Figures below show how eliminating any error source from the analyzed system certainly improves the performance of deterministic observers.

Please note that estimation accuracy was not degraded to a large extent for the stochastic observers which underlines their robustness in this sense.

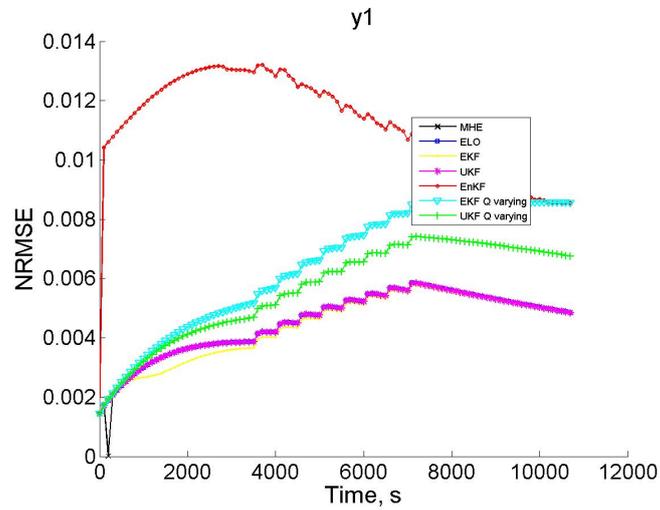


Figure A-1: NRMSE y_1 , open-loop case without any error source

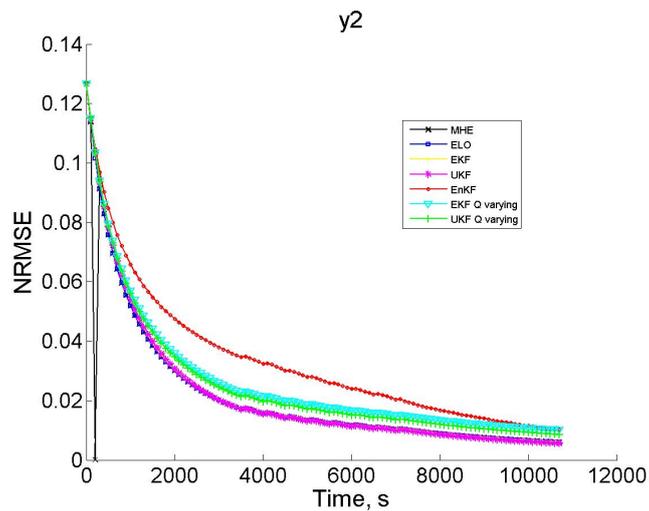


Figure A-2: NRMSE y_2 , open-loop case without any error source

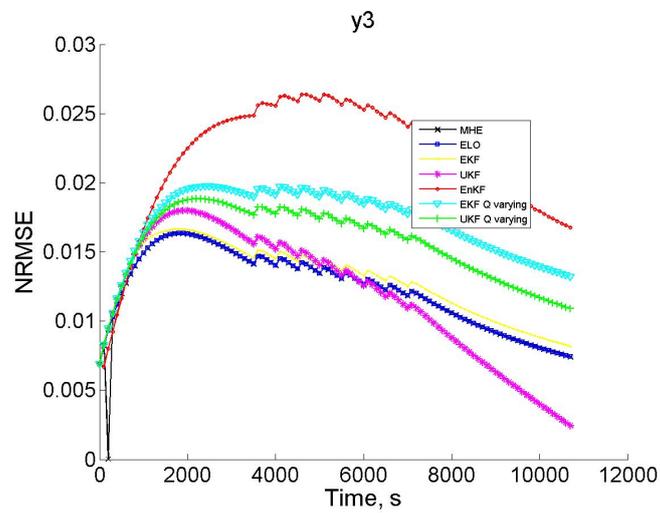


Figure A-3: NRMSE y_3 , open-loop case without any error source

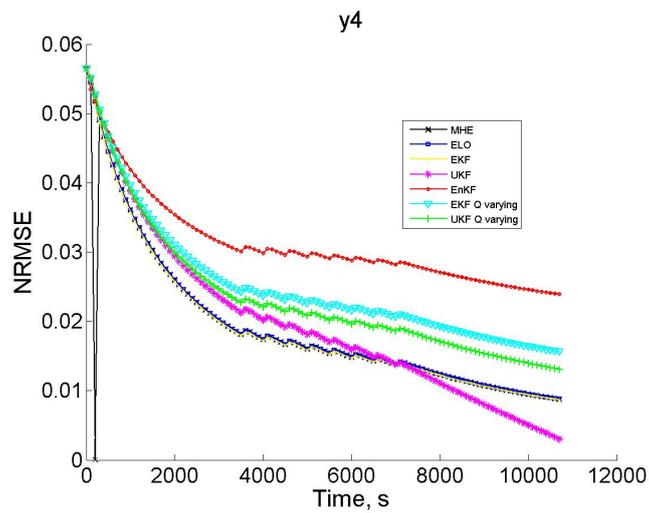


Figure A-4: NRMSE y_4 , open-loop case without any error source

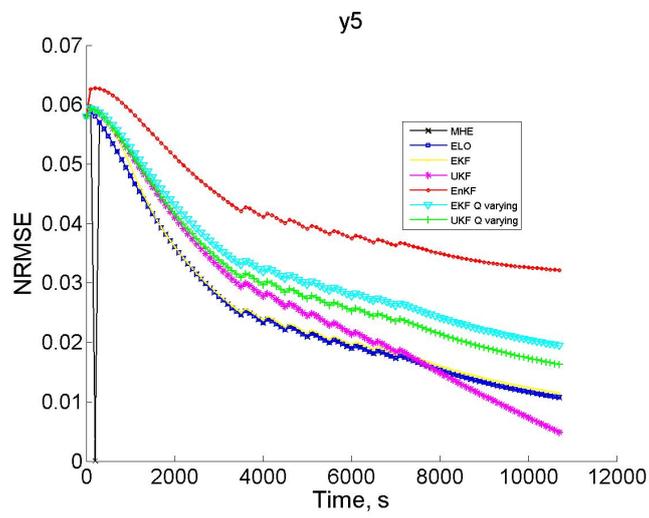


Figure A-5: NRMSE y_5 , open-loop case without any error source

Appendix B

Closed-loop nominal case: complementary figures

Before we proceed to show the results achieved in the closed-loop nominal case we give relevant parameters used for each observer:

- ELO, observer gain $K = (0.054, 0.649, 2.711, 19.822, 6.184)$,
- MHE, estimation depth, 500 s ,
- EKF, system's noise initial covariance matrix

$$\mathbf{Q} = 9 \begin{pmatrix} 5 \cdot 10^{20} & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 \cdot 10^{-2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \cdot 10^{-9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \cdot 10^{-4} \end{pmatrix}, \quad (\text{B-1})$$

measurement's noise initial covariance matrix

$$\mathbf{R} = 9 \begin{pmatrix} 5 \cdot 10^{12} & 0 & 0 & 0 & 0 \\ 0 & 3 \cdot 10^2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 8 \cdot 10^{-13} & 0 \\ 0 & 0 & 0 & 0 & 7 \cdot 10^{-16} \end{pmatrix}, \quad (\text{B-2})$$

item UKF, system's noise initial covariance matrix

$$\mathbf{Q} = 9 \begin{pmatrix} 5 \cdot 10^{18} & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 \cdot 10^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 \cdot 10^{-2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \cdot 10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \cdot 10^{-7} \end{pmatrix}, \quad (\text{B-3})$$

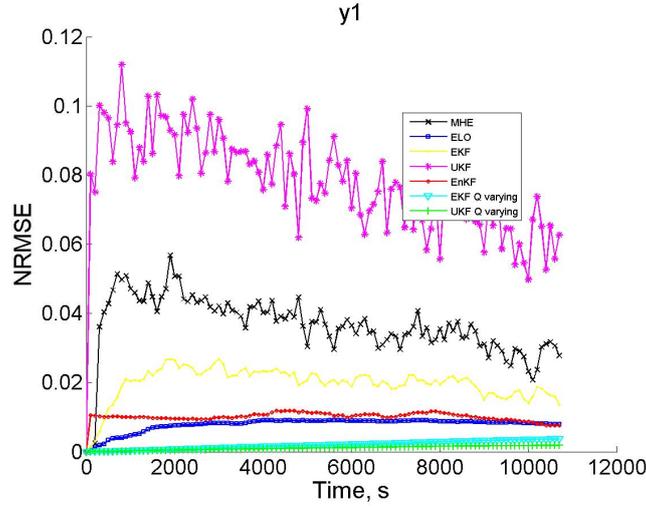


Figure B-1: NRMSE y_1 , closed-loop, nominal case

measurement's noise initial covariance matrix

$$\mathbf{R} = 9 \begin{pmatrix} 1 \cdot 10^9 & 0 & 0 & 0 & 0 \\ 0 & 9 \cdot 10^4 & 0 & 0 & 0 \\ 0 & 0 & 1 \cdot 10^{-5} & 0 & 0 \\ 0 & 0 & 0 & 1 \cdot 10^{-1} & 0 \\ 0 & 0 & 0 & 0 & 1 \cdot 10^{-5} \end{pmatrix}, \quad (\text{B-4})$$

- EnKF, 20 ensemble members were used and the noise added to the observation was drawn from a zero-mean normal distribution with variances given by

$$\mathbf{R} = 3 \cdot 10^5 \begin{pmatrix} 1 \cdot 10^{12} & 0 & 0 & 0 & 0 \\ 0 & 9 \cdot 10^4 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \cdot 10^{-1} & 0 \\ 0 & 0 & 0 & 0 & 1 \cdot 10^{-4} \end{pmatrix}. \quad (\text{B-5})$$

For the variants of the EKF and UKF where a time-varying Q matrix was used, the parameters remained the same as for the classical implementations.

Now we show the behavior of the NRMSE for the present case,

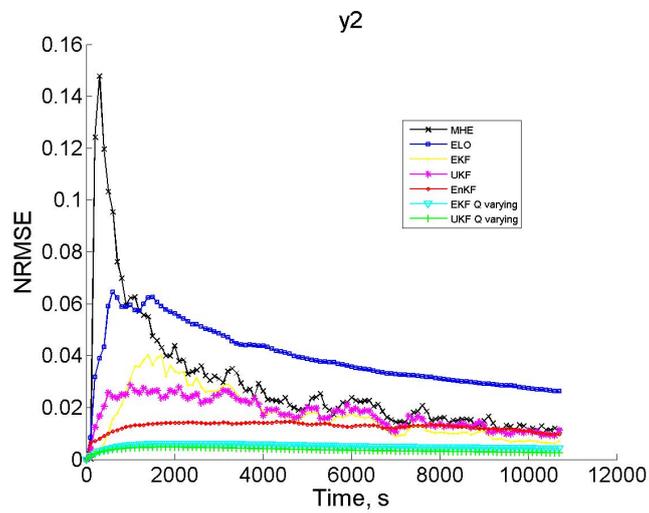


Figure B-2: NRMSE y_2 , closed-loop, nominal case

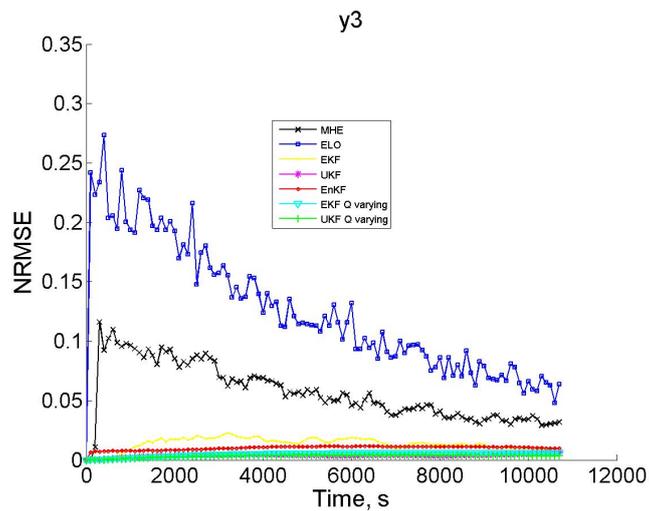


Figure B-3: NRMSE y_3 , closed-loop, nominal case

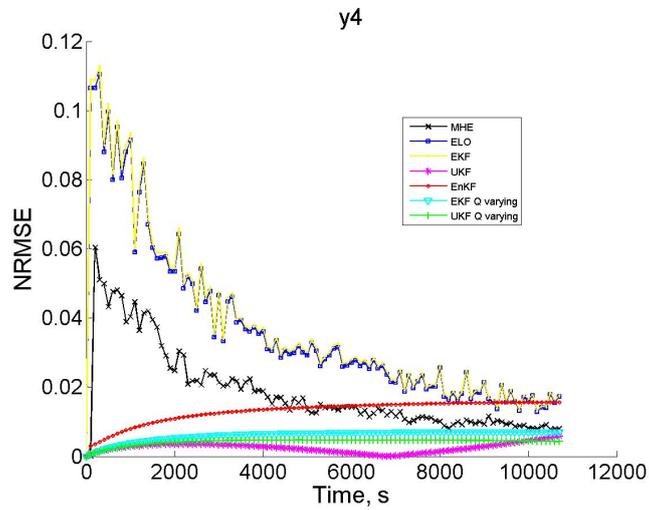


Figure B-4: NRMSE y_4 , closed-loop, nominal case

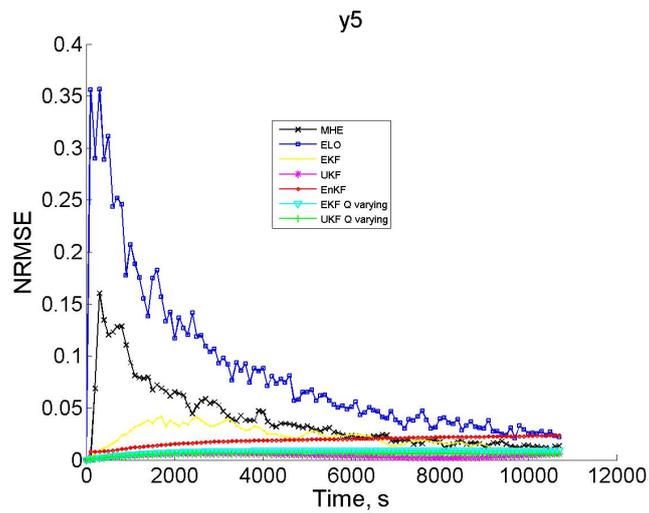


Figure B-5: NRMSE y_5 , closed-loop, nominal case

Closed-loop unknown initial conditions case: complementary figures

Unknown initial conditions cause that the plant and the observer-controller pair are both initialized at different points, this gives rise to larger estimation errors at the beginning of the simulation for all variables but y_1 when compared to the nominal case. Since the initial errors are larger than for the nominal case, and given the restriction on the inputs of the system, some of the filters are not capable of achieving the same final estimation error than in the nominal case. That is due to convergence speed rate. This statement should remain true as long as the system is fully observable, from control theory; if a system is observable, then reconstructing the values of state variables should be feasible in finite time.

The growth rate tracking error is shown below,

Table C-1: Initial conditions for the plant, observer and controller, unknown initial conditions case

Variable	Plant	Observer & controller
m_0	$1.08 \cdot 10^{10}$	$1.13 \cdot 10^{10}$
m_1	$3.42 \cdot 10^{05}$	$3.59 \cdot 10^{05}$
m_2	$3.69 \cdot 10^{02}$	$3.87 \cdot 10^{02}$
m_3	$8.30 \cdot 10^{-02}$	$8.72 \cdot 10^{-02}$
m_4	$2.49 \cdot 10^{-05}$	$2.61 \cdot 10^{-05}$
C	$4.57 \cdot 10^{-01}$	$4.66 \cdot 10^{-01}$

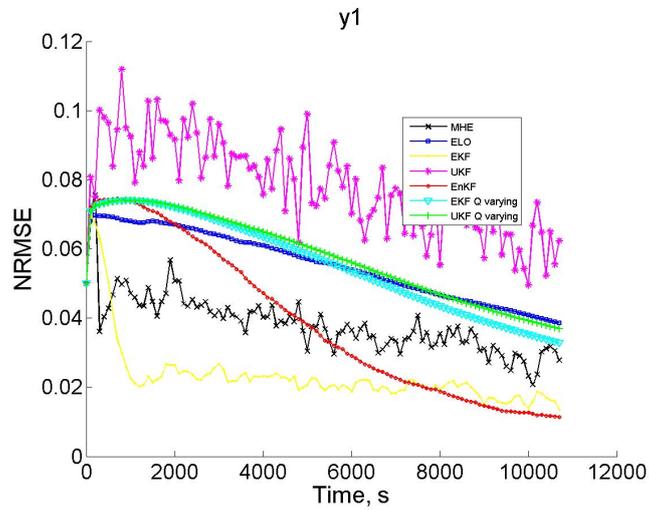


Figure C-1: NRMSE y_1 , closed-loop, uncertain initial conditions

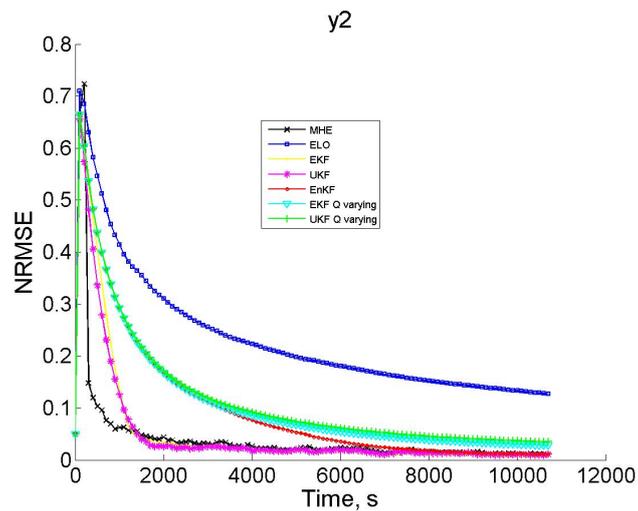


Figure C-2: NRMSE y_2 , closed-loop, uncertain initial conditions

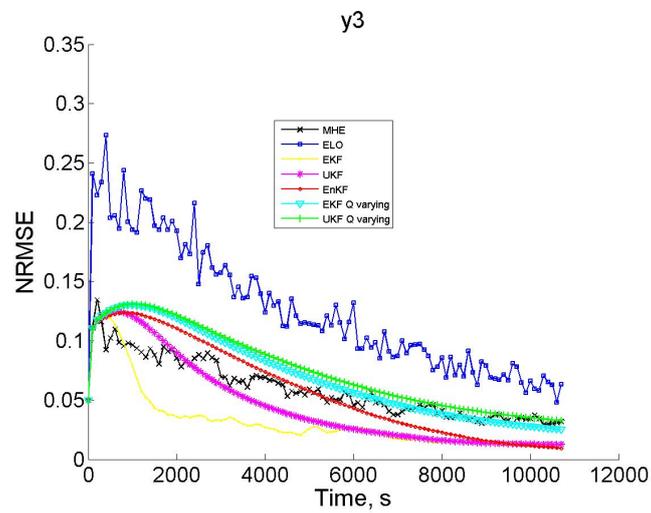


Figure C-3: NRMSE y_3 , closed-loop, uncertain initial conditions

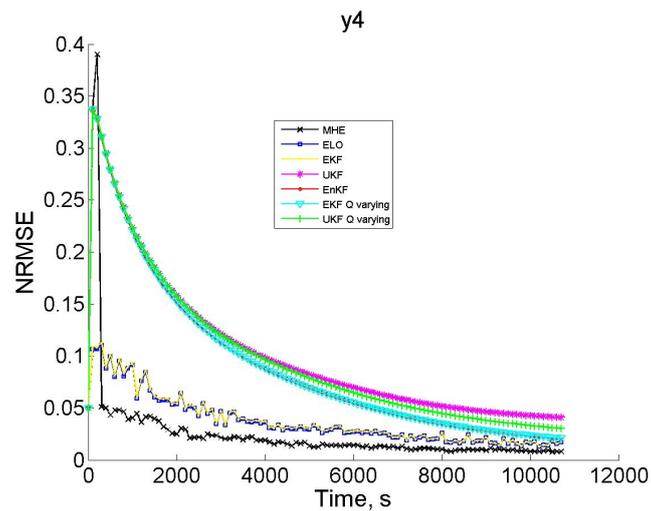


Figure C-4: NRMSE y_4 , closed-loop, uncertain initial conditions

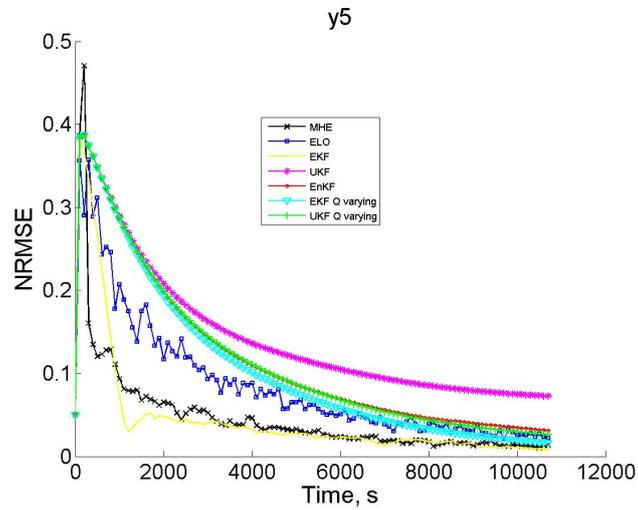


Figure C-5: NRMSE y_5 , closed-loop, uncertain initial conditions

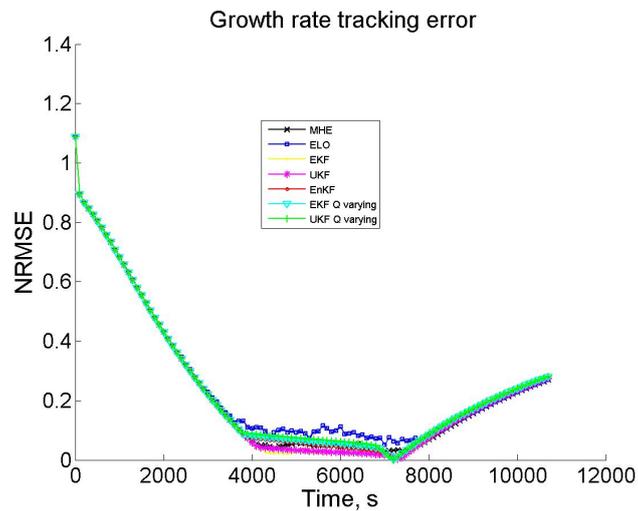


Figure C-6: Growth rate tracking error, closed-loop, uncertain initial conditions

Closed-loop systematic measurement error case: complementary figures

In this case an offset of +5% of the plant's output value is introduced in the measurement feeding the observer.

The offset increases the estimation error, but the observers still converge, although they seem to be converging to a biased value.

Systematic measurement error generates an offset in the control performance for some filters. This can be attributed to the fact that those filters rely more in the measurement from the plant than in the predictions made by their internal model of the system.

In general, if a model is not able to accurately reproduce reality, which is always the case, systematic measurement error will lead to an offset in tracking trajectories, this is because if we are not able to accurately make predictions with a model then we must rely on measurements from the plant, however; if measurements are erroneous, it is not possible to know where the plant is standing, and even if the plant is already at a desired location in state-space, the inputs of the controller will drift it away because the controller, based on erroneous measurements, thinks that the plant is not at the aimed location.

The growth rate tracking error is shown below,

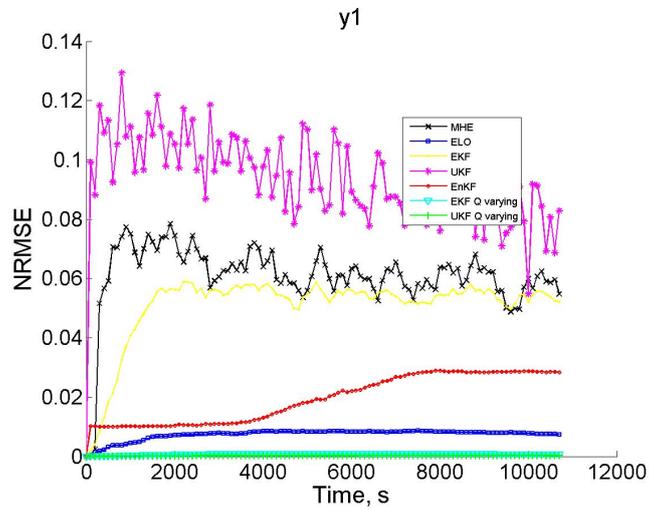


Figure D-1: NRMSE y_1 , closed-loop, systematic measurement error

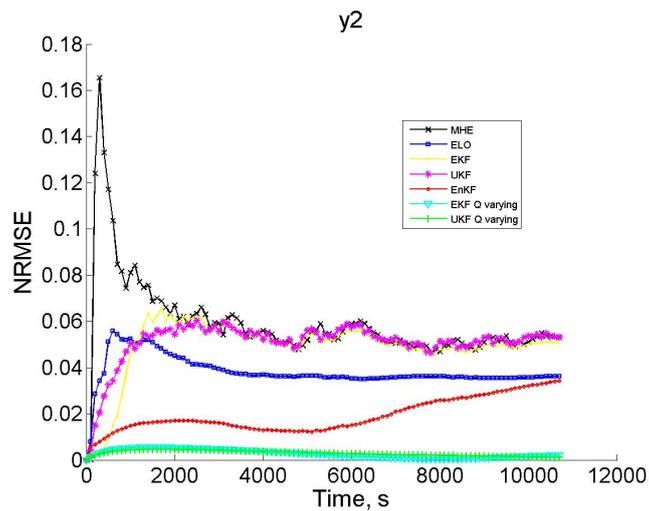


Figure D-2: NRMSE y_2 , closed-loop, systematic measurement error

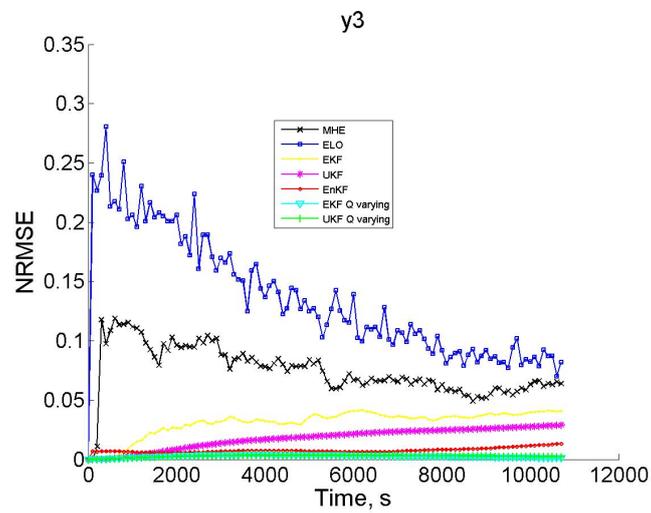


Figure D-3: NRMSE y_3 , closed-loop, systematic measurement error

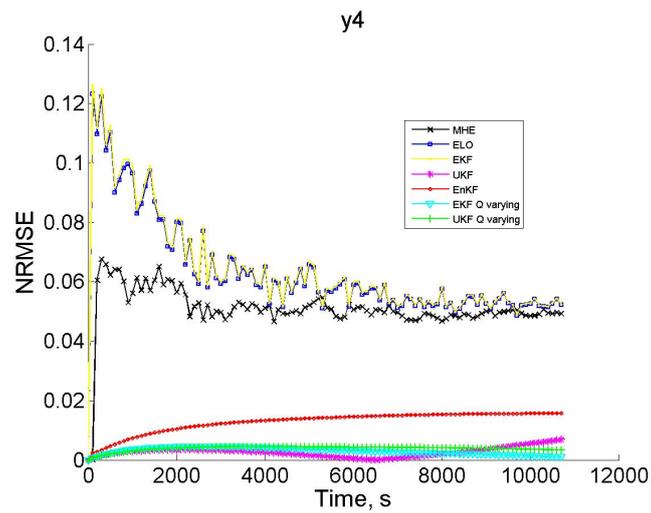


Figure D-4: NRMSE y_4 , closed-loop, systematic measurement error

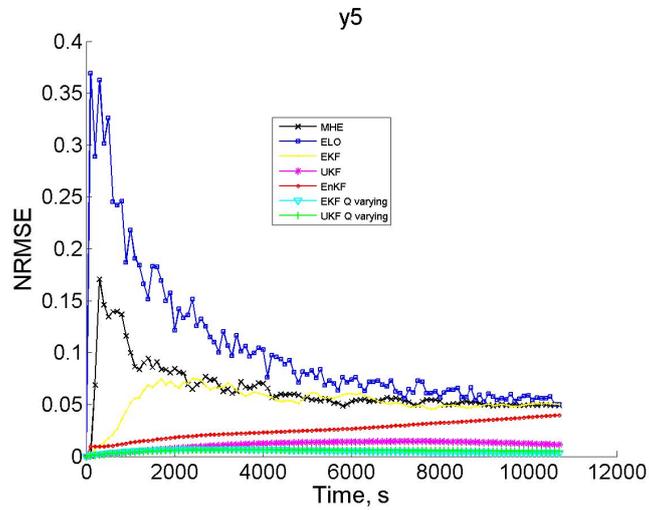


Figure D-5: NRMSE y_5 , closed-loop, systematic measurement error

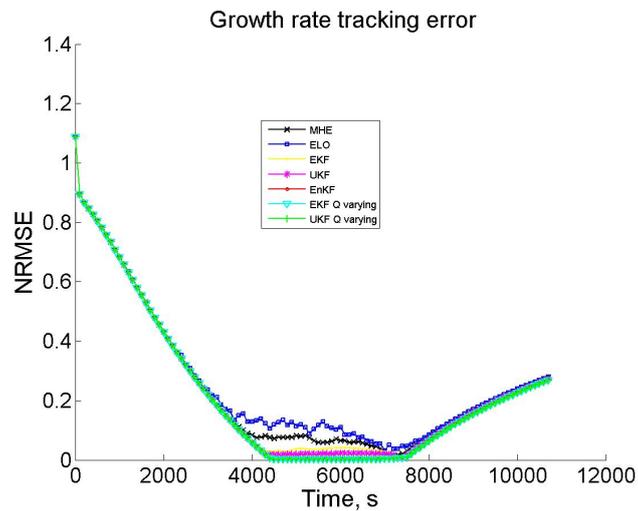


Figure D-6: Growth rate tracking error, closed-loop, uncertain initial conditions

Appendix E

Closed-loop model mismatch case: complementary figures

In this case the parameters k_g and k_b were varied in a +35% with respect to their nominal value. The perturbation was made only on the observers and the optimizer, the plant preserved its nominal value.

UKF and EKF with a varying Q matrix as well as the EnKF showed divergence and had to be retuned. ELO also diverged, although this is not a surprise as it had shown troubles when only stochastic noise was introduced into the system. Re-tuning the observers suffices to solve this issue. In fact, retuned observers are used later on when all the sources of error are introduced into the system at the same time. In the case of the MHE, the use of an optimization algorithm allows it to compensate for the discrepancy between the model used by the observer-controller pair and that in the plant.

The growth rate tracking error is shown below.

Model mismatch affect the estimation accuracy of the system to a little extent since the graphs for the estimation error remain almost unchanged, with respect to those for the nominal case, for various filters. The explanation is that, variation in the parameters of a system affects its dynamics and therefore the values of the system variables, but they can be affected in different degrees if the relationship is not linear.

Table E-1: Parameters model mismatch

Variable	Plant	Observer & controller
k_g	$7.50 \cdot 10^{-5}$	$1.01 \cdot 10^{-4}$
k_b	$1.02 \cdot 10^{14}$	$1.38 \cdot 10^{14}$

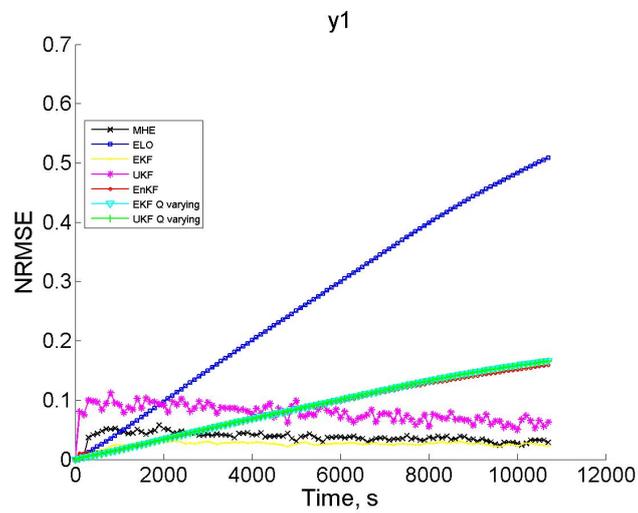


Figure E-1: NRMSE y_1 , closed-loop, model mismatch

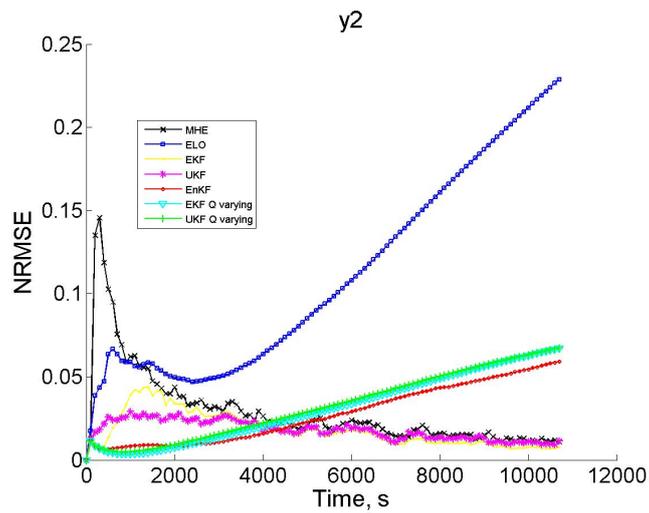


Figure E-2: NRMSE y_2 , closed-loop, model mismatch

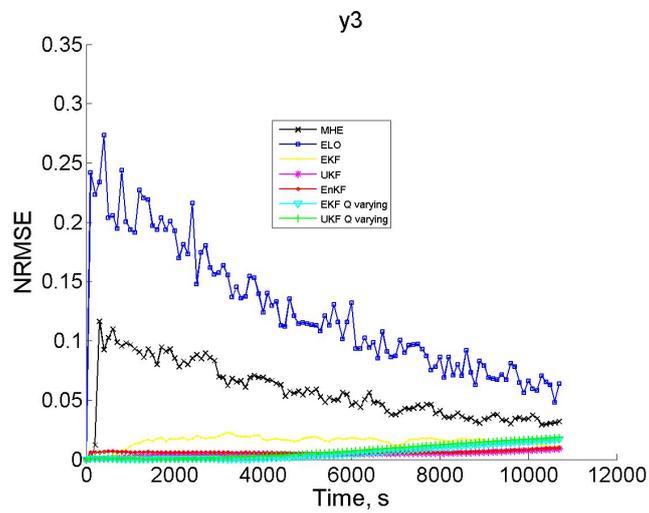


Figure E-3: NRMSE y_3 , closed-loop, model mismatch

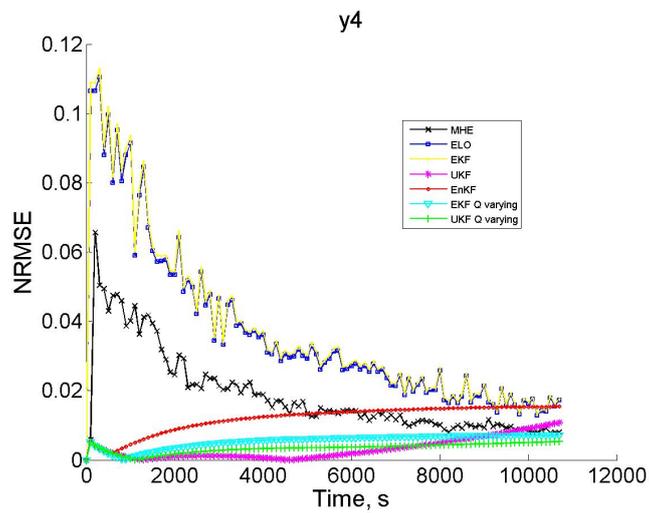


Figure E-4: NRMSE y_4 , closed-loop, model mismatch

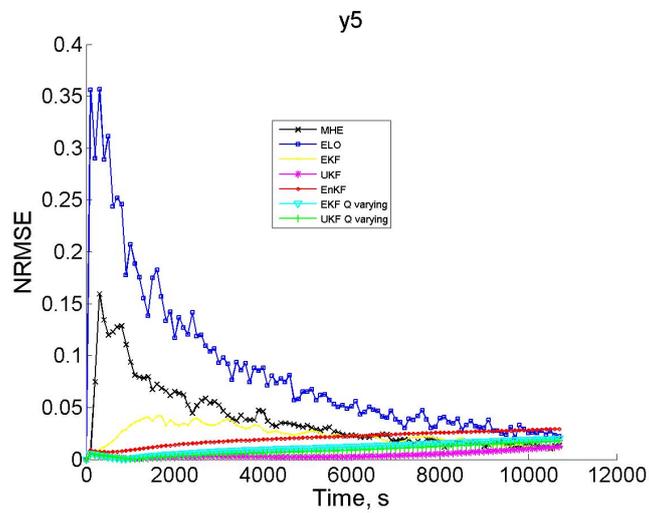


Figure E-5: NRMSE y_5 , closed-loop, model mismatch

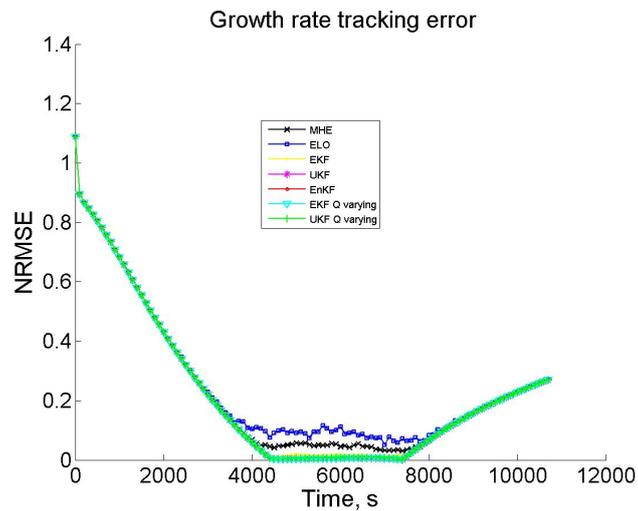


Figure E-6: Growth rate tracking error, closed-loop, uncertain initial conditions

As an example consider a system with two variables and described by,

$$\dot{s}_1 = k_a s_1 + s_2 \quad (\text{E-1})$$

$$\dot{s}_2 = k_a s_2. \quad (\text{E-2})$$

And suppose now that we are interested in the quantity,

$$l = \frac{s_1}{s_2} \quad (\text{E-3})$$

after solving the set of differential equations it turns out that model mismatch is alleviated when estimating l due to the division involved in the output equation.

If the effects of model mismatch do not vanish due to the relations between variables of the system, then that will lead to make the controller believe it is at a certain point in state-space that in reality has not been reached.

Failing to describe the behavior of a system and based on the internal model principle should lead to offset in the variables estimation, and this offset would lead to an offset when a trajectory needs to be followed.

In our case study model mismatch did not generate offset, but in general it should not be the case.

Appendix F

Disturbance rejection

F-1 Disturbance rejection for linear systems

Synthesizing controllers which regulate systems despite uncertainty in plant and controller parameters is a classical problem in control theory. The purpose of the compensator is double, one is to provide closed-loop stability, and the second is to regulate the output. The case of disturbance rejection shows that closed-loop controller performance is directly related to how accurately the disturbance model represents the actual disturbances entering a process. This concept is known as the “internal model principle” and is proposed and discussed by Francis and Wonham [13] for linear multivariable systems. Roughly speaking this principle says that in order to counteract the effects of disturbances acting on a system, it is needed that the model used to build a controller contains a copy of the dynamic structure of the disturbance. For the analysis, we consider the linear, time-invariant, discrete-time system

$$\begin{aligned}x_{k+1} &= Ax_k + Bu_k, \quad k = 0, 1, 2, \dots \\y_k &= Cx_k\end{aligned}\tag{F-1}$$

in which $y \in R^l$ is the output or measured variable, $u \in R^m$, is the system input, and $x \in R^n$ is the state of the system. We assume throughout that (A, B) is stabilizable and (C, A) is detectable.

There might be different control strategies, nevertheless in the MPC framework the control input is obtained by solving a quadratic programming problem:

$$\left\{v_j^*\right\}_0^{N-1} \triangleq \left[\sum_{j=0}^{\infty} (z_j^T C^T Q C z_j + v_j R v_j + \Delta v_j^T S \Delta v_j) \right]\tag{F-2}$$

subject to:

$$\begin{aligned}
z_{j+1} &= Az_j + Bv_j & \forall j = 0, \infty \\
\Delta v_j &\triangleq v_j - v_{j-1} \\
z_0 &= \hat{x}_k - x_s \\
v_{-1} &= u_{k-1} - u_s \\
v_j &= 0 & \forall j = N, \infty \\
y_{min} &\leq C(z_j + x_s) \leq y_{max} & \forall j = j_1, \infty \\
u_{min} &\leq v_j + u_s \leq u_{max} & \forall j = 0, N-1 \\
\Delta_{min} &\leq v_j - v_{j-1} \leq \Delta_{max} & \forall j = 0, N-1
\end{aligned}$$

with x_s, u_s the steady-state targets for the state and the input respectively. The current input u_k is obtained as $u_k = \hat{v}_0 + u_s$, where \hat{v}_0 is obtained by solving the optimization problem stated above. The results obtained under this framework are valid for a variety of control algorithms that include LQG controllers as well.

Unmeasured disturbance models

Generally speaking there is no possibility to directly measure the magnitude of a disturbance, but in order to counteract its effect, its influence on the system must be inferred. Therefore, it is possible to create a model representing the sought-after disturbance. Once this is accomplished, estimation techniques may be applied in order to determine the value of the disturbance at a certain time. The estimated value will be used by the controller to counteract the effect of the disturbance on the system.

How the disturbance is modeled is up to the practitioner, the disturbance may perturb any component of the system, that is, it may disturb the states, the input, or the output. Reconciling these facts, what matters in the end, is that the effect produced by any virtual signal on the system equals that of the real disturbance.

System identification methods may be employed to obtain a model of the disturbance, nevertheless, linear state-space models give us enough flexibility to account for many types of disturbances and in many applications this method is sufficient. Examples of disturbance models are given below. In all cases $d \in R^{n_d}$, where n_d is the number of augmented disturbance states, A_d , and C_d , determine the effects of these states on the system.

Constant output disturbance:

$$\begin{aligned}
\begin{pmatrix} x_{k+1} \\ d_{k+1} \end{pmatrix} &= \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} \begin{pmatrix} x_k \\ d_k \end{pmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k \\
y_k &= \begin{bmatrix} C & C_d \end{bmatrix} \begin{pmatrix} x_k \\ d_k \end{pmatrix}.
\end{aligned} \tag{F-3}$$

Constant state disturbance:

$$\begin{aligned} \begin{pmatrix} x_{k+1} \\ d_{k+1} \end{pmatrix} &= \begin{bmatrix} A & A_d \\ 0 & I \end{bmatrix} \begin{pmatrix} x_k \\ d_k \end{pmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k \\ y_k &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{pmatrix} x_k \\ d_k \end{pmatrix} \end{aligned} \quad (\text{F-4})$$

As can be seen from the models above, the practitioner can adjust the way the disturbance affects the system by playing with the system and output matrices such that the desired disturbance dynamics is achieved. Combining the system states, and those from the disturbance, it is possible to build a composite model with a “super state”, once this model is obtained, the “super states” can be estimated using typical observation techniques. The set of equations of a composite system is,

$$\begin{aligned} x_{k+1}^{\tilde{}} &= \tilde{A}x_k^{\tilde{}} + \tilde{B}u_k^{\tilde{}}, \quad k = 0, 1, 2, \dots \\ \hat{y}_k &= \tilde{C}x_k^{\tilde{}} \end{aligned} \quad (\text{F-5})$$

where

$$\begin{aligned} \tilde{x}_k &= (x_k, d_k)^T \\ \tilde{A} &= \begin{bmatrix} A & A_{d_1} \\ 0 & A_{d_2} \end{bmatrix} \\ \tilde{B} &= \begin{bmatrix} B \\ 0 \end{bmatrix} \\ \tilde{C} &= \begin{bmatrix} C & C_d \end{bmatrix} \end{aligned} \quad (\text{F-6})$$

with adequate dimensions.

In the case of deterministic systems a Luenberger observer can be used, and for stochastic systems a Kalman filter is adequate.

As shown by Bitmead et al. [3], the composite model can include the reference to be tracked by the system output, thus the augmented system may be used to solve an offset-free tracking problem, too.

As the main goal is to estimate the value of the disturbance, the composed system must be observable, or at least detectable. Guidelines to achieve a disturbance design which is detectable can be found in the bibliography and depend on the design method chosen. Muske and Badgwell [27], and Pannocchia and Rawlings [30] offer clear and complete details. Detectability of the augmented system is a necessary and sufficient condition for a stable estimator to exist according to Pannocchia and Rawlings [30]. They also show that the maximum dimension of the disturbance such that the augmented system is detectable is equal to the number of measurements, which might result quite counterintuitive.

Since the state vector has been augmented, the computational time for estimation and also for the optimization part will increase. This might become an issue when dealing with high order systems. Therefore the practitioner must be aware of this problem.

It is also important to notice that disturbances are not controllable by the system inputs u . However, since they are observable, we use their estimates to remove their influence from the controlled variables, as it was stated before.

Controller design

Bitmead et al. [3] states that the GPC problem may be thought as a special case of an LQG control strategy. LQG controllers are generated through the interconnection of a linear state-variable feedback control law and a linear state estimator. Indeed in the case where the plant is linear and the noise affecting the plant and the measurements are zero mean Gaussian processes as is the initial condition x_0 , this interconnection provides the optimal dynamic output feedback control law, where optimality is measured according to the LQ criterion. This is the so-called “separation principle” of linear optimal control. In his book, Gevers gives the solution of the LQG problem, this solution was used to obtain the controller gains of example 2 in this appendix. The stability properties of the controllers are also studied in chapter 4 of that book. The majority of the methodologies to achieve offset-free control when constant disturbances are affecting the system take the work done in Davison and Smith [7] as a basis. Synthesizing their work we have that:

- For there to be a solution to the problem of placing the poles of the closed-loop system anywhere in the complex plane, it is necessary that there should be at least as many manipulated inputs to the system as there are outputs for a solution to exist.
- If there is a solution, the minimum order of the feedback control system which can be used is equal to the number of outputs of the system.
- The controlled system so obtained has the additional property that it will remain asymptotic regulation for any finite changes, large or small, which may occur in the plant parameters A or B , or feedback gains, provided the augmented system still remains stable.
- Dealing with ramp-like or parabolic disturbances should present no problem when following the same approach.

Pannocchia and Rawlings [30] show that:

- If the number of measurements is greater than the number of manipulated variables, one cannot attempt to control without offset all the measured variables. However, one can choose, as controlled variables m linear combinations of the measured outputs, where m is smaller than the number of manipulated outputs.

The addition of constant, unmeasured disturbance states to the augmented model introduces unstable modes into the system that are not controllable. Therefore, the augmented system is not stabilizable and cannot be used directly in the model predictive control algorithm. This problem is overcome by using the original process model in (F-2) and shifting the steady-state target to remove the effects of the estimated constant disturbance states.

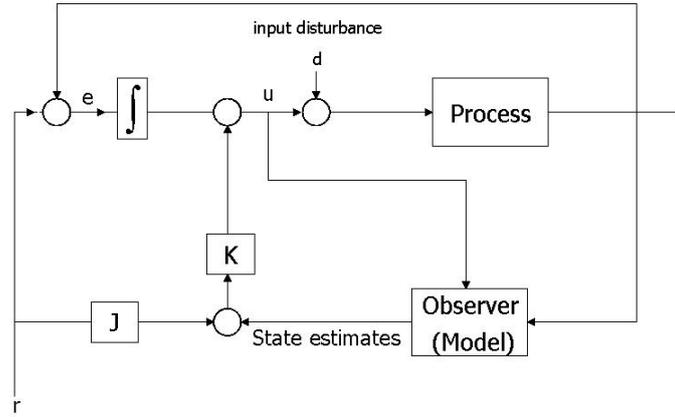


Figure F-1: System with an observer and a controller with integral action

Integral action

The second approach to suppress disturbances is that one using integral action. Shinskey [41] pinpoints that a PID controller can outperform the typical MPC implementation when a disturbance enters the input of a process upstream, of a dominant lag.

Integral action is obtained by augmenting the state that integrates the error $y^{ref} - y$, i.e.

$$x_{k+1}^i = x_k^i + y_k^{ref} - Cx_k. \quad (\text{F-7})$$

System (F-1) augmented with the new state becomes

$$\begin{pmatrix} x_{k+1} \\ d_{k+1} \end{pmatrix} = \begin{bmatrix} A & 0 \\ -C & I \end{bmatrix} \begin{pmatrix} x_k \\ x_k^i \end{pmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} 0 \\ I \end{bmatrix} y^{ref}. \quad (\text{F-8})$$

When state feedback is used the control law has the structure

$$u_k = -Lx_k - L^i x_k^i + L^{ref} y^{ref} \quad (\text{F-9})$$

Under these circumstances, the steady-state error will be zero as long as the closed-loop system is stable, even if there are minor errors in the process model. A block diagram for this case is shown in Figure F-1. When the states of the system are estimated, equation (F-9) changes the actual state variables to those given by an estimator, in essence it remains the same. The corresponding equations can be found for instance in Åström and Wittenmark [42].

Remarks

Elimination of steady-state offset is accomplished in two basic ways. The first method involves modifying the controller objective to include integration of the tracking error. This method, employed by the PID control algorithm, can also be used in the LQG or MPC framework. In the MPC framework, the integral term is incorporated by augmenting the process model with tracking error states. For large-scale systems, this augmentation can significantly increase

the computational cost of the dynamic optimization. Another disadvantage of this approach is the requirement of an anti-windup algorithm for the integral term to prevent an unnecessary, and sometimes costly, performance penalty. Furthermore, if the perturbations pertain to those disrupting a set point then the integral term can be thought as of as constantly calculating the value of the control required at the set point to cause the error go to zero, this disables the necessity to rest the controller. For these reasons, some form of integral control is typically included in most control systems. More generally, the external signals frequently include persistent deterministic components and the control engineer is required to design a controller which will force the steady-state error to be zero in the presence of such signals. A second general approach to eliminating steady-state offset involves augmenting the process model to include a disturbance. This disturbance, which is estimated from the measured process variables, may in specific cases be assumed to remain constant in the future and its effect on the controlled variables is removed by shifting the steady-state target for the controller. Although this method avoids the requirement of an anti-windup algorithm, it has the disadvantage of requiring that a separate disturbance model be designed and the disturbance states estimated. What this actually means is that a state-space representation can be made, such that, it represents the dynamics exhibited by the disturbance. It must be kept in mind that this “dynamics” must be inferred from the process behavior since we are stating that the disturbances are unmeasured; therefore we will try to guess what model suits better to mimic the influence of the disturbances on the system.

Examples

Control of roll angle of small airplane

The transfer function between the roll angle and the position of the ailerons is given by:

$$G(s) = \frac{1}{s^2(s+1)} \quad (\text{F-10})$$

This problem was solved by switching the analysis to discrete-time. Since this is a problem involving a deterministic model, the states of the system were estimated with a Luenberger observer. To control the system an LQ controller was used, once it was stabilized a step-like input disturbance was induced into the system. The response of the system to this disturbance can be seen in Figure F-2. From there, it can be seen that the controller itself is not able to completely vanish the effect of the disturbance. Integral action was used to remove the offset due to the disturbance, and can be seen in Figure F-3.

Control of the level in a tank

The level of a tank is controlled by a valve at the bottom. The flow at the valve is proportional to the level of the tank. Both the plant and the measurements are corrupted by uncorrelated zero mean noise. The state-space model of the system is given by:

$$\begin{aligned} x_{k+1} &= 0.905x_k + 0.095u_k + 0.095w_k \\ y_k &= x_k + v_k \end{aligned} \quad (\text{F-11})$$

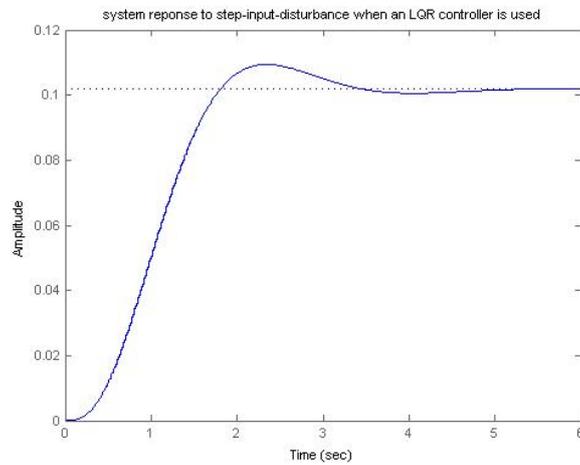


Figure F-2: Disturbance rejection, no integral action

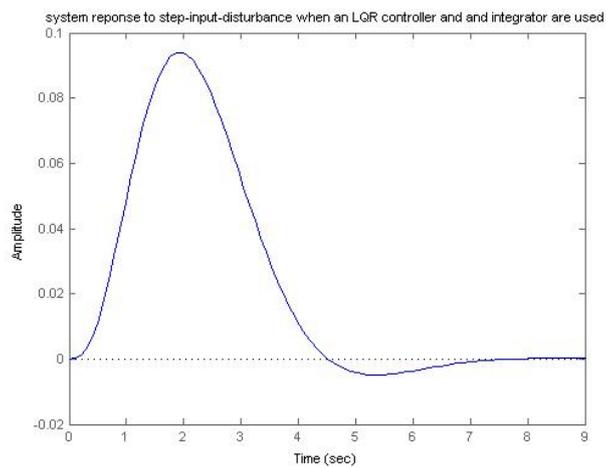


Figure F-3: Disturbance rejection, integral action

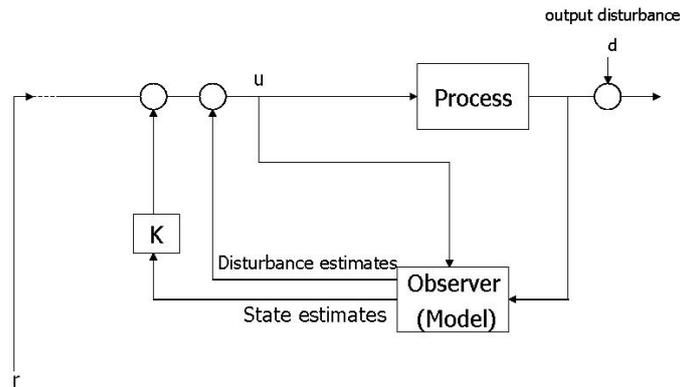


Figure F-4: system with LQG regulator

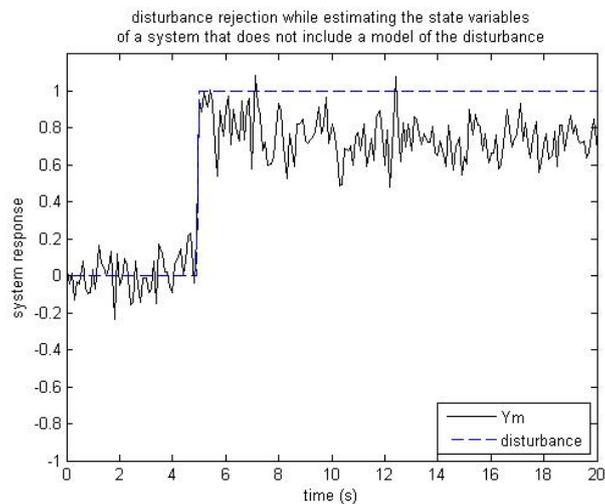


Figure F-5: non-augmented system's disturbance rejection

To solve this problem the states of the system must be estimated, as it is a stochastic model we use a Kalman filter to perform the task. The system is then controlled by an LQ regulator, which coupled with the Kalman filter gives rise to an LQG controller. The structure is as in Figure F-4.

As it can be seen in Figure F-5, the controller by itself is not capable to eliminate the effects of a step-like disturbance at the output.

To counteract the disturbance, the estimator is modified to estimate an augmented system where a model of the disturbance has been included. To proceed in accordance with the theory presented in the previous section it can be verified that the augmented system is detectable. Also, since there is only one measurable variable, only one disturbance can be added to the system. Calculating the controller gain was based on the results of Bitmead et al. [3]. The final result can be seen in Figure F-6.

A few comments on this example: the disturbance is applied 5 seconds after the simulation is initialized. The initial conditions are set to 0 for any state variable. despite the fact the

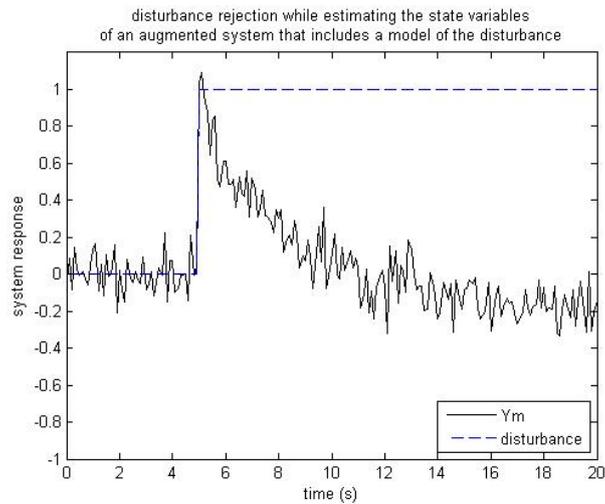


Figure F-6: augmented system's disturbance rejection

model of the disturbance is a constant in time, some uncertainty had to be assumed when the filter was used otherwise the filter would believe that it is not necessary to use information from measurements to update the estimated value of the disturbance. The filter will “think” that the prediction made by the model is perfect, since the model indicates the disturbance is a constant, and will never vary from the value assigned by the initial conditions.

CSD, nominal case vs. general case: complementary figures

In the figures below we compare the CSD of the nominal case versus that in the general case. All filters are shifted away from the desired growth rate as different error sources are introduced into the system.

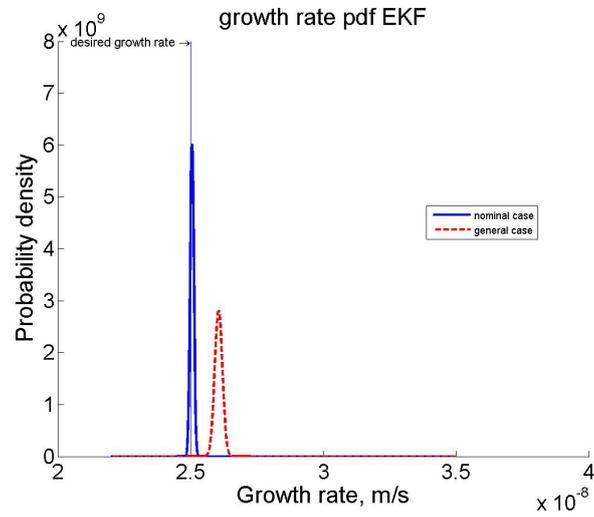


Figure G-1: CSD, nominal vs. general case

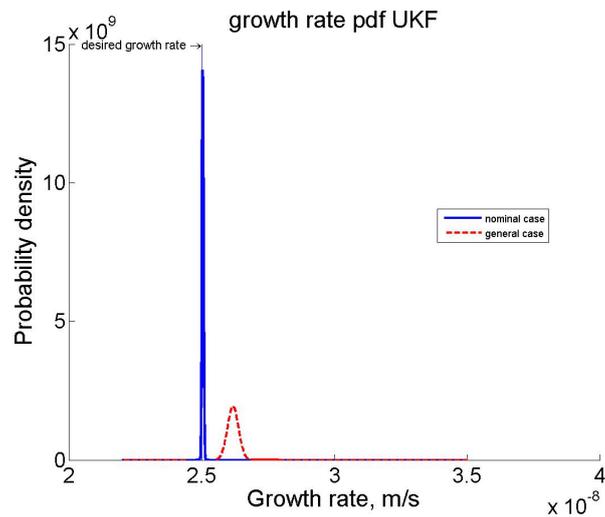


Figure G-2: CSD, nominal vs. general case

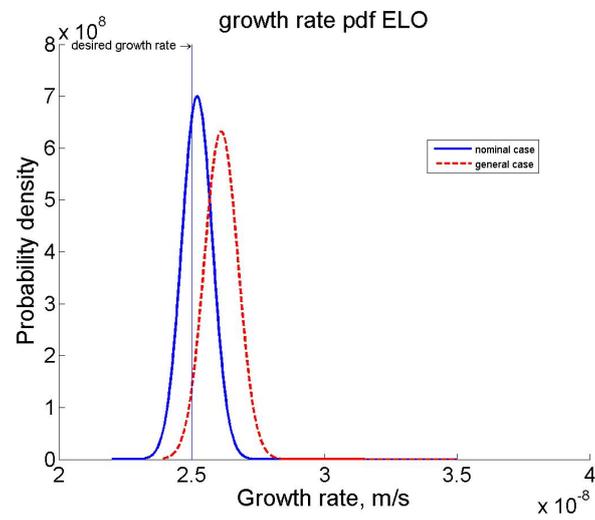


Figure G-3: CSD, nominal vs. general case

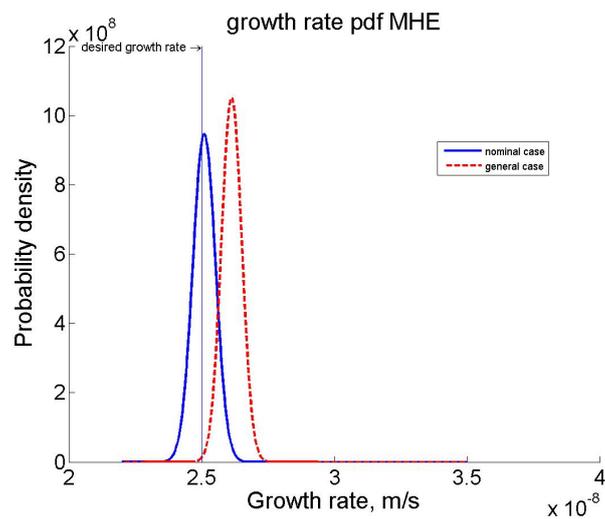


Figure G-4: CSD, nominal vs. general case

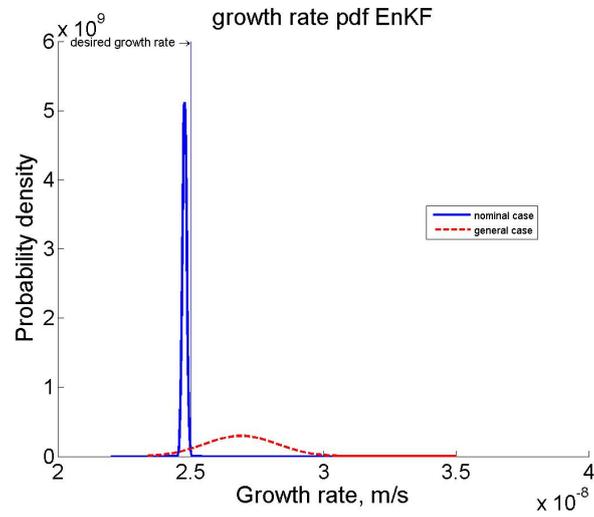


Figure G-5: CSD, nominal vs. general case

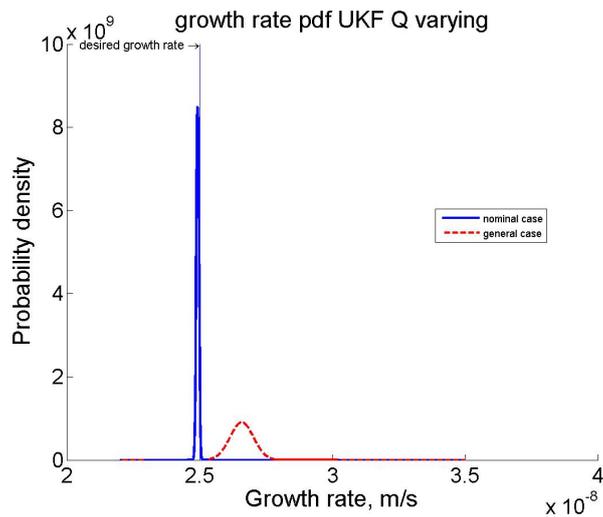


Figure G-6: CSD, nominal vs. general case

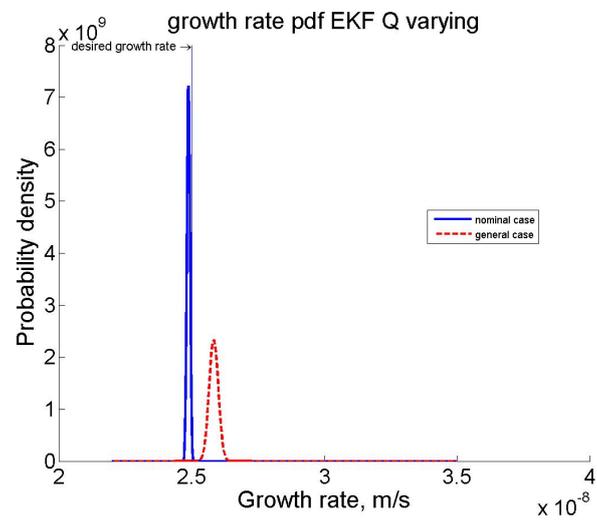


Figure G-7: CSD, nominal vs. general case

