**Master Thesis** 

# Performance indicators and monitoring of multi-stage NMPC in a real-time software framework

Jorge Daniel Mendoza Ramírez 189603



Process Dynamics and Operations Group Department of Biochemical and Chemical Engineering TU Dortmund

Prof. Dr.-Ing. Sebastian Engell

Supervising tutor: M.Sc. Alexandru Tatulea-Codrean

Dortmund, December 15, 2017

# Declaration

I herewith declare that this document has been written by me, and that this thesis is a result of my own efforts. No assistance, sources and aids other than those indicated have been used and all literature references used have been mentioned with the corresponding annotations.

Dortmund, December 15, 2017

\_\_\_\_\_

Jorge Daniel Mendoza Ramírez

# Acknowledgments

Firstly, I would like to thank my supervisor M.Sc. Alexandru Tatulea-Codrean. He recognized the importance of my research topic, defined the main scope and gave me the opportunity to work on it as well as freedom to explore some new ideas. His constant guidance and encouragement were quite valuable through the development of the thesis.

I would also like to thank Prof. Dr.-Ing. Sebastian Engell for his valuable advice, great interest shown in the project and constant support to students in the BCI department.

I express my gratitude to Dipl.-Ing. Clemens Lindscheid and M.Sc. Reinaldo Hernández, whose interest in both theoretical and practical issues in process control contributed greatly to the successful implementation of the project. Special thanks go to Rafael Saldaña for sharing his corrections on the soft constraints of the optimizer as well as to the rest of the members of my project group: Mukul Joshi, Talha Badr and Filip Chocholak, in appreciation of the great efforts put forth to implement the real-time software environment to evaluate multi-stage NMPC.

Important mention must also be made regarding funding, for which the contributions made by DAAD and the BCI department were truly useful.

Finally, I would like to thank my family and my girlfriend Stefany for the everlasting love and patience that they always show. Special thanks to my friends and to the Reich family for making my stay in Germany so pleasant and enjoyable.

### Abstract

Model predictive control (MPC) has been widely used in the process industry with reportedly remarkable economic benefits for quite a while now in spite of the fact that no systematic approach to evaluate its performance is yet available. This inconvenience not only hinders monitoring and maintenance of current implementations but also restricts the chances for new initiatives in MPC to be implemented in real plants.

In this thesis a methodology for performance assessment of such advanced controllers is exposed, taking some special consideration for applications based on multi-stage nonlinear model predictive control (multi-stage NMPC), a novel proposed formulation to handle nonlinear systems with uncertainties efficiently.

In this context, the stage cost is proposed as the fundamental measure of performance. From this definition, some performance indices can be computed by means of comparisons between the achieved stage cost in the process and the predicted stage cost from the controller.

The predicted stage cost as a performance benchmark has to be computed in general by means of Monte Carlo simulations. Nonetheless, for special cases such as linear systems with normal statistical distributions and systems with polynomial nonlinearities some efficient computation procedures can be utilized. Additionally, the use of Markov chains is explored in order to handle cases with complex distributions.

The proposed methodologies are illustrated with applications on industrial case studies. The range of possible applications is shown to be not only restricted to performance monitoring but also to cover cases such as controller selection, fault detection and training of operators.

# Table of contents

1. Introduction	1
1.1. Motivation	1
1.2. Goals	2
2. State of the art	3
2.1. Introduction	3
2.2. Literature review	3
2.2.1. Minimum variance benchmark	3
2.2.2. MPC performance assessment	4
2.2.3. Statistics of benchmarks	5
2.3. Real-time software framework to assess multi-stage NMPC	6
2.3.1. do-mpc platform	6
2.3.2. General scheme	7
2.3.3. Real-time modules	8
2.3.3.1. Process module	8
2.3.3.2. Estimator module	10
2.3.3.3. Optimizer module	11
3. Control performance assessment	
3.1. Introduction	13
3.2. Proposed methodologies	
3.2.1. Definition of performance metrics	
3.2.1.1. Theoretical background	
3.2.1.2. Main contribution	
3.2.1.2.1. Achieved performance and benchmark	13
3.2.1.2.2. Fendinance indices	10
3.2.2. Computation of benchmark statistics	
3.2.2.1 General case	20
3 2 2 1 1 Theoretical background	20
3.2.2.1.2. Implementation details	
3.2.2.2. Linear case with normal distributions	
3.2.2.2.1. Theoretical background	
3.2.2.2.2. Main contribution	

3.2	2.2.2.3. Implementation details	25
3.2.2	2.3. Nonlinear case with polynomial nonlinearities	26
3.2	2.2.3.1. Theoretical background	26
3.2	2.2.3.2. Implementation details	29
3.2.2	2.4. Nonlinear case with complex distributions	30
3.2	2.2.4.1. Theoretical background	31
3.2	2.2.4.2. Main contribution	31
3.2	2.2.4.3. Implementation details	33
3.3. Pr	oposed applications	34
3.3.1.	Controller selection	34
3.3.2.	Performance monitoring and fault detection	36
3.3.3.	Re-tuning of the controller	38
3.3.4.	Training of operators	38
		10
4. Case s	tudies	40
4.1. 00		40
4.1.1.		40
4.1.2.	Estimator module	
4.1.3.	Optimizer module	43
4.2. P€	erformance monitoring on a CSTR	45
4.2.1.	Process module	45
4.2.2.	Estimator module	47
4.2.3.	Optimizer module	48
4.2.4.	Performance assessment module	49
4.3. Pe	erformance monitoring on a polymerization reactor	50
4.3.1.	Process module	50
4.3.2.	Estimator module	52
4.3.3.	Optimizer module	53
4.3.4.	Performance assessment module	55
5. Result	s and discussion	57
5.1. Co	ontroller selection for a 4-Tank system	57
5.1.1.	Definition and computation of benchmark	57
5.1.2.	Comparison of different structures	60
5.1.3.	Analysis of dependence on the prediction horizon	64

5	5.2. Pe	rformance monitoring on a CSTR	67
	5.2.1.	Normal operation	67
	5.2.2.	Definition and computation of benchmarks	68
	5.2.3.	Fault detection	71
	5.2.4.	Analysis of scenarios	73
5	i.3. Pe	rformance monitoring on a polymerization reactor	76
	5.3.1.	Normal operation	76
	5.3.2.	Definition and computation of benchmarks	77
	5.3.3.	Fault detection	78
6.	Conclu	isions	82
7.	Future	work	85
8.	Appen	dix	86
<b>8</b> . A	Appen A. Conf	dix iguration parameters of case studies	<b> 86</b> 86
8. A	Appen A. Conf A.1. C	dix iguration parameters of case studies Controller selection for a 4-Tank system	
<b>8.</b>	Appen A. Conf A.1. C A.1.	dix iguration parameters of case studies Controller selection for a 4-Tank system I. Process module	
<b>8</b> .	<b>Appen</b> A. Conf A.1. C A.1. A.1.2	dix iguration parameters of case studies Controller selection for a 4-Tank system Process module 2. Estimator module	
<b>8.</b>	Appen A. Conf A.1. C A.1. A.1.2 A.2. F	dix iguration parameters of case studies Controller selection for a 4-Tank system Process module 2. Estimator module Performance monitoring on a CSTR	
<b>8</b> .	Appen A. Conf A.1. C A.1. A.1.2 A.2. F A.2.	dix iguration parameters of case studies Controller selection for a 4-Tank system Process module Estimator module Performance monitoring on a CSTR Process module	
<b>8</b> .	Appen A. Conf A.1. C A.1.2 A.2. F A.2.2 A.2.2	dix         riguration parameters of case studies         Controller selection for a 4-Tank system         I.       Process module         2.       Estimator module         Performance monitoring on a CSTR         I.       Process module         2.       Estimator module	
<b>8.</b>	Appen A. Conf A.1. C A.1. A.1. A.2. F A.2. A.2. A.2. A.2.	dix         iguration parameters of case studies         Controller selection for a 4-Tank system         I.       Process module         2.       Estimator module         Performance monitoring on a CSTR         I.       Process module         Performance monitoring on a CSTR         Process module         Process module	<b>86</b> 86 86 87 87 88 88 90 91
8. A	Appen A. Conf A.1. C A.1. A.1. A.2. F A.2. A.2. A.3. F A.3.	dix         iguration parameters of case studies         Controller selection for a 4-Tank system         I.       Process module         2.       Estimator module         Performance monitoring on a CSTR         I.       Process module         Process module         Performance monitoring on a CSTR         Process module         Process module         Process module         Process module         Performance monitoring on a polymerization reactor         Process module         Process module	
8. ,4	Appen A. Conf A.1. C A.1. A.1. A.1. A.1. A.1. A.1. A.1. A.1	dix         riguration parameters of case studies         Controller selection for a 4-Tank system         1. Process module         2. Estimator module         2. Estimator module         2. Process module         3. Process module         4. Process module         5. Process module         6. Process module         7. Process module         8. Process module         9. Estimator module         9. Process module         9. Process module         9. Estimator module         9. Estimator module         9. Estimator module         9. Process module         9. Estimator module         9. Process module         9. Estimator module	

# List of Figures

Figure 2.3.1: do-mpc platform modularization	7
Figure 2.3.2: Real-time do-mpc platform scheme	8
Figure 2.3.3.1: Process module flow chart	9
Figure 2.3.3.2: Estimator module flow chart	10
Figure 2.3.3.3: Optimizer module flow chart	11
Figure 3.2.1.3.1: Evaluator module in real-time scheme	18
Figure 3.2.1.3.2: Evaluator module flow chart	19
Figure 3.2.2.1.2: Monte Carlo simulation flow chart	22
Figure 3.2.2.2.3: Procedure for linear systems with normal distributions	25
Figure 3.2.2.3.2: Procedure for nonlinear case with polynomial nonlinea	rities
(PCET Toolbox)	29
Figure 3.2.2.4.2: Markov chain model example	32
Figure 3.2.2.4.3: Markov chain Monte Carlo simulation flow chart	33
Figure 3.3.1: Procedure for controller selection	35
Figure 3.3.2: Procedure for performance monitoring and fault detection	37
Figure 4.1.1: Schematic representation of 4-Tank system (Botelho et al. 207	16)
	40
Figure 4.2.1: Schematic representation of CSTR (Kvasnica et al. 2010)	45
Figure 4.3.1: Schematic representation of batch polymerization system	
(Lucía et al. 2014)	50
Figure 4.3.4: Markov chain that describes one actuator fault in batch reacto	r
	55
Figure 5.1.1.1: Time response of the 4-Tank system with unconstrained MP	C
	58
Figure 5.1.1.2: Desired benchmark for controller selection	59
Figure 5.1.2.1: Time response of the 4-Tank system with different structur	es of
multi-stage MPC	61
Figure 5.1.2.2: Comparison of achieved performance	62
Figure 5.1.2.3: $\alpha_k^d$ - Performance index for different control structures	62
Figure 5.1.2.4: $\beta_k^d$ - Performance index for different control structures	63
Figure 5.1.3.1: Time response of the 4-Tank system with different predi	ction
horizons	65

Figure 5.1.3.2: $\alpha_k^d$ - Performance index for controllers with different prediction
horizons
Figure 5.1.3.3: $\beta_k^d$ - Performance index for controllers with different prediction
horizons
Figure 5.2.1: Normal operation of the CSTR
Figure 5.2.2: Achieved performance during normal operation of the CSTR 69
Figure 5.2.3.1: Operation of the CSTR under process events
Figure 5.2.3.2: Achieved performance during abnormal operation of the CSTR
Figure 5.2.3.3: $\alpha$ -Performance indices for performance monitoring on the CSTR
Figure 5.2.3.4: $\beta$ -Performance indices for performance monitoring on the CSTR
Figure 5.2.4.1: Analysis of scenarios for normal operation with $\alpha$ -index
Figure 5.2.4.2: Analysis of scenarios for normal operation with $\beta$ -index
Figure 5.3.1: Normal operation of the polymerization reactor
Figure 5.3.2: Achieved performance during normal operation of the
polymerization reactor
Figure 5.3.3.1: Operation of the polymerization reactor under process events
Figure 5.3.3.2: Achieved performance during abnormal operation of the
polymerization reactor
Figure 5.3.3.3: $\alpha$ -Performance indices for performance monitoring on the
polymerization reactor
Figure 5.3.3.4: $\beta$ -Performance indices for performance monitoring on the
polymerization reactor
Figure 5.3.3.5: Effect of delays on the predictions

# List of Tables

Table 3.2.2.3.1: Wiener-Askey polynomial chaos	29
Table 4.1.1: Process events for 4-Tank system	42
Table 4.2.1: Process events for CSTR	47
Table 4.2.4: Statistical description of $\delta u_1$ and $\delta u_2$ for performance monitoring	on
CSTR	49
Table 4.3.1: Process events for batch polymerization reactor	52
Table 4.3.4.1: Transition probability matrix for benchmark computation	55
Table 4.3.4.2: Statistical description of $\delta u_1$ and $\delta u_2$ for performance monitor	ing
on the polymerization reactor	56
Table 5.1.1.1: Computation results for controller selection benchmark	59
Table 5.1.1.2: Computation times for controller selection benchmark	.60
Table 5.1.2.1: Controller structures for analysis	60
Table 5.1.2.2: Performance specifications for controller selection	63
Table 5.1.2.3: Aggregate performance for controller structure selection	63
Table 5.1.3.1: Controller prediction horizon for analysis	64
Table 5.1.3.2: Aggregate performance for controller prediction horizon selecti	on
	67
Table 5.2.2.1: Historical benchmark for the CSTR	69
Table 5.2.2.2: Computation results for online performance monitor	ing
benchmark $L_{jk}^{b1}$ for the CSTR	70
Table 5.2.2.3: Computation times for online performance monitoring benchm	ark
$L_{jk}^{b1}$ for the CSTR	70
Table 5.2.4.1: Averaged weights with $\alpha$ -index for the CSTR monitoring	75
Table 5.2.4.2: Averaged weights with $\beta$ -index for the CSTR monitoring	75
Table A.1.1.1: Nominal parameter values for the 4-Tank system	86
Table A.1.1.2: Initial values for the 4-Tank system	86
Table A.1.1.3: Equilibrium point for the 4-Tank system	87
Table A.1.1.4: Noise distribution for the 4-Tank system	87
Table A.1.2.1: Initial values for estimator for the 4-Tank system	87

Table A.1.2.2: Noise covariance matrices for estimator for the 4-Tank system	m
	8
Table A.2.1.1: Nominal parameter values for the CSTR         8	8
Table A.2.1.2: Initial values for the CSTR	39
Table A.2.1.3: Noise distribution for the CSTR         8	9
Table A.2.2.1: Initial values for estimator for the CSTR         90	0
Table A.2.2.2: Noise covariance matrices for estimator for the CSTR	90
Table A.3.1.1: Nominal parameter values for the batch polymerization reactor	
	<del>)</del> 1
Table A.3.1.2: Initial values for the batch polymerization reactor         92	2
Table A.3.1.3: Noise distribution for the batch polymerization reactor	)2
Table A.3.2.1: Initial values for estimator for the batch polymerization reactor	
	3
Table A.3.2.2: Noise covariance matrices for the batch polymerization reactor	
	94

### 1. Introduction

#### 1.1. Motivation

Since the late 1970's, the field of model predictive control has been in the center of attention of the process control community mainly due to its practical industrial success. This type of advanced controller is especially suitable for complex systems subject to constraints with strong interaction between variables such as those typically encountered in the chemical industry. The good performance of this controller relies heavily on a precise representation of the process, from which the optimal control strategy can be obtained in order to improve the overall operation of the plant.

Unfortunately, this requirement is hardly ever fulfilled as the presence of uncertainties not only in the formulation of the model, but also during process operation is inevitable in real plants. This fact is especially critical in the context of economic NMPC, in which typically some process variables are intended to be driven to their physical, quality or safety bounds. Therefore, in order to guarantee reliable satisfaction of constraints, robust techniques must be employed such as multi-stage NMPC (Lucía et al. 2014). Multi-stage NMPC is based on the representation of the uncertainties by means of a scenario tree with the capacity to include feedback in the predictions and has demonstrated great potential in comparison to other robust techniques.

Along with the progress in the development of MPC formulations, the field of control performance assessment has also been an important topic in the research community since the work of Åström (1970). Nonetheless, despite the efforts, the topic of performance assessment for MPC applications is still an open issue due to its complex nature and no systematic approach has been yet applied in industrial practices. Such standardized performance assessment methodologies would prove to be quite valuable to improve significantly the credibility of MPC solutions in practical terms.

For complex applications, MPC is considered mainly as a guideline controller, the outputs of which may be disregarded by the operators whenever they appear counterproductive. Provided intuitive performance indicators, field engineers may be more inclined to accept the outputs generated by the MPC controller and focus on those loops that are performing below optimal conditions with respect to a reference benchmark and may need maintenance.

On the other hand, applications on the control design stage may also be envisioned, as performance-based methodologies would assist control engineers to develop standard design procedures and would serve as a powerful tool in the analysis and selection of control structures as will be illustrated later in this work.

#### 1.2. Goals

In order to address the topic of multi-stage NMPC performance assessment, focus has been set on the following goals:

- Propose suitable performance indicators for control performance assessment. A literature review of reported methods for process monitoring is required in order to define appropriate measures of performance for multi-stage NMPC applications.
- Provide insights in the computation of the proposed benchmarks for different cases. Different numerical and statistical methods will be evaluated in order to reduce the computational complexity of the assessment.
- 3. Introduce feasible applications for the proposed methodologies in control design and performance monitoring.

# 2. State of the art

### 2.1. Introduction

In this chapter the most relevant documentation used in this project will be reviewed. Important results such as the historical minimum variance benchmark and alternative methodologies for MPC assessment will be discussed along with theoretical results on the statistics of some benchmarks. Additionally, the design and development of the real-time environment used in the project will be exposed highlighting its main components and functionality.

#### 2.2. Literature review

#### 2.2.1. Minimum variance benchmark

In 1970, Åström made the first reported contribution to the field of control performance assessment while presenting some remarks on his formulation of the so-called minimum variance strategy (Åström, 1970). This control law was formulated in the context of unconstrained SISO stochastic linear systems and had the property of optimality in the sense of output variance. Åström made the observation that the analysis of the actual output variance in comparison to the theoretical minimum would be quite convenient to evaluate if a system was optimally operated.

The minimum output variance can be computed from the process transfer function and is proved to be independent from any control action. Harris (1989) proposed to obtain such lower bound for the output variance by time series analysis of routine closed-loop data, thus providing a practical means to define a benchmark for performance assessment.

It must be clearly specified that selecting this benchmark does not necessarily mean that the minimum variance controller must be implemented in the plant. In fact, such a controller may even not be feasible to use in practice due to process constraints. Nonetheless, as a benchmark, it provides quite useful information on the current state of the controller and the plant. For example, if the assessment indicates good performance relative to this benchmark, then further tuning or changes in the control algorithm will not provide considerable improvement. In this case, further reduction of output variance may only be possible by re-engineering of the plant itself. On the other hand, if the assessment indicates poor performance then further analysis should be performed on the controller tuning, process model, robustness and constraints.

Following this direction, there have been numerous efforts to extend this benchmark to multivariable systems, all with limited success, mainly due to the complexity of the resulting algorithms and the requirement for deeper process knowledge (Harris et al. 1999). Some simplifications and modifications of the original benchmark, including user-defined requirements in order to motivate its industrial application, were more recently reported such as in Yuan, Lennox and McEwan (2009) and Liu, Huang and Wang (2011).

#### 2.2.2. MPC performance assessment

In addition to output variance monitoring, there have been other techniques reported for MPC performance assessment, which can be classified, in general terms, as model-based and model-free approaches.

Clearly, the minimum variance benchmark belongs to the model-based group and its extended form for multivariable systems is historically considered as one of the most frequently used benchmarks for MPC applications (Kadali and Huang, 2008). Prior to its selection, however, it must be evaluated if the optimization objectives are reasonably compatible and if this benchmark would provide useful and realistic performance values.

Taking this into consideration, a more natural selection for MPC applications would be the linear-quadratic Gaussian (LQG) benchmark. In this case, explicit consideration of the input variance and its range constraint is taken in order to determine the minimum output variance, thus providing a more realistic performance benchmark (Huang and Shah, 1999). The optimal curve lies on the Pareto front and the sub-optimal controller lies above this trade-off curve, from

which a performance measurement can be directly computed. In this context, Kadali and Huang (2002) and Dai and Yang (2004) developed methods to compute the required process model with subspace identification techniques in order to improve its practical applicability.

Another line of research proposes a simulation-based approach, in which the predicted design objective function from the optimizer is compared with the actual achieved objective function, computed from routine process data, in order to determine a performance index (Patwardhan, 1999). Such an index is a direct measure of the deviation from the expected performance and does not require any time series or identification procedures, thus a low performance index clearly indicating changes in the process or presence of disturbances. Along this line, Schäfer and Cinar (2002) proposed to use this index for diagnosis tasks and a historical benchmark for monitoring purposes on some case studies. More recently, in Ellis and Christofides (2014), a framework for performance monitoring in economic MPC applications is presented based on these concepts.

For the cases in which no process model is available, performance assessment based on historic data must be performed. There are several methods reported in the literature, mainly based on conventional data analysis, estimation of the closed-loop response and signal processing approaches, as summarized in Kadali and Huang (2008).

#### 2.2.3. Statistics of benchmarks

Performance assessment involves handling of stochastic variables, implying that a proper analysis of its statistics is mandatory in order to compare accurately the achieved measure with the benchmark.

Harris (1989), in his work regarding the estimation of the minimum variance benchmark by time series, analyzed the sampling properties of this estimate in order to find an appropriate number of samples required for this benchmark. Zhang and Henson (1999) analyzed statistically the ratio of the expected and actual performance in order to detect more reliably significant changes in the controller performance. A more detailed discussion on the statistics of performance indices was presented by Harris (2004). In this work, he focused on the ratio of common quadratic forms and approximations to compute confidence intervals. More recently, Zagrobelny (2014) developed a theoretical benchmark for MPC based on the analysis of a linear unconstrained system and provided closed formulas for its moments.

#### 2.3. Real-time software framework to assess multi-stage NMPC

#### 2.3.1. do-mpc platform

do-mpc is an open-source software platform designed to facilitate the rapid development of NMPC applications in a friendly and standardized environment (Lucía et al. 2014). It consists of a set of template scripts in Python with efficient implementations of NMPC and multi-stage NMPC. do-mpc relies heavily on third party software such as CasADi and IPOPT in order to compute integrations, formulate optimization problems and solve them efficiently.

do-mpc employs a modularized approach in order to provide increased flexibility for the users. The building blocks of do-mpc are structured as follows: model and problem descriptor, optimizer, estimator and simulator. Such a structure is schematically represented in Figure 2.3.1. In this platform, the simulation is performed purely offline and the data exchange between the corresponding modules is synchronized on a fixed sequence of steps for each iteration.

From the diagram, it must be clear that the optimizer sends the current control input  $u_k$  (*px*1) to the process and the estimator, the process sends the current output  $y_k$  (*mx*1) to the estimator, with which the estimator can compute and send the estimated state  $\hat{x}_{k|k}$  (*nx*1) to the optimizer.

In order to bring do-mpc one step closer to industrial practices, a real-time implementation of the platform was developed. In this environment, all the modules run asynchronously and only exchange information through a

database. The main advantage of this implementation is that it allows studying the control performance in a realistic scenario in which limited resources in computing power, computing and communication delays, access to the database and synchronization issues play a major role.



Figure 2.3.1: do-mpc platform modularization

#### 2.3.2. General scheme

In the real-time environment, each module has its own clock to read from and write to the database as shown schematically in Figure 2.3.2. The database selected for the implementation was the open-source MongoDB. MongoDB allows saving and organizing the process information as data structures in BSON documents, so that each module can access any required field in a safe and efficient manner.

From the diagram, it must be clear that the process is allowed to read the current true state  $x_k$  from its own collection, read the current control input  $u_k$  from the optimizer's collection and write the next true state  $x_{k+1}$  and the next process output  $y_{k+1}$  to its own collection. The optimizer is allowed to read from the estimator's collection the current estimated state  $\hat{x}_{k|k-1}$  to compute and write the next control input  $u_{k+1}$  to its own collection. The estimator is allowed to read from the next control input  $u_{k+1}$  to its own collection. The estimator is allowed to read from ead from its own collection the current estimated state  $\hat{x}_{k|k-1}$ , the current process output  $y_k$  from the process's collection and the current control input  $u_k$ 

from the optimizer's collection to compute and write the next estimated state  $\hat{x}_{k+1|k}$  to its own collection. In the next section, further details on the operation of these modules will be provided.



Figure 2.3.2: Real-time do-mpc platform scheme

#### 2.3.3. Real-time modules

#### 2.3.3.1. Process module

The process module was written in MATLAB and replaces the original simulator module from do-mpc. The main operations performed in this module are presented in Figure 2.3.3.1.

As shown, this module works with an Excel sheet, in which the nominal value of the process parameters, noise variances, initial conditions and configuration of process events, such as presence of disturbances in a predetermined time interval, are defined. To complete the initialization phase, the communication with the database is established and the initial true state, process output and control input are written. After initialization, the periodic execution of the module is scheduled by means of MATLAB's timer functionality. At each time period the current true state  $x_k$  and control input  $u_k$  are read from the database. The process module then evaluates if at the current time some event must be activated and then proceeds with the integration. The process noise  $w_k$  is sampled from a normal distribution and kept constant during the integration. The measurement noise  $v_k$  is also sampled from a normal distribution and added to the output after the integration.

Finally, the next true state  $x_{k+1}$  and process output  $y_{k+1}$  are written to the database. The loop repeats itself until the number of iterations is completed. In the last iteration, a synchronization signal is sent to the database in order to stop the execution of the other modules.



Figure 2.3.3.1: Process module flow chart

#### 2.3.3.2. Estimator module

The estimator module is required in order to reproduce a typical situation in which only a limited amount of process measurements is available and a state-feedback scheme must be employed for control purposes. The estimator module was written in Python based on the well-known Extended Kalman Filter. The main operations performed in this module are presented in Figure 2.3.3.2.



Figure 2.3.3.2: Estimator module flow chart

In the initialization step the noise covariance matrices are specified, along with the initial value of the estimated states and a nominal model for the process. In this step, the communication with MongoDB is also established by means of the pymongo driver. For the periodic execution of the module, a timer code was implemented based on a Python's generator. The generator provides the next instant in which the estimator should run, taking into consideration the desired time period and the previous computing time, so that the estimator can be triggered at specific time points. At each time period, the update step is performed taking the current process output  $y_k$  from the database to perform the correction on the current estimated state  $\hat{x}_{k|k-1}$  to obtain  $\hat{x}_{k|k}$  by means of the well-known Kalman gain. Following, the prediction step is performed taking the corrected estimated state  $\hat{x}_{k|k}$  from the previous step and the current control input  $u_k$  from the database in order to predict the next estimated state  $\hat{x}_{k+1|k}$  based on the nominal model. After the prediction, the next estimated state is written to the database. The operation of the estimator concludes when it receives a stop signal from the process.

#### 2.3.3.3. Optimizer module

The optimization module is arguably the most important block of do-mpc as it represents its main line of study. This module was written in Python based on do-mpc version 1.0. The main operations performed in this module are presented in Figure 2.3.3.3.



Figure 2.3.3.3: Optimizer module flow chart

In the initialization step the communication with MongoDB is established and the optimization problem is set up. This includes decision variables, cost function, bound constraints in the prediction horizon and uncertainty values for multi-stage NMPC. The timer for periodic execution is implemented similarly as in the case of the estimator module.

In order to compensate for the feedback delay, an additional prediction step is performed from the current estimated state  $\hat{x}_{k|k-1}$  to obtain  $\hat{x}_{k+1|k-1}$ , which is set as the initial condition in the prediction horizon. Once this step is completed, the nonlinear optimization problem (NLP) is solved by means of the IPOPT tool. The exit flag of the optimizer is taken into consideration so that only feasible solutions are written to the database.

Finally, the optimal solution of the current problem is used to prepare the next optimization problem. The operation of the optimizer concludes when it receives a stop signal from the process.

### 3. Control performance assessment

#### 3.1. Introduction

In this chapter some methodologies for control performance assessment are proposed along with some possible applications. Each main part will be preceded by a theoretical background and followed by an implementation procedure.

#### 3.2. Proposed methodologies

#### 3.2.1. Definition of performance metrics

In this section the basic performance measurement is defined along with the reference benchmark. Additionally, some performance indices are proposed in order to facilitate a comparison between them. The procedure developed focuses especially on performance assessment for multi-stage NMPC applications, for which a short review is included in the theoretical background.

#### 3.2.1.1. Theoretical background

The optimization problem for multi-stage NMPC is reviewed as presented in Lucía et al. (2014) in order to introduce some fundamentals, terminology and notations that will be useful in the main part.

Multi-stage NMPC is a robust NMPC approach based on a scenario tree construction that describes the evolution of uncertainties and their effect on the process. This tree is constructed assuming that, at each point in the prediction horizon, some discrete set of uncertainty values will be realized in the process, the effect of which is a branching in the process prediction. Future inputs are allowed to depend on previous values of the uncertainties, implicitly incorporating feedback into the scheme. The uncertain nonlinear system is represented as:

$$x_{k+1}^{j} = f\left(x_{k}^{p(j)}, u_{k}^{j}, d_{k}^{r(j)}\right),$$
 (1)

where the state  $x_{k+1}^{j}$  at time k+1 in branch j is a function of the state  $x_{k}^{p(j)}$  at time k in the previous branch, the control input  $u_{k}^{j}$  at time k in branch j and the realization of the uncertainty  $d_{k}^{r(j)}$  for that branch at time k. The set of all possible indices (j, k) in the tree construction will be denoted as I.

Each path from the first node  $x_0$  to a terminal one  $x_K^i$  is called a scenario  $S_i$ , where K is the prediction horizon. From this explanation, it must be clear that the number of scenarios N grows significantly fast with the prediction horizon, the number of uncertainties and the set of uncertainty values considered. This practical problem can be handled by assuming that the uncertainties remain constant after a time point, which is called the robust horizon.

The optimization problem is written in general as follows:

$$\min_{\boldsymbol{x}_{k}^{j}, \boldsymbol{u}_{k}^{j} \forall (j,k) \in I} \left( \sum_{i=1}^{N} \left( \omega_{i} J_{i} \left( \boldsymbol{x}_{k+1}^{j}, \boldsymbol{u}_{k}^{j} \right) \right)^{\alpha} \right)^{1/\alpha}$$
(2a)

subject to:

$$x_{k+1}^{j} = f\left(x_{k}^{p(j)}, u_{k}^{j}, d_{k}^{r(j)}\right), \forall (j, k+1) \in I,$$
(2b)

$$g(\boldsymbol{x}_{k}^{j}, \boldsymbol{u}_{k}^{j}) \leq 0, \forall (j, k) \in I,$$
(2c)

$$u_{k}^{j} = u_{k}^{l} \text{ if } x_{k}^{p(j)} = x_{k}^{p(l)}, \forall (j,k), (l,k) \in I,$$
(2d)

where (2a) is the cost function, (2b) the process model, (2c) represents general nonlinear constraints on the states and inputs and (2d) a non-anticipativity constraint.

The cost of each scenario  $S_i$  with weight  $\omega_i$  is denoted as  $J_i(x_{k+1}^j, u_k^j)$  and is defined as:

$$J_{i}(\boldsymbol{x}_{k+1}^{j}, \boldsymbol{u}_{k}^{j}) = \sum_{k=0}^{K-1} L(\boldsymbol{x}_{k+1}^{j}, \boldsymbol{u}_{k}^{j}), \forall \boldsymbol{x}_{k+1}^{j}, \boldsymbol{u}_{k}^{j} \in S_{i},$$
(3)

where  $L(x_{k+1}^{j}, u_{k}^{j})$  is a general stage cost.

#### 3.2.1.2. Main contribution

#### 3.2.1.2.1. Achieved performance and benchmark

Given the stage cost  $L(x_{k+1}^{j}, u_{k}^{j})$ , the measurement of achieved performance is defined in general as:

$$L_k^a = L\big(\widehat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{u}_k\big) \tag{4}$$

Alternatively, for many practical applications, in which the stage cost is typically expressed exclusively in terms of measured states and control inputs, the following definition can also be provided:

$$L_k^a = L_s(\boldsymbol{y}_k, \boldsymbol{u}_k), \tag{5}$$

where  $L(\mathbf{x}_k, \mathbf{u}_k) = L_s(\mathbf{y}_k, \mathbf{u}_k)$ .

For monitoring purposes, this performance measurement will be compared to a model-based benchmark defined as:

$$L_{jk}^{b} = L\left(\widehat{x}_{k|k_{0}}^{j}, \widehat{u}_{k|k_{0}}^{n1(j)}\right), \forall (j, k - k_{0}) \in I, k > k_{0},$$
(6)

where state predictions  $\hat{x}_{k|k_0}^j$  are computed from time  $k_0$  using (2b) and the optimizer predicted inputs  $\hat{u}_{k|k_0}^{n1(j)}$ . The superscript n1(j) represents one of the branches that emerge from node  $\hat{x}_{k|k_0}^j$ .

This benchmark represents the achievable performance of the system by design and is in general, a stochastic variable if noise covariance information is

included in the predictions and if the predicted uncertainty realizations  $\hat{d}_{k|k_0}^{r(j)}$  are sampled from probability distributions.

Depending on the selection of  $k_0$ , it is possible to define this benchmark based on one-step or multi-step predictions up to the prediction horizon:

$$L_{jk}^{b1} = L\left(\widehat{\boldsymbol{x}}_{k|k-1}^{j}, \widehat{\boldsymbol{u}}_{k|k-1}^{n1(j)}\right), \forall (j, 1) \in I,$$

$$(7)$$

$$L_{jk}^{bK} = L\left(\widehat{x}_{k|k-(K-1)}^{j}, \widehat{u}_{k|k-(K-1)}^{n1(j)}\right), \forall (j, K-1) \in I,$$
(8)

in order to evaluate the short and long term predictive capacity of the optimizer, respectively. Taking into consideration the size problem of the scenario tree, it is worth providing expressions for the benchmark for the simple case in which the robust horizon equals 1:

$$L_{jk}^{b1} = L(\hat{x}_{k|k-1}^{j}, \hat{u}_{k|k-1}^{j}), \forall j = 1, ..., N$$
(9)

$$L_{jk}^{bK} = L\left(\widehat{\boldsymbol{x}}_{\boldsymbol{k}|\boldsymbol{k}-(\boldsymbol{K}-1)}^{j}, \widehat{\boldsymbol{u}}_{\boldsymbol{k}|\boldsymbol{k}-(\boldsymbol{K}-1)}^{j}\right), \forall j = 1, \dots, N$$
(10)

#### 3.2.1.2.2. Performance indices

Once the achieved performance measurement and the benchmark have been defined, some performance indices can be computed in order to establish a systematic comparison between them.

Following the work of Patwardhan (1999), the  $\alpha$ -index is defined as the ratio of the expected value of the benchmark and the achieved stage cost:

$$\alpha_{jk}^b = \frac{E[L_{jk}^b]}{L_k^a} \tag{11}$$

The  $\alpha$ -index evaluates the proximity of the achieved performance to the expected value of the benchmark. This definition represents an instantaneous performance indicator prone to noisy readings. Therefore, it is advisable to define also time-averaged versions of the  $\alpha$ -index such as follows:

$$\alpha_{jk}^{b1} = \frac{\sum_{k'=k-N_W}^k E[L_{jk'}^{b1}]}{\sum_{k'=k-N_W}^k L_{k'}^a},$$
(12)

$$\alpha_{jk}^{bK} = \frac{\sum_{k'=k-N_K}^k E[L_{j'kk'}^{bK}]}{\sum_{k'=k-N_K}^k L_{k'}^a},$$
(13)

where  $N_w$  is a user-defined time window. The auxiliary variable  $L_{j'kk'}^{bK}$  is defined:  $L_{j'kk'}^{bK} = L\left(\widehat{\mathbf{x}}_{k'|k-(K-1)}^{j'}, \widehat{\mathbf{u}}_{k'|k-(K-1)}^{n1(j')}\right), \forall (j', k' - (k - (K - 1))) \in I, k' > k - (K - 1)$ so that with  $N_K = K - 2$ , the whole prediction horizon can be chosen as an averaging window for the multi-step case.

Additionally to the  $\alpha$ -index, it is possible to define another performance index in which the distribution spread of the benchmark is explicitly taken into account. This spread may be considered in terms of quantiles or a certain number of standard deviations around the expected value. The  $\beta$ -index is defined as:

$$\beta_{jk}^{b} = \begin{cases} \frac{q_{h}[L_{jk}^{b}]}{L_{k}^{a}} & L_{k}^{a} > q_{h}[L_{jk}^{b}] \\ 1 & q_{l}[L_{jk}^{b}] \le L_{k}^{a} \le q_{h}[L_{jk}^{b}], \\ \frac{q_{l}[L_{jk}^{b}]}{L_{k}^{a}} & L_{k}^{a} < q_{l}[L_{jk}^{b}] \end{cases}$$
(14)

where  $q_l[.]$  and  $q_h[.]$  are estimations of lower and upper bounds for a probability distribution, respectively. Analogously, time-averaged versions of the  $\beta$ -index can be formulated as follows:

$$\beta_{jk}^{b1} = \begin{cases} \frac{\sum_{k'=k-N_{w}}^{k} q_{h}[L_{jk'}^{b1}]}{\sum_{k'=k-N_{w}}^{k} L_{k'}^{a}} & \sum_{k'=k-N_{w}}^{k} L_{k'}^{a} > \sum_{k'=k-N_{w}}^{k} q_{h}[L_{jk'}^{b1}] \\ 1 & \sum_{k'=k-N_{w}}^{k} q_{l}[L_{jk'}^{b1}] \le \sum_{k'=k-N_{w}}^{k} L_{k'}^{a} \le \sum_{k'=k-N_{w}}^{k} q_{h}[L_{jk'}^{b1}] & (15) \\ \frac{\sum_{k'=k-N_{w}}^{k} q_{l}[L_{jk'}^{b1}]}{\sum_{k'=k-N_{w}}^{k} L_{k'}^{a}} & \sum_{k'=k-N_{w}}^{k} L_{k'}^{a} < \sum_{k'=k-N_{w}}^{k} q_{l}[L_{jk'}^{b1}] \end{cases}$$

$$\beta_{jk}^{bK} = \begin{cases} \frac{\sum_{k'=k-N_{K}}^{k} q_{h} \left[ L_{jkk'}^{bK} \right]}{\sum_{k'=k-N_{K}}^{k} L_{k'}^{a}} & \sum_{k'=k-N_{K}}^{k} L_{k'}^{a} > \sum_{k'=k-N_{K}}^{k} q_{h} \left[ L_{jkk'}^{bK} \right] \\ 1 & \sum_{k'=k-N_{K}}^{k} q_{l} \left[ L_{jkk'}^{bK} \right] \le \sum_{k'=k-N_{K}}^{k} L_{k'}^{a} \le \sum_{k'=k-N_{K}}^{k} q_{h} \left[ L_{jkk'}^{bK} \right] & (16) \\ \frac{\sum_{k'=k-N_{K}}^{k} q_{l} \left[ L_{jkk'}^{bK} \right]}{\sum_{k'=k-N_{K}}^{k} L_{k'}^{a}} & \sum_{k'=k-N_{K}}^{k} L_{k'}^{a} < \sum_{k'=k-N_{K}}^{k} q_{l} \left[ L_{jkk'}^{bK} \right] \end{cases}$$

#### 3.2.1.3. Implementation details

In addition to the modules presented in Figure 2.3.2, an evaluator module is proposed in order to compute periodically performance indices based on process data and results from the optimizer and the estimator as explained in the previous section. The scheme is represented graphically in Figure 3.2.1.3.1. As shown in the figure, the predicted inputs  $\hat{u}_{k'|k}^{j} \forall k' = k, ..., k + K - 1$  for all scenarios have to be stored in the database for further analysis in the evaluator module.



Figure 3.2.1.3.1: Evaluator module in real-time scheme

The basic steps to implement the performance assessment in the evaluator module are summarized as a flowchart in Figure 3.2.1.3.2.



Figure 3.2.1.3.2: Evaluator module flow chart

At each time step, according to the definition of achieved performance adopted,  $\hat{x}_{k|k-1}$  (or  $y_k$ ) and  $u_k$  are read from the database to compute  $L_k^a$  with (4) (or (5)). A buffer of size  $\max(N_w, N_K)$  containing past values of  $L_k^a$  is required in order to compute time-averaged performance indices as explained in last section.

The next step is to simulate the system to obtain the predicted states  $\hat{x}_{k'|k}^{j}$ , for all scenarios, integrating the process model (1) from the initial condition  $\hat{x}_{k|k-1}$ with control input sequences  $\hat{u}_{k'|k}^{j}$ . The lack of precision in the initial condition, which can be measured by the estimator's error covariance matrix  $P_{x,k|k-1}$ , the effect of noise represented by the covariance matrices  $Q_{w,k}$  and  $R_{v,k}$ , for process and measurement noise, respectively, and the realization of the uncertainties  $\hat{d}_{k'|k}^{r(j)}$  should be taken into consideration. Samples of the benchmark are computed with (6) considering the variability of the uncertain parameters. From the samples, some statistics can be estimated such as moments and quantiles. These results should be conveniently stored in a buffer for future performance evaluations. Finally, retrieving the necessary elements from the buffers, the different performance indices can be computed with (11)-(16).

#### 3.2.2. Computation of benchmark statistics

The main idea behind the computation of the benchmark is to perform simulations. In the general case, a number of Monte Carlo simulations are required in order to estimate properly its statistics. Nonetheless, some special cases are worth analyzing in order to produce more computationally efficient methodologies. For uncertainties with complex distributions, a methodology based on Markov chains is presented.

#### 3.2.2.1. General case

#### 3.2.2.1.1. Theoretical background

A short review of Monte Carlo methods for systems simulation is presented as in Brandimarte (2014).

Monte Carlo methods are a broad class of computational algorithms used to simulate systems affected by randomness, in which typically, several random scenarios are generated numerically and suitable statistics are estimated from the results. These conceptually simple methods provide flexible and valuable tools to perform simulations of complex systems.

A typical Monte Carlo simulation consists of the following steps:

1. Generate independent samples from stochastic variables. This requires efficient computation algorithms and a thorough analysis on the nature of

the uncertain parameters in order to develop realistic probability distributions.

- Simulate the system. For this step a detailed mathematical model with correctly estimated parameters is mandatory. Efficient methods to compute numeric integrals are also in order.
- 3. The outcomes are collected in order to compute appropriate statistical measurements.

Following this methodology, it is possible to compute the benchmark in the general case as will be explained in the following section.

#### 3.2.2.1.2. Implementation details

The Monte Carlo simulation is implemented as explained in the previous section and represented graphically in Figure 3.2.2.1.2.

The procedure is initialized by reading from the database the initial condition  $\hat{x}_{k|k-1}$ , the future control inputs  $\hat{u}_{k'|k}^{j}$ , the estimator's error covariance matrix  $P_{x,k|k-1}$ , the noise covariance matrices  $Q_{w,k}$  and  $R_{v,k}$ , as well as the realization of the uncertainties  $\hat{d}_{k'|k}^{r(j)}$ .

At time t = 0, a sample is taken from the initial condition's multivariable normal distribution  $N_n(\hat{x}_{0|-1}, P_{x,0|-1})$ . Analogously, at each time step, values for the noises are sampled from their distributions  $N_n(0, Q_{w,k})$  and  $N_m(0, R_{v,k})$ . The realization of uncertainties can also be considered as a stochastic variable if a proper probability distribution is assigned.

The simulation is performed from time t = 0 to  $t = t_{end} = (K - 1)T$ , with time steps equal to the optimizer's period *T*. At each time step, the stage cost is computed and stored in a buffer.

This complete procedure is iterated  $N_{mc}$  times for each scenario. For each time step, estimations for the expected value and lower and upper bounds for the stage cost's distribution are computed and stored in a buffer.



Figure 3.2.2.1.2: Monte Carlo simulation flow chart

#### 3.2.2.2. Linear case with normal distributions

The case of linear systems with normal distributions for uncertainties is worth analyzing as it allows formulating closed-form statistics for common benchmarks. The procedure is based on important properties of normal distributions, which are reviewed in the theoretical section.

#### 3.2.2.2.1. Theoretical background

Linear systems have been widely studied in the literature and their application in industrial control systems is commonplace. Linear models are commonly used in practice due to their simplicity, provided that they can represent correctly a process in a required range of operation. This is specially the case in control systems in which the objective is to keep the process as close as possible to some desired operating point.

Furthermore, it is a common practice to consider noise as the result of several unmodeled random effects. With this justification, the central limit theorem allows to employ a normal distribution for noises, from which a distribution for all the states in the process can be computed (Zagrobelny, 2014). This result can be obtained from exploiting certain properties of the normal distribution.

As presented in Tong (1990), it can be proved that the family of multivariate normal distributions is closed under linear transformations and combinations of random variables. Furthermore, it can be proved that if  $X_1 \sim N_n(\mu_1, \Sigma_1)$  and  $X_2 \sim N_n(\mu_2, \Sigma_2)$  are independent normally-distributed variables, then

$$Y = C_1 X_1 + C_2 X_2 \sim N_n (C_1 \mu_1 + C_2 \mu_2, C_1 \Sigma_1 C_1' + C_2 \Sigma_2 C_2'),$$
(17)

from which the mean and variance of a linear combination can be computed.

Another quite useful result is presented in Mathai (1992) regarding the statistics of quadratic forms. It is shown that if  $X \sim N_n(\mu, \Sigma)$ , Q(X) = X'AX, A = A', then

$$E(Q(X)) = tr(A\Sigma) + \mu' A\mu, \qquad (18)$$

$$Var(Q(\mathbf{X})) = 2tr(A\Sigma)^2 + 4\boldsymbol{\mu}' A\Sigma A\boldsymbol{\mu},$$
(19)

from which, the expected value and variance of a quadratic stage cost can be exactly computed.

#### 3.2.2.2.2. Main contribution

Given a time-discrete linear system:

$$x_{k+1}^{j} = A_{d,k} x_{k}^{p(j)} + B_{d,k} u_{k}^{j} + D_{d,k} d_{k}^{r(j)} + w_{k},$$
(20a)

$$\boldsymbol{y}_{k}^{j} = C_{k}\boldsymbol{x}_{k}^{j} + \boldsymbol{v}_{k}, \tag{20b}$$

with deterministic control inputs  $\boldsymbol{u}_{k}^{j}$  and normally-distributed stochastic variables  $\boldsymbol{w}_{k} \sim N_{n}(\boldsymbol{0}, Q_{w,k}), \ \boldsymbol{v}_{k} \sim N_{m}(\boldsymbol{0}, R_{v,k})$  and  $\boldsymbol{d}_{k}^{r(j)} \sim N_{d}(\boldsymbol{\mu}_{d,k}^{r(j)}, \boldsymbol{\Sigma}_{d,k}^{r(j)})$ . If  $\boldsymbol{x}_{0} \sim N_{n}(\boldsymbol{\mu}_{x,0}, \boldsymbol{\Sigma}_{x,0})$ , then, by (17):

$$\boldsymbol{x}_{k}^{j} \sim N_{n} \left( \boldsymbol{\mu}_{x,k}^{j}, \boldsymbol{\Sigma}_{x,k}^{j} \right)$$
(21)

and

$$\boldsymbol{y}_{\boldsymbol{k}}^{j} \sim N_{m} \left( \boldsymbol{\mu}_{\boldsymbol{y},\boldsymbol{k}}^{j}, \boldsymbol{\Sigma}_{\boldsymbol{y},\boldsymbol{k}}^{j} \right),$$
(22)

where

$$\boldsymbol{\mu}_{x,k+1}^{j} = A_{d,k} \boldsymbol{\mu}_{x,k}^{p(j)} + B_{d,k} \boldsymbol{u}_{k}^{j} + D_{d,k} \boldsymbol{\mu}_{d,k}^{r(j)}$$
(23)

$$\Sigma_{x,k+1}^{j} = A_{d,k} \Sigma_{x,k}^{p(j)} A_{d,k}' + D_{d,k} \Sigma_{d,k}^{r(j)} D_{d,k}' + Q_{w,k}$$
(24)

$$\boldsymbol{\mu}_{\boldsymbol{y},\boldsymbol{k}}^{j} = C_{\boldsymbol{k}} \boldsymbol{\mu}_{\boldsymbol{x},\boldsymbol{k}}^{j} \tag{25}$$

$$\Sigma_{y,k}^{j} = C_k \Sigma_{x,k}^{j} C_k' + R_{v,k}$$
(26)

Equations (23)-(26) provide a theoretical procedure to compute the expected value and variance of all the states and outputs of the system by means of difference equations. It must be clear that statistics for linear stage costs can also be computed directly by this method.

Furthermore, for a typical quadratic stage cost in set point tracking problems:  $L(\mathbf{y}_k) = (\mathbf{y}_k - \mathbf{y}_{sp,k})' Q_{y,k} (\mathbf{y}_k - \mathbf{y}_{sp,k}), \text{ the following result holds by (18) and (19):}$ 

$$E\left(L(\boldsymbol{y}_{\boldsymbol{k}}^{j})\right) = tr(Q_{\boldsymbol{y},\boldsymbol{k}}\boldsymbol{\Sigma}_{\boldsymbol{y},\boldsymbol{k}}^{j}) + \left(\boldsymbol{\mu}_{\boldsymbol{y},\boldsymbol{k}}^{j} - \boldsymbol{y}_{\boldsymbol{s}\boldsymbol{p},\boldsymbol{k}}\right)'Q_{\boldsymbol{y},\boldsymbol{k}}\left(\boldsymbol{\mu}_{\boldsymbol{y},\boldsymbol{k}}^{j} - \boldsymbol{y}_{\boldsymbol{s}\boldsymbol{p},\boldsymbol{k}}\right)$$
(27)

$$Var\left(L(\boldsymbol{y}_{\boldsymbol{k}}^{\boldsymbol{j}})\right) = 2tr\left(Q_{y,k}\boldsymbol{\Sigma}_{y,k}^{\boldsymbol{j}}\right)^{2} + 4\left(\boldsymbol{\mu}_{y,\boldsymbol{k}}^{\boldsymbol{j}} - \boldsymbol{y}_{\boldsymbol{sp},\boldsymbol{k}}\right)'Q_{y,k}\boldsymbol{\Sigma}_{y,k}^{\boldsymbol{j}}Q_{y,k}\left(\boldsymbol{\mu}_{y,\boldsymbol{k}}^{\boldsymbol{j}} - \boldsymbol{y}_{\boldsymbol{sp},\boldsymbol{k}}\right)$$
(28)

Therefore, the presented procedure allows efficient computation of the benchmark's statistics for the linear case with normal distributions.

#### 3.2.2.2.3. Implementation details

The proposed method for linear systems with normal distributions is implemented as explained in the previous section and represented graphically in Figure 3.2.2.2.3.



Figure 3.2.2.2.3: Procedure for linear systems with normal distributions

After reading from the database the required information, the difference equations (23)-(26) are solved in order to compute the expected value and variance of states and outputs. It must be kept into consideration that  $\hat{d}_{k'|k}^{r(j)}$  has to be modeled as a normally-distributed variable in the evaluator module. The simulation is performed for each scenario from time t = 0 to t = (K - 1)T, with time steps equal to the optimizer's period *T*. At each time step, the results obtained with (27) and (28) should be stored in a buffer for future evaluations.
## 3.2.2.3. Nonlinear case with polynomial nonlinearities

The theory of polynomial chaos expansion (Wiener, 1938) can be efficiently applied to estimate the benchmark statistics in the context of nonlinear systems, for which a review of its fundamentals will be explained in the theoretical section, as detailed in Xiu and Karniadakis (2002) and in Streif et al. (2014).

#### 3.2.2.3.1. Theoretical background

In 1938, Wiener presented his theory on homogeneous chaos expansion in order to represent general second-order random processes by means of Hermite polynomials and combinations of them, which he called Hermite-chaos.

The Hermite polynomials have multiple interesting properties, among which, orthogonality is especially important in this context. These polynomials are orthogonal with respect to a weighting function, which has the same form as a Gaussian probability density function:

$$\langle \mathbf{H}_{i}(\mathbf{x}),\mathbf{H}_{j}(\mathbf{x})\rangle = \int_{-\infty}^{\infty} e^{-\frac{\mathbf{x}^{2}}{2}} \mathbf{H}_{i}(\mathbf{x})\mathbf{H}_{j}(\mathbf{x})d\mathbf{x} = \sqrt{2\pi} \,i!\,\delta_{ij},\tag{29}$$

where  $H_i(x)$  is the Hermite polynomial of degree i and  $\delta_{ij}$  is the Kronecker delta. Following Wiener's formulation, the Hermite-chaos expansion for a general second-order random process can be represented as follows:

$$x(\boldsymbol{\xi}) = a_0 H_0 + \sum_{i_1=1}^n a_{i_1} H_1(\xi_{i_1}) + \sum_{i_1=1}^n \sum_{i_2=1}^{i_1} a_{i_1 i_2} H_2(\xi_{i_1}, \xi_{i_2}) + \sum_{i_1=1}^n \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} H_3(\xi_{i_1}, \xi_{i_2}, \xi_{i_3}) + \cdots,$$
(30)

where  $H_m(\xi_{i_1}, ..., \xi_{i_n})$  denotes the Hermite-chaos of order m in the variable  $\xi$ , which is composed of n independent Gaussian variables. The coefficients  $a_{i_1...i_m}$  describe the influence of each Hermite-chaos on the stochastic variable x.

For computational purposes, this infinite summation must be truncated at some order p and is conveniently represented as:

$$x(\boldsymbol{\xi}) \approx \sum_{j=0}^{P} \hat{a}_j \, \psi_j(\boldsymbol{\xi}), \tag{31}$$

where there is a one-to-one correspondence between the functions  $H_m(\xi_{i_1}, ..., \xi_{i_n})$  and  $\psi_j(\xi)$ . The number of terms of the summation can be proven to be given by:

$$P + 1 = \frac{(n+p)!}{n!p!} \tag{32}$$

From (32), it is clear that the number of coefficients grows significantly with the number of uncertainties and the maximum order of the Hermite-chaos.

With such proper considerations, this method can be applied to efficiently solve stochastic differential equations as follows. Given a process model:

$$\frac{dx}{dt} = f(x, u, d), \tag{33}$$

with deterministic inputs u and stochastic uncertainties d. The state x and the uncertainty d can be represented by the following expansion:

$$x(\xi, t) = \sum_{j=0}^{P} \hat{x}_{j}(t) \psi_{j}(\xi)$$
(34a)

$$d(\boldsymbol{\xi}, \mathbf{t}) = \sum_{j=0}^{P} \hat{d}_{j}(t) \psi_{j}(\boldsymbol{\xi})$$
(34b)

Normally, the uncertainty's distribution is known, from which the coefficients  $\hat{d}_j$  can be directly computed. Replacing (34) in (33), the following ordinary differential equation (ODE) system for the coefficients  $\hat{x}_j$  is obtained:

$$\sum_{j=0}^{P} \frac{d\hat{x}_{j}(t)}{dt} \psi_{j}(\xi) = f\left(\sum_{j=0}^{P} \hat{x}_{j}(t) \psi_{j}(\xi), u, \sum_{j=0}^{P} \hat{d}_{j}(t) \psi_{j}(\xi)\right)$$
(35)

This system can be further simplified by means of a Galerkin projection method, by taking the inner product on both sides and using (29):

$$\frac{d\hat{x}_k(t)}{dt} = \frac{\langle f\left(\sum_{j=0}^P \hat{x}_j(t)\psi_j(\xi), u, \sum_{j=0}^P \hat{d}_j(t)\psi_j(\xi)\right), \psi_k(\xi)\rangle}{\sqrt{2\pi}\,k!} \quad \forall k = 0, 1, \dots, P$$
(36)

In order to compute the coefficients  $\hat{x}_k(t)$  from (36), the expansion for the initial state should be obtained first given its distribution. For the integration, conventional ODE solvers can be used such as Runge-Kutta methods. It must be considered that the right-hand side integrals in (36), result of the Galerkin projection, are in general quite cumbersome to compute, except for cases in which the nonlinearities in *f* are polynomials, as explained in Streif et al. (2014).

Once the coefficients  $\hat{x}_k(t)$  have been obtained, the expected value and variance of  $x(\xi, t)$  can be derived as follows:

$$E(x(\boldsymbol{\xi}, \mathbf{t})) = E\left(\sum_{k=0}^{P} \hat{x}_{k}(t) \psi_{k}(\boldsymbol{\xi})\right) = \sum_{k=0}^{P} \hat{x}_{k}(t) E\left(\psi_{k}(\boldsymbol{\xi})\right)$$
(37)

$$Var(x(\boldsymbol{\xi}, t)) = E(x^{2}(\boldsymbol{\xi}, t)) - (E(x(\boldsymbol{\xi}, t)))^{2}$$
$$= \sum_{k=0}^{P} \hat{x}_{k}^{2}(t) E(\psi_{k}^{2}(\boldsymbol{\xi})) - (E(x(\boldsymbol{\xi}, t)))^{2}$$
(38)

The Hermite-chaos expansion was proven to be effective in solving stochastic differential equations with Gaussian uncertainties. In order to handle other types of typical uncertainty distributions such as uniform and Poisson, a generalization of the original Wiener-chaos expansion has been proposed based on polynomials in the Askey scheme as explained in Xiu and Karniadakis (2002) and summarized in Table 3.2.2.3.1.

Random variable $(\xi)$	Wiener-Askey chaos $\{\psi(\xi)\}$	Support
Gaussian	Hermite-chaos	(−∞,∞)
Gamma	Laguerre-chaos	[0,∞)
Beta	Jacobi-chaos	[ <i>a</i> , <i>b</i> ]
Uniform	Legendre-chaos	[ <i>a</i> , <i>b</i> ]
Poisson	Charlier-chaos	{0, 1, 2, }
Binomial	Krawtchouk-chaos	$\{0, 1,, N\}$

 Table 3.2.2.3.1: Wiener-Askey polynomial chaos

# 3.2.2.3.2. Implementation details

For the application of the polynomial chaos expansion theory, the PCET toolbox developed at the Technische Universität Chemnitz and reported in Streif et al. (2014) and Petzke et al. (2015) was employed. The procedure is represented graphically in Figure 3.2.2.3.2.



Figure 3.2.2.3.2: Procedure for nonlinear case with polynomial nonlinearities (PCET Toolbox)

The toolbox allows defining models in terms of their basic components: states, inputs, parameters and outputs, based on an organizational structure. Each component requires a certain name, which will be used in turn to write the differential and output equations of the system. Currently, there is only support for polynomial nonlinearities in the differential equations and for normal and uniform distributions to describe uncertainties in the initial states and parameters. The inputs are considered deterministic and can be defined as time sequences in piecewise fashion. Once the data structures are completed and the order of the polynomials is specified, the expanded equations (36) are composed and saved in files. It must be considered that the initial setup of the system takes a considerable amount of time in general.

The toolbox provides an update function to efficiently modify the description of the uncertainties in the initial states and parameters after the model has been defined. For the case of normal distributions, the mean and standard deviation can be adjusted, whereas for uniform distributions, the lower and upper bounds. This functionality is quite useful for setting up the integrations for different scenarios. The Galerkin projection method is provided as in (36), by which the composed system is then integrated to obtain the coefficients of the expansion for a given input sequence. Finally, the central moments are computed such as with (37) and (38) from the coefficients obtained in the previous step.

### 3.2.2.4. Nonlinear case with complex distributions

Markov chain models are proposed to represent complex uncertainties which cannot be correctly described by typical distributions. A short review on Markov chain fundamentals is therefore presented in the theoretical background section as in Brandimarte (2014). The application based on the Markov chain Monte Carlo methodology is presented in the following subsections.

#### 3.2.2.4.1. Theoretical background

A discrete-time Markov chain is a random process with a state variable  $X_k$ , k = 0, 1, 2, 3, ..., taking values on a discrete set. Typically, this type of process is represented with a directed graph, in which nodes correspond to states and directed arcs to the transition probabilities between them.

In order to determine the future evolution of the process, only the current state and the transition probabilities are required, which is an essential property of a Markov chain. For stationary processes, in which the transition probabilities do not depend on time, the chain can be described by a transition probability matrix, such that:

$$P_{ij}^{mc} = P\{X_{k+1} = j | X_k = i\}$$
(39)

Given the initial state distribution  $\pi_0 \in \Re^N$ ,  $\pi_{i,0} = P\{X_0 = i\}$ , the probability that the system will be in the next step in state  $X_1 = j$  is given by the chain rule:

$$P\{X_1 = j\} = \sum_{i=1}^{N} P\{X_1 = j | X_0 = i\}. P\{X_0 = i\}$$
(40)

or in matrix form:

$$\boldsymbol{\pi}_{1}^{T} = \boldsymbol{\pi}_{0}^{T} P^{mc} \tag{41}$$

The rule for distribution evolution can be generalized as follows:

$$\pi_{k+1}^{T} = \pi_{k}^{T} P^{mc} = \pi_{0}^{T} (P^{mc})^{k+1}$$
(42)

#### 3.2.2.4.2. Main contribution

Markov chains are envisioned to represent complex uncertainties such as hardware malfunctions. Such description may prove especially useful for cases regarding slow processes with rather long prediction horizons, in which a correct analysis of the uncertainty evolution should be taken into consideration to perform realistic predictions for performance monitoring purposes. The states of the Markov model  $X_{k'-k}$  are proposed to be the indices r(j) of the uncertainty realizations in the multi-stage NMPC scheme. A graphical representation for a simple example is shown in Figure 3.2.2.4.2. The state space is the set {L, N, H} and corresponds to the typical case of one uncertainty with three uncertainty values in the robust formulation.



Figure 3.2.2.4.2: Markov chain model example

Therefore, for a given scenario, the effective uncertainty realization in the evaluator will be given by:

$$\widehat{\boldsymbol{d}}_{\boldsymbol{k}\boldsymbol{\prime}|\boldsymbol{k}}^{*} = \widehat{\boldsymbol{d}}_{\boldsymbol{k}\boldsymbol{\prime}|\boldsymbol{k}}^{X_{\boldsymbol{k}^{\prime}-\boldsymbol{k}}},\tag{43}$$

where  $X_0 = r(j), \forall (j, 1) \in I$ . The initial state distribution is thus assigned by the following rule:

$$\pi_{i,0} = \begin{cases} 1, & if \ i = r(j) \\ 0 & otherwise \end{cases}$$
(44)

and its evolution is determined by (42). For the example, the transition probability matrix is given by:

$$P^{mc} = \begin{bmatrix} P_{LL}^{mc} & P_{LN}^{mc} & P_{LH}^{mc} \\ P_{NL}^{mc} & P_{NN}^{mc} & P_{NH}^{mc} \\ P_{HL}^{mc} & P_{HN}^{mc} & P_{HH}^{mc} \end{bmatrix},$$
(45)

which may be estimated based on maintenance history and process knowledge.

As a generalization, each state in the Markov model may be related to a standard statistical distribution, such as normal or uniform, representing the variability in the possible faults realization.

## 3.2.2.4.3. Implementation details

The procedure for Markov chain Monte Carlo simulation is represented graphically in Figure 3.2.2.4.3.



Figure 3.2.2.4.3: Markov chain Monte Carlo simulation flow chart

The procedure is initialized by reading from the database the required information. The initial state distribution of the Markov chain is assigned

according to (44). At each time period, a sample from the Markov chain model is taken in order to determine the state  $X_{k'-k}$ , from which an uncertainty realization can be obtained for the integration process. Following, the state probabilities are updated according to (42). The simulation is performed for a whole prediction horizon,  $N_{mc}$  times for each scenario. For each time step, estimations for the expected value, lower and upper bounds for the stage cost distribution are computed and stored properly.

## 3.3. Proposed applications

In this section some possible applications based on performance assessment are proposed in order to illustrate the potential of this methodology.

## 3.3.1. Controller selection

Performance assessment techniques may prove quite useful in the design stage of control systems. With such methodologies it would be possible to develop systematic design tools to automate the analysis and comparison of controllers, leading to the most suitable choice.

For controller selection based on performance assessment, a target design performance  $L_k^d$  must be specified in terms of an expected value and reasonable upper and lower bounds. Such desired performance may be related to the performance of an ideal controller or user defined. In this context, the following performance indices are proposed:

$$\alpha_k^d = \frac{E[L_k^d]}{L_k^a} \tag{46}$$

$$\beta_{k}^{d} = \begin{cases} \frac{q_{h}[L_{k}^{d}]}{L_{k}^{a}} & L_{k}^{a} > q_{h}[L_{k}^{d}] \\ 1 & q_{l}[L_{k}^{d}] \le L_{k}^{a} \le q_{h}[L_{k}^{d}] \\ \frac{q_{l}[L_{k}^{d}]}{L_{k}^{a}} & L_{k}^{a} < q_{l}[L_{k}^{d}] \end{cases}$$
(47)

34

Furthermore, in order to facilitate the comparison between different control structures, overall averaged measures of performance may be defined such as follows:

$$\Delta \alpha = \frac{\sum_{k=1}^{N_k} \alpha_k^d}{N_k},\tag{48}$$

$$\Delta\beta = \frac{\sum_{k=1}^{N_k} \beta_k^d}{N_k},\tag{49}$$

where  $N_k$  time stamps have been recorded from simulations. Once such indicators have been computed, a controller, from a list of possible candidates, is validated if  $\Delta \alpha$  and  $\Delta \beta$  satisfy performance specifications such as:

$$\Delta \alpha \ge \varepsilon_{\alpha} \tag{50}$$

$$\Delta\beta \ge \varepsilon_{\beta} \tag{51}$$

for given  $\varepsilon_{\alpha}$  and  $\varepsilon_{\beta}$ . With these considerations, a proposed procedure for controller selection is represented graphically in Figure 3.3.1.



Figure 3.3.1: Procedure for controller selection

#### 3.3.2. Performance monitoring and fault detection

The benchmark for performance monitoring has been defined in section 3.2.1.2, along with the corresponding performance indices. Additionally for this task, a benchmark based on historical data  $L_k^h$  may prove quite useful for practical online applications. This approach requires expert process knowledge to determine a time window in which the performance could be considered appropriate. Similarly to (46)-(47), the following indices are defined:

$$\alpha_k^h = \frac{E[L_k^h]}{L_k^a} \tag{52}$$

$$\beta_{k}^{h} = \begin{cases} \frac{q_{h}[L_{k}^{h}]}{L_{k}^{a}} & L_{k}^{a} > q_{h}[L_{k}^{h}] \\ 1 & q_{l}[L_{k}^{h}] \le L_{k}^{a} \le q_{h}[L_{k}^{h}] \\ \frac{q_{l}[L_{k}^{h}]}{L_{k}^{a}} & L_{k}^{a} < q_{l}[L_{k}^{h}] \end{cases}$$
(53)

Furthermore, in order to facilitate the online monitoring task, the following indices are defined:

$$\alpha_k^b = \alpha_{j^*k}^b, \quad j^* = argmin_j (|\alpha_{jk}^b - 1|), \tag{54}$$

$$\beta_{k}^{b} = \beta_{j^{*}k}^{b}, \quad j^{*} = argmin_{j}(|\beta_{jk}^{b} - 1|), \tag{55}$$

which represent the best indices found for all scenarios.

A thorough analysis on these indices may shed some light not only on the detection of performance degradation but also on possible root causes as discussed in Schäfer and Cinar (2002). Certainly, several root causes will have a significant impact on the history-based indices, while this is not necessarily true for those indices based on a process model. For instance, factors such as a significant plant-model mismatch, presence of disturbances out of the range

of uncertainty or actuator faults will not influence the model-based benchmark significantly, leading to relatively low values of  $\alpha_k^b$  and  $\beta_k^b$ . On the other hand, factors such as a poor tuning of the controller and input saturation will surely affect the model-based benchmark, leading to relatively small changes in the respective indices.

With these considerations, a proposed procedure for performance monitoring and fault detection is represented graphically in Figure 3.3.2.



Figure 3.3.2: Procedure for performance monitoring and fault detection

Based on historical data, a reference of acceptable performance  $L_k^h$  is selected. After the analysis of a time window and computation of indices, a flag of normal operation is raised if  $\alpha_k^h$  and  $\beta_k^h$  are in acceptable levels. Otherwise, the values of  $\alpha_k^b$  and  $\beta_k^b$  are taken into consideration to indicate the status of the system after the fault was detected.

#### 3.3.3. Re-tuning of the controller

In most cases the weights  $\omega_i$  of the scenarios are considered invariant with time and equal to each other so that  $\omega_i = \frac{1}{N}$ , indicating that all scenarios are equally important for robust reasons.

However, for some applications it may be desirable to prioritize performance over robustness. For such cases, an update formula, based on performance assessment methodologies, can be proposed to adjust the weights of the scenarios dynamically considering the model-based performance indices as follows:

$$\omega_{jk} = \frac{\frac{c_{\alpha}}{\left|\alpha_{jk}^{b}-1\right|+\varepsilon} + \frac{c_{\beta}}{\left|\beta_{jk}^{b}+1\right|+\varepsilon}}{\sum_{j} \left(\frac{c_{\alpha}}{\left|\alpha_{jk}^{b}-1\right|+\varepsilon} + \frac{c_{\beta}}{\left|\beta_{jk}^{b}+1\right|+\varepsilon}\right)},$$
(56)

where  $c_{\alpha}$  and  $c_{\beta}$  are weighting coefficients and  $\varepsilon$  is a small number. Therefore, more weight is systematically given to scenarios with indices close to 1 in order to improve the performance of the system. In general for this operation, the indices obtained from multi-step predictions  $\alpha_{jk}^{bK}$  and  $\beta_{jk}^{bK}$  should be employed, as they are directly related to the scenarios. Only for cases in which the robust horizon equals 1, the indices obtained from one-step predictions  $\alpha_{jk}^{b1}$  and  $\beta_{jk}^{b1}$  may also be used.

## 3.3.4. Training of operators

In recent times, the use of dynamic simulation and operator training systems has become a reality in the process industry. Such systems allow training operators, instructors and plant management in how to best operate their facilities. Given the current trend of advanced control implementations, such tools will prove mandatory to test and validate control strategies and logic before start-up of main facilities and to investigate future engineering solutions. This is especially true for MPC applications, whose counterintuitive behavior may be hard to analyze even for expert plant personnel. Provided performance

indicators in the training environment, such analysis would be considerably facilitated, allowing operators not only to evaluate the effects of controller parameters tuning, but also to recognize normal from abnormal situations with easiness. Troubleshooting procedures and settings for emergency control strategies without risking plant operation may therefore be put in practice beforehand increasing the confidence in operators.

Furthermore, for a known given process scenario set up by an experienced operator, the visualization and study of the controller predictions would allow a better understanding on the decision criterion employed by the optimizer. Such simulation should be performed under realistic considerations and typical process events, with which the operators are already familiarized. With the support of the presented performance indicators, training tasks such as pretuning of new control loops and evaluation of design changes before implementation may be safely performed.

# 4. Case studies

In this chapter the case studies used in this project will be presented. The description includes differential equations and process parameters, the setup of the estimators and the optimization problems. All tables with configuration parameters will be included in the appendix section.

## 4.1. Controller selection for a 4-Tank system

## 4.1.1. Process module

A quadruple-tank process as presented in Botelho et al. (2016) was used as case study for controller selection. The system consists of four interconnected tanks as shown in Figure 4.1.1.



Figure 4.1.1: Schematic representation of 4-Tank system (Botelho et al. 2016)

From a reservoir, water is pumped to the tanks by means of voltages  $v_{p1}$  and  $v_{p2}$ . Additionally, the flux to the tanks can be regulated by means of two valves with openings  $x_{v1}$  and  $x_{v2}$ . The state vector is defined as  $\mathbf{x} = [h_1 \ h_2 \ h_3 \ h_4]^T$  and the control input as  $\mathbf{u} = [v_{p1} \ v_{p2} \ x_{v1} \ x_{v2}]^T$ .

From mass balances involving tank levels, the nonlinear differential equations that describe the system can be obtained:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \tag{57a}$$

$$f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} -\frac{c_{d1}}{A_{s1}} (x_1)^{e_{p1}} + \frac{c_{d3}}{A_{s1}} (x_3)^{e_{p3}} + \frac{u_3k_1}{A_{s1}} u_1 \\ -\frac{c_{d2}}{A_{s2}} (x_2)^{e_{p2}} + \frac{c_{d4}}{A_{s2}} (x_4)^{e_{p4}} + \frac{u_4k_2}{A_{s2}} u_2 \\ -\frac{c_{d3}}{A_{s3}} (x_3)^{e_{p3}} + \frac{(1-u_4)k_2}{A_{s3}} u_2 \\ -\frac{c_{d4}}{A_{s4}} (x_4)^{e_{p4}} + \frac{(1-u_3)k_1}{A_{s4}} u_1 \end{bmatrix}$$
(57b)

Nominal values for the parameters of the system are shown in Table A.1.1.1.

From (57), the linearization of the system around the equilibrium point  $(x_s, u_s)$  is given by:

$$\Delta \dot{x} = A(x_s, u_s) \Delta x + B(x_s, u_s) \Delta u, \ \Delta x = x - x_s, \ \Delta u = u - u_s$$
(58a)

 $A(\boldsymbol{x}_s, \boldsymbol{u}_s)$ 

$$= \begin{bmatrix} -\frac{c_{d1}e_{p1}}{A_{s1}}(x_{s1})^{e_{p1}-1} & 0 & \frac{c_{d3}e_{p3}}{A_{s1}}(x_{s3})^{e_{p3}-1} & 0 \\ 0 & -\frac{c_{d2}e_{p2}}{A_{s2}}(x_{s2})^{e_{p2}-1} & 0 & \frac{c_{d4}e_{p4}}{A_{s2}}(x_{s4})^{e_{p4}-1} \\ 0 & 0 & -\frac{c_{d3}e_{p3}}{A_{s3}}(x_{s3})^{e_{p3}-1} & 0 \\ 0 & 0 & 0 & -\frac{c_{d4}e_{p4}}{A_{s4}}(x_{s4})^{e_{p4}-1} \end{bmatrix}$$

$$B(\boldsymbol{x}_{s}, \boldsymbol{u}_{s}) = \begin{bmatrix} \frac{k_{1}u_{s3}}{A_{s1}} & 0 & \frac{k_{1}u_{s1}}{A_{s1}} & 0\\ 0 & A_{s2} & 0 & A_{s2}\\ 0 & \frac{k_{2}(1-u_{s4})}{A_{s3}} & 0 & -\frac{k_{2}u_{s2}}{A_{s3}}\\ \frac{k_{1}(1-u_{s3})}{A_{s4}} & 0 & -\frac{k_{1}u_{s1}}{A_{s4}} & 0 \end{bmatrix}$$
(58c)

The deviations  $\Delta x$  are available as output measurements. The initial value of  $\Delta x$  and  $\Delta u$ , as well as the equilibrium point selected for the study are shown in Tables A.1.1.2 and A.1.1.3, respectively.

(58b)

Uncertainties in the form of normally-distributed noise and deterministic disturbances will affect the process. Parametric disturbances  $\delta c_{d3}$  and  $\delta c_{d4}$  as deviations from the nominal values in Table A.1.1.1 and input disturbances  $\delta u_1$  and  $\delta u_2$  as deviations from the inputs sent from the controller will be considered. The vector of disturbances  $d = [\delta c_{d3} \ \delta c_{d4} \ \delta u_1 \ \delta u_2]^T$  is added to the model, along with the noise terms:

$$\Delta \dot{\boldsymbol{x}} = A(\boldsymbol{x}_s, \boldsymbol{u}_s) \Delta \boldsymbol{x} + B(\boldsymbol{x}_s, \boldsymbol{u}_s) \Delta \boldsymbol{u} + D(\boldsymbol{x}_s, \boldsymbol{u}_s) \boldsymbol{d} + \boldsymbol{w},$$
(59a)

$$y = \Delta x + v, \tag{59b}$$

where all matrices are computed considering nominal parameter values and *D* is given by:

$$D(\mathbf{x}_{s}, \mathbf{u}_{s}) = \begin{bmatrix} \frac{(x_{s3})^{e_{p3}}}{A_{s1}} & \frac{0}{(x_{s4})^{e_{p4}}} & \frac{k_{1}u_{s3}}{A_{s1}} & \frac{0}{A_{s2}} \\ 0 & A_{s2} & 0 & A_{s2} \\ -\frac{(x_{s3})^{e_{p3}}}{A_{s3}} & 0 & 0 & \frac{k_{2}(1-u_{s4})}{A_{s3}} \\ 0 & -\frac{(x_{s4})^{e_{p4}}}{A_{s4}} & \frac{k_{1}(1-u_{s3})}{A_{s4}} & \frac{A_{s3}}{A_{s3}} \end{bmatrix}$$
(59c)

The noise distribution and the description of the deterministic disturbances as process events are shown in Tables A.1.1.4 and 4.1.1, respectively. The system (59) must be discretized in time considering the period of the optimizer in order to apply the results obtained in section 3.2.2.2.

Variable	Start time (min)	End time (min)	Nominal value	Test Value
$\delta c_{d3}$	10.0	15.0	0.0	-0.4
$\delta c_{d4}$	10.0	15.0	0.0	-0.3
$\delta u_1$	10.0	15.0	0.0	0.4

Table 4.1.1: Process events for 4-Tank system

#### 4.1.2. Estimator module

The estimator is a Kalman filter working with the nominal model (58). Table A.1.2.1 shows the initial conditions  $\Delta \hat{x}_{0|-1} = [\Delta \hat{x}_{01}, \Delta \hat{x}_{02}, \Delta \hat{x}_{03}, \Delta \hat{x}_{04}]^T$  and  $P_{x,0|-1} = diag(P_{x01}, P_{x02}, P_{x03}, P_{x04})$ . On the other hand, the noise covariance matrices, tuned from process information,  $Q_w = diag(Q_{w1}, Q_{w2}, Q_{w3}, Q_{w4})$  and  $R_v = diag(R_{v1}, R_{v2}, R_{v3}, R_{v4})$  are shown in Table A.1.2.2. The period for the estimator was chosen as  $T_e = 5.0$  s.

### 4.1.3. Optimizer module

For the benchmark, the performance of an unconstrained MPC controller was taken into consideration. For such controller, the optimization problem is formulated as follows:

$$(\Delta \boldsymbol{u}_{\boldsymbol{k}'} - \Delta \boldsymbol{u}_{\boldsymbol{k}'-1})^T \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} (\Delta \boldsymbol{u}_{\boldsymbol{k}'} - \Delta \boldsymbol{u}_{\boldsymbol{k}'-1})$$
(60a)

subject to:

$$\Delta x_{k'+1} = A_d(x_s, u_s) \Delta x_{k'} + B_d(x_s, u_s) \Delta u_{k'}, \Delta x_0 = \Delta \widehat{x}_{k+1|k-1}, \quad (60b)$$

where  $A_d(\mathbf{x}_s, \mathbf{u}_s)$  and  $B_d(\mathbf{x}_s, \mathbf{u}_s)$  are time discretizations of (58b) and (58c).

For the test cases based on multi-stage MPC, the optimization problems are formulated as:

subject to:

$$\Delta x_{k'+1}^{j} = A_{d}(x_{s}, u_{s}) \Delta x_{k'}^{p(j)} + B_{d}(x_{s}, u_{s}) \Delta u_{k'}^{j} + D_{d}(x_{s}, u_{s}) d_{k'}^{r(j)},$$
  

$$\Delta x_{0} = \Delta \hat{x}_{k+1|k-1}, \forall (j, k'+1) \in I,$$
(61b)

$$-1 \le \Delta x_{i,k'}^{j} \le 1, \forall i = 1, ..., 4, \forall (j,k') \in I,$$
(61c)

$$-2.5 \le \Delta u_{i,k'}^{j} \le 2.5, \forall i = 1, 2, \forall (j,k') \in I,$$
(61d)

$$-0.25 \le \Delta u_{i,k'}^{j} \le 0.25, \forall i = 3, 4, \forall (j, k') \in I,$$
(61e)

$$\Delta u_{k'}^{j} = \Delta u_{k'}^{l} \ if \ \Delta x_{k'}^{p(j)} = \Delta x_{k'}^{p(l)}, \forall (j,k'), (l,k') \in I,$$
(61f)

where  $d_{k'}^{r(j)}$  takes all combinations of 3 elements from vector *d*, defined in section 4.1.1. The uncertainty values considered are [-1 0 1] for all cases.  $D_d$  is the respective time discretization of (59c) for each combination. Radau collocation points have been used as discretization method with second degree polynomials and two finite elements per control interval. The robust horizon is equal to 1 and the period of the optimizer was chosen as T<sub>o</sub> = 5.0 s.

## 4.2. Performance monitoring on a CSTR

#### 4.2.1. Process module

A continuously stirred tank reactor (CSTR) as presented in Kvasnica et al. (2010) was used as a case study for performance monitoring. In the CSTR three reactions take place to convert the input feed component A in the desired component B including two side products C and D as shown in Figure 4.2.1.



Figure 4.2.1: Schematic representation of CSTR (Kvasnica et al. 2010)

The CSTR is equipped with a jacket to cool down the reactor. The input feed flow *F* and the cooling power  $\dot{Q}$  define the control input vector  $\boldsymbol{u} = [F \dot{Q}]^T$ . Furthermore, a state vector composed of concentrations and temperatures is defined as  $\boldsymbol{x} = [C_a C_b T_R T_K]^T$ , where  $C_a$  and  $C_b$  represent the concentration of components A and B, respectively. On the other hand, the temperatures  $T_R$  and  $T_K$  represent the reactor and the cooling jacket temperature, respectively.

From mass balances involving components concentration and energy balances regarding temperatures, the nonlinear differential equations that describe the system can be obtained:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}), \tag{62a}$$

$$f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} u_1(C_{a,in} - x_1) - K_1 x_1 - K_3 x_1^2 \\ -u_1 x_2 + K_1 x_1 - K_2 x_2 \\ \frac{K_1 x_1 H_R^{ab} + K_2 x_1 H_R^{bc} + K_3 x_1^{2} H_R^{ad}}{-\rho C_p} + u_1 (T_{in} - x_3) + \frac{K_w A_R}{\rho C_p V_R} (x_4 - x_3) \\ \frac{u_2 + K_w A_R (x_3 - x_4)}{m_K C_{p,K}} \end{bmatrix}, \quad (62b)$$

where the reaction rates are driven by the Arrhenius law:

$$K_1 = \beta^{ab} K_0^{ab} e^{\left(-\frac{E_A^{ab}}{x_3 + 273.15}\right)}$$
(62c)

$$K_2 = K_0^{bc} e^{\left(-\frac{E_A^{bc}}{x_3 + 2^{73.15}}\right)}$$
(62d)

$$K_3 = K_0^{ad} e^{\left(-\alpha^{ad} \frac{E_A^{ad}}{x_3 + 273.15}\right)}$$
(62e)

Nominal values of the individual parameters of the system are shown in Table A.2.1.1. On the other hand, the initial value of x and u is shown in Table A.2.1.2.

Uncertainties in the form of normally-distributed noise and multiplicative deterministic disturbances will affect the process. Input disturbances  $\delta u_1$  and  $\delta u_2$  as deviations from the inputs sent from the controller and a parametric disturbance  $\delta C_{a,in}$  as deviation from the nominal value in Table A.2.1.1 will be considered. The vector of disturbances  $d = [\delta u_1 \, \delta u_2 \, \delta C_{a,in}]^T$  is included in the model, along with the noise terms:

$$\dot{\boldsymbol{x}} = f_d(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}) + \boldsymbol{w} \tag{63a}$$

$$y = x + v \tag{63b}$$

$$f_{d}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}) = \begin{bmatrix} u_{1}d_{1}(C_{a,in}d_{3} - x_{1}) - K_{1}x_{1} - K_{3}x_{1}^{2} \\ -u_{1}d_{1}x_{2} + K_{1}x_{1} - K_{2}x_{2} \\ \frac{K_{1}x_{1}H_{R}^{ab} + K_{2}x_{1}H_{R}^{bc} + K_{3}x_{1}^{2}H_{R}^{ad}}{-\rho C_{p}} + u_{1}d_{1}(T_{in} - x_{3}) + \frac{K_{w}A_{R}}{\rho C_{p}V_{R}}(x_{4} - x_{3}) \\ \frac{u_{2}d_{2} + K_{w}A_{R}(x_{3} - x_{4})}{m_{K}C_{p,K}} \end{bmatrix}$$
(63c)

The noise distribution and the description of the deterministic disturbances as process events are shown in Tables A.2.1.3 and 4.2.1, respectively.

Variable	Start time (min)	End time (min)	Nominal Value	Test Value
$\delta u_1$	20.0	25.0	1.0	1.25
$\delta u_2$	10.0	15.0	1.0	0.8
$\delta C_{a,in}$	30.0	35.0	1.0	0.8

Table 4.2.1: Process events for CSTR

#### 4.2.2. Estimator module

The Extended Kalman filter uses the nominal model (62) extended for input disturbance estimation assuming a zero order hold behavior for  $\delta u_1$  and  $\delta u_2$ . Considering that  $\hat{x} = [\hat{C}_a \hat{C}_b \hat{T}_R \hat{T}_K \delta \hat{u}_1 \delta \hat{u}_2]^T$ , the extended model used by the estimator is given by:

$$\hat{\boldsymbol{x}} = f_e(\hat{\boldsymbol{x}}, \boldsymbol{u}), \tag{64a}$$

$$\widehat{\boldsymbol{y}} = [\boldsymbol{I}_4 \quad \boldsymbol{0}_{4x2}]\widehat{\boldsymbol{x}},\tag{64b}$$

$$f_e(\hat{\mathbf{x}}, \mathbf{u}) = \begin{bmatrix} f_d([\hat{x}_1 \ \hat{x}_2 \ \hat{x}_3 \ \hat{x}_4]^T, \mathbf{u}, [\hat{x}_5 \ \hat{x}_6 \ 1]^T) \\ \mathbf{0}_{2x1} \end{bmatrix},$$
(64c)

where *I* is the identity matrix and **0** is the zero matrix. Table A.2.2.1 shows the initial conditions  $\hat{x}_{0|-1} = [\hat{x}_{01}, \hat{x}_{02}, ..., \hat{x}_{06}]$  and  $P_{x,0|-1} = diag(P_{x01}, P_{x02}, ..., P_{x06})$ . On the other hand, the noise covariance matrices, tuned from process information,  $Q_w = diag(Q_{w1}, Q_{w2}, ..., Q_{w6})$  and  $R_v = diag(R_{v1}, R_{v2}, R_{v3}, R_{v4})$  are shown in Table A.2.2.2. The period for the estimator was chosen as  $T_e = 5.0$  s.

# 4.2.3. Optimizer module

The optimization problem for multi-stage NMPC is formulated as follows:

subject to:

$$\boldsymbol{x}_{k'+1}^{j} = f_d\left(\boldsymbol{x}_{k'}^{p(j)}, \boldsymbol{u}_{k'}^{j} \begin{bmatrix} \boldsymbol{d}_{k'}^{r(j)} \\ 1 \end{bmatrix}\right), \boldsymbol{x}_0 = \widehat{\boldsymbol{x}}_{k+1|k-1}, \forall (j,k'+1) \in I,$$
(65b)

$$1.5 \le x_{1,k'}^j \le 2.3, \ \forall (j,k') \in I,$$
 (65c)

$$0.6 \le x_{2,k'}^j \le 0.8, \ \forall (j,k') \in I,$$
(65d)

$$120 \le x_{3,k'}^j \le 160, \ \forall (j,k') \in I,$$
(65e)

$$40 \le x_{4,k'}^j \le 180, \ \forall (j,k') \in I,$$
(65f)

$$5 \le u_{1,k'}^j \le 250, \forall (j,k') \in I,$$
 (65g)

$$-85000 \le u_{2,k'}^{j} \le 0, \forall (j,k') \in I,$$
(65h)

$$\boldsymbol{u}_{k'}^{j} = \boldsymbol{u}_{k'}^{l} \text{ if } \boldsymbol{x}_{k'}^{p(j)} = \boldsymbol{x}_{k'}^{p(l)}, \forall (j,k'), (l,k') \in I,$$
(65i)

The uncertainties considered in the optimization problem are  $\delta u_1$  and  $\delta u_2$  with uncertainty values [0.7 1.0 1.3]. Radau collocation points have been used as discretization method with second degree polynomials and two finite elements per control interval. The robust horizon is equal to 1 and the period of the optimizer was chosen as  $T_0 = 5.0$  s.

## 4.2.4. Performance assessment module

The benchmark for online performance assessment for this case study was computed by means of the PCET toolbox with the Galerkin projection method. The configuration included 8 uncertain parameters:  $[w_1 w_2 w_3 w_4 v_1 v_2 \delta u_1 \delta u_2]$  with order 1 for the polynomials. The statistical description of  $\delta u_1$  and  $\delta u_2$  for each scenario is given in Table 4.2.4. The reaction rates were approximated by a 4th-degree Taylor expansion in order to make the equations compatible with the toolbox.

Scenario	<b>Distribution for</b> $\delta u_1$	Distribution for $\delta u_2$
1	N(0.7750, 0.0375 <sup>2</sup> )	$N(0.7750, 0.0375^2)$
2	N(0.7750, 0.0375 <sup>2</sup> )	N(1,0.0150 <sup>2</sup> )
3	N(0.7750, 0.0375 <sup>2</sup> )	N(1.2250, 0.0375 <sup>2</sup> )
4	N(1,0.0150 <sup>2</sup> )	$N(0.7750, 0.0375^2)$
5	N(1,0.0150 <sup>2</sup> )	$N(1, 0.0150^2)$
6	N(1,0.0150 <sup>2</sup> )	$N(1.2250, 0.0375^2)$
7	N(1.2250, 0.0375 <sup>2</sup> )	$N(0.7750, 0.0375^2)$
8	N(1.2250, 0.0375 <sup>2</sup> )	$N(1, 0.0150^2)$
9	N(1.2250, 0.0375 <sup>2</sup> )	N(1.2250, 0.0375 <sup>2</sup> )

**Table 4.2.4**: Statistical description of  $\delta u_1$  and  $\delta u_2$  for performance monitoring on CSTR

### 4.3. Performance monitoring on a polymerization reactor

#### 4.3.1. Process module

An industrial batch polymerization reactor as presented in Lucía et al. (2014) was used as a case study for performance monitoring. In the reactor a very exothermic reaction takes place in order to produce polymer B from monomer A as shown in Figure 4.3.1. The reactor is equipped with a jacket and an external heat exchanger to allow regulation of internal temperature. The input feed flow of A  $\dot{m}_F$  and the coolant temperatures at the inlet of the jacket  $T_M^{IN}$  and at the inlet of the heat exchanger  $T_{AWT}^{IN}$  define the input vector  $\boldsymbol{u} = [\dot{m}_F T_M^{IN} T_{AWT}^{IN}]^T$ . Furthermore, a state vector composed of masses and temperatures in the system is defined as  $\boldsymbol{x} = [m_W m_A m_P T_R T_S T_M T_{EK} T_{AWT} T_{adiab} m_F^{acc}]^T$ , where  $m_W$  is the mass of water in the reactor,  $m_A$  the mass of monomer,  $m_P$  the mass of product,  $T_R$  the temperature of the mixture in the heat exchanger,  $T_{AWT}$  the temperature of the coolant leaving the heat exchanger,  $T_{adiab}$  the maximum temperature in the reactor under cooling system failure and  $m_F^{acc}$ , the total amount of material that has been fed in the reactor.



Figure 4.3.1: Schematic representation of batch polymerization system (Lucía et al. 2014)

From mass balances and energy balances regarding temperatures in the system, the nonlinear differential equations that describe the system can be obtained:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \tag{66a}$$

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

$$\begin{bmatrix} u_{1}\omega_{W,F} \\ u_{1}\omega_{A,F} - k_{R1}m_{A,R} - \frac{k_{R2}m_{AWT}x_{2}}{m_{ges}} \\ k_{R1}m_{A,R} + \frac{k_{R2}m_{AWT}x_{2}}{m_{ges}} \\ \frac{1}{c_{p,R}m_{ges}} \Big( u_{1}c_{p,F}(T_{F} - x_{4}) + \Delta H_{R}k_{R1}m_{A,R} - k_{K}A(x_{4} - x_{5}) - \dot{m}_{AWT}c_{p,R}(x_{4} - x_{7}) \Big) \\ \frac{1}{c_{p,S}m_{S}} \Big( k_{K}A(x_{4} - x_{5}) + k_{K}A(x_{6} - x_{5}) \Big) \\ \frac{1}{c_{p,W}m_{M,KW}} \Big( \dot{m}_{M,KW}c_{p,W}(u_{2} - x_{6}) + k_{K}A(x_{5} - x_{6}) \Big) \\ \frac{1}{c_{p,R}m_{AWT}} \Big( \dot{m}_{AWT}c_{p,R}(x_{4} - x_{7}) + \alpha^{exp}(x_{8} - x_{7}) + \frac{k_{R2}m_{AWT}\Delta H_{R}x_{2}}{m_{ges}} \Big) \\ \frac{1}{c_{p,W}m_{AWT,KW}} \Big( \dot{m}_{AWT,KW}c_{p,W}(u_{3} - x_{8}) + \alpha^{exp}(x_{7} - x_{8}) \Big) \\ \frac{\Delta H_{R}}{c_{p,R}m_{ges}} \dot{x}_{2} - (\dot{x}_{1} + \dot{x}_{2} + \dot{x}_{3}) \Big( \frac{x_{2}\Delta H_{R}}{c_{p,R}m_{ges}^{2}} \Big) + \dot{x}_{4} \\ u_{1} \end{bmatrix}$$
(66b)

where

$$U = \frac{x_3}{x_2 + x_3}$$
(66c)

$$m_{ges} = x_1 + x_2 + x_3 \tag{66d}$$

$$k_{R1} = k_0 e^{\frac{-E_a}{R(x_4 + 273.15)}} (k_{U1}(1 - U) + k_{U2}U)$$
(66e)

$$k_{R2} = k_0 e^{\frac{-E_a}{R(x_7 + 273.15)}} (k_{U1}(1 - U) + k_{U2}U)$$
(66f)

$$k_{K} = \frac{k_{WS}x_1 + k_{AS}x_2 + k_{PS}x_3}{m_{ges}}$$
(66g)

$$m_{A,R} = x_2 - \frac{m_{AWT} x_2}{m_{ges}}$$
 (66h)

51

Nominal values of the individual parameters are shown in Table A.3.1.1. On the other hand, the initial value of x and u is shown in Table A.3.1.2.

Uncertainties in the form of normally-distributed noise and multiplicative deterministic disturbances will affect the process. Input disturbances  $\delta u_1$ ,  $\delta u_2$  and  $\delta u_3$  as deviations from the inputs sent from the controller will be considered. The vector of disturbances  $d = [\delta u_1 \ \delta u_2 \ \delta u_3]$  is included in the model, along with the noise terms:

$$\dot{\boldsymbol{x}} = f_d(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}) + \boldsymbol{w} \tag{67a}$$

$$y = x + v \tag{67b}$$

$$f_d(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}) = f(\boldsymbol{x}, \boldsymbol{u} \circ \boldsymbol{d}) \tag{67c}$$

The operator • denote the component-wise Hadamard product. The noise distribution and the description of the deterministic disturbances as process events are shown in Tables A.3.1.3 and 4.3.1, respectively.

Variable	Start time (min)	End time (min)	Nominal Value	Test Value
$\delta u_1$	30.0	40.0	1.0	1.0005
$\delta u_2$	55.0	65.0	1.0	0.995
$\delta u_3$	80.0	90.0	1.0	0.995

 Table 4.3.1: Process events for batch polymerization reactor

### 4.3.2. Estimator module

The Extended Kalman filter uses the nominal model (66) extended for input disturbance estimation assuming a zero order hold behavior for  $\delta u_1$  and  $\delta u_2$ . Considering that  $\hat{x} = [\hat{m}_W \, \hat{m}_A \, \hat{m}_P \, \hat{T}_R \, \hat{T}_S \, \hat{T}_M \, \hat{T}_{EK} \, \hat{T}_{AWT} \, \hat{T}_{adiab} \, \hat{m}_F^{acc} \, \delta \hat{u}_1 \, \delta \hat{u}_2]^T$ , the extended model used by the estimator is given by:

$$\dot{\hat{\boldsymbol{x}}} = f_e(\hat{\boldsymbol{x}}, \boldsymbol{u}) \tag{68a}$$

$$\widehat{\mathbf{y}} = \begin{bmatrix} \mathbf{I}_{10} & \mathbf{0}_{10x2} \end{bmatrix} \widehat{\mathbf{x}}$$
(68b)

$$f_e(\widehat{\boldsymbol{x}}, \boldsymbol{u}) = \begin{bmatrix} f_d([\widehat{x}_1 \ \widehat{x}_2 \ \dots \ \widehat{x}_{10}]^T, \boldsymbol{u}, [\widehat{x}_{11} \ \widehat{x}_{12} \ 1]^T) \\ \boldsymbol{0}_{2x1} \end{bmatrix}$$
(68c)

Table A.3.2.1 shows the initial conditions  $\hat{x}_{0|-1} = [\hat{x}_{01}, \hat{x}_{02}, ..., \hat{x}_{012}]$  and  $P_{x,0|-1} = diag(P_{x01}, P_{x02}, ..., P_{x012})$ . On the other hand, the noise covariance matrices, tuned from process information,  $Q_w = diag(Q_{w1}, Q_{w2}, ..., Q_{w12})$  and  $R_v = diag(R_{v1}, R_{v2}, ..., R_{v10})$  are shown in Table A.3.2.2. The period for the estimator was chosen as  $T_e = 5$  s.

#### 4.3.3. Optimizer module

The polymerization reactor is a batch process that will be operated optimally in terms of an economic cost function. For this case study the use of soft constraints will prove quite valuable to maintain the reactor temperature inside safety bounds.

The optimization problem for multi-stage NMPC is formulated as follows:

$$\begin{aligned} \mathbf{x}_{k'}^{j}, \mathbf{u}_{k'}^{j} \forall (j, k') \in I \sum_{j=1}^{9} \sum_{k'=1}^{29} -x_{3,k'}^{j} + \left(\mathbf{u}_{k'}^{j} - \mathbf{u}_{k'-1}^{p(j)}\right)^{T} \begin{bmatrix} 10^{-5} & 0 & 0 \\ 0 & 2x10^{-4} & 0 \\ 0 & 0 & 10^{-4} \end{bmatrix} \left(\mathbf{u}_{k'}^{j} - \mathbf{u}_{k'-1}^{p(j)}\right) + \\ & 10 \left(H(x_{3,k'}^{j} - 368.15)\right) \left(x_{3,k'}^{j} - 368.15\right)^{2} + \\ & 10 \left(1 - H(x_{3,k'}^{j} - 358.15)\right) \left(x_{3,k'}^{j} - 358.15\right)^{2} \end{aligned}$$
(69a)

subject to:

$$\mathbf{x}_{k'+1}^{j} = f_d\left(\mathbf{x}_{k'}^{p(j)}, \mathbf{u}_{k''}^{j} \begin{bmatrix} \mathbf{d}_{k'}^{r(j)} \\ 1 \end{bmatrix}\right), \mathbf{x}_0 = \hat{\mathbf{x}}_{k+1|k-1}, \forall (j, k'+1) \in I,$$
(69b)

$$x_{i,k'}^{j} \ge 0, \ \forall i = 1, 2, \forall (j,k') \in I,$$
(69c)

53

$$x_{3,k'}^j \ge 26, \ \forall (j,k') \in I,$$
 (69d)

$$353.15 \le x_{4,k'}^j \le 373.15, \ \forall (j,k') \in I,$$
(69e)

$$298 \le x_{i,k'}^j \le 400, \ \forall i = 5, 6, \forall (j,k') \in I,$$
(69f)

$$288 \le x_{i,k'}^j \le 400, \ \forall i = 7, 8, \forall (j,k') \in I,$$
(69g)

$$0 \le x_{9,k'}^{j} \le 30000, \ \forall (j,k') \in I,$$
(69h)

$$x_{10,k'}^j \le 392.15, \ \forall (j,k') \in I,$$
 (69i)

$$0 \le u_{1,k'}^{J} \le 30000, \forall (j,k') \in I,$$
(69j)

$$333.15 \le u_{i,k'}^{j} \le 373.15, \forall i = 2, 3, \forall (j,k') \in I,$$
(69k)

$$\boldsymbol{u}_{k'}^{j} = \boldsymbol{u}_{k'}^{l} \text{ if } \boldsymbol{x}_{k'}^{p(j)} = \boldsymbol{x}_{k'}^{p(l)}, \forall (j,k'), (l,k') \in I,$$
(691)

where H(.) denotes the Heaviside step function. This function is used to define conveniently soft constraints for the temperature of the reactor  $x_{4,k'}^{j}$  in (69a). With the soft constraints, some weight is added to the cost function if  $x_{4,k'}^{j}$  lies outside the range [358.15 368.15].

The uncertainties considered in the optimization problem are  $\delta u_1$  and  $\delta u_2$  with uncertainty values [0.999 1 1.001] and [0.99 1 1.01], respectively. Radau collocation points have been used as discretization method with second degree polynomials and one finite element per control interval. The robust horizon is equal to 1 and the period of the optimizer was chosen as  $T_0 = 5.0$  s.

## 4.3.4. Performance assessment module

The benchmark for performance assessment for this case study was computed considering a Markov chain model for input disturbances. Assuming that the Markov chain that describes one actuator fault can be correctly represented as shown in Figure 4.3.4, the entries of the transition probability matrix for the system of two independent actuators are given in Table 4.3.4.1. On the other hand, the statistical description of  $\delta u_1$  and  $\delta u_2$  for each scenario is given in Table 4.3.4.2.



Figure 4.3.4: Markov chain that describes one actuator fault in batch reactor

P <sup>mc</sup> <sub>ij</sub>	LL	LN	LH	NL	NN	NH	HL	HN	HH
LL	0.9976	0.0012	0	0.0012	0	0	0	0	0
LN	0.0003	0.9982	0.0003	0	0.0012	0	0	0	0
LH	0	0.0012	0.9976	0	0	0.0012	0	0	0
NL	0.0003	0	0	0.9982	0.0012	0	0.0003	0	0
NN	0	0.0003	0	0.0003	0.9988	0.0003	0	0.0003	0
NH	0	0	0.0003	0	0.0012	0.9982	0	0	0.0003
HL	0	0	0	0.0012	0	0	0.9976	0.0012	0
HN	0	0	0	0	0.0012	0	0.0003	0.9982	0.0003
HH	0	0	0	0	0	0.0012	0	0.0012	0.9976

Table 4.3.4.1: Transition probability matrix for benchmark computation

Scenario	<b>Distribution for</b> $\delta u_1$	<b>Distribution for</b> $\delta u_2$
1	N(0.9992, 0.0001 <sup>2</sup> )	N(0.9925, 0.0013 <sup>2</sup> )
2	N(0.9992, 0.0001 <sup>2</sup> )	N(1,0.0005 <sup>2</sup> )
3	N(0.9992, 0.0001 <sup>2</sup> )	N(1.0075, 0.0013 <sup>2</sup> )
4	N(1,0.0001 <sup>2</sup> )	N(0.9925, 0.0013 <sup>2</sup> )
5	N(1,0.0001 <sup>2</sup> )	N(1,0.0005 <sup>2</sup> )
6	N(1,0.0001 <sup>2</sup> )	N(1.0075, 0.0013 <sup>2</sup> )
7	N(1.0008, 0.0001 <sup>2</sup> )	N(0.9925, 0.0013 <sup>2</sup> )
8	N(1.0008, 0.0001 <sup>2</sup> )	N(1,0.0005 <sup>2</sup> )
9	N(1.0008, 0.0001 <sup>2</sup> )	N(1.0075, 0.0013 <sup>2</sup> )

**Table 4.3.4.2**: Statistical description of  $\delta u_1$  and  $\delta u_2$  for performancemonitoring on the polymerization reactor

# 5. Results and discussion

In this chapter results obtained with case studies on controller selection and performance monitoring are presented and discussed. Results show how suitable performance indicators allow systematic comparison of controllers and analysis of control performance in normal and abnormal situations. All computational tasks were performed on a PC system with following specifications: Intel Core i7-6700 CPU @3.40 GHz, 4.00 GB RAM.

# 5.1. Controller selection for a 4-Tank system

## 5.1.1. Definition and computation of benchmark

As explained in section 3.3.1, the first step in the controller selection procedure is to define a desired target performance  $L_k^d$ , so that only test controllers that perform closely or better than desired can be taken into consideration. Such target performance has been established for this case study based on an ideal unconstrained MPC controller.

The time response of such a system under the test events shown in Table 4.1.1 is presented in Figure 5.1.1.1. The results obtained for the ideal case free of events are shown only for comparative purposes. The events time window is indicated with dashed lines.

At the beginning of the simulation, the tank levels  $h_1$  and  $h_2$  are above the desired equilibrium point, for which both pumps reduce the flow and the valves restrict the amount of water that enters the tanks until the equilibrium point is achieved. The input disturbance  $\delta u_1$  and the parametric disturbance  $\delta c_{d3}$  contribute both to a decrease in the level of tank 1, while the parametric disturbance  $\delta c_{d4}$  contribute to an increase in the level of tank 2. The response of the controller during the presence of disturbances is therefore to increase significantly the effective flow to tank 1 and to reduce the flow to tank 2. After the events time window, the system converges asymptotically to the equilibrium point.



Figure 5.1.1.1: Time response of the 4-Tank system with unconstrained MPC

The desired benchmark  $L_k^d$  may be conveniently defined based on (9) and efficiently computed as explained in section 3.2.2.2 for one scenario. Figure 5.1.1.2 shows the desired model-based benchmark along with the performance achieved by the unconstrained MPC controller  $L_k^a$ . A band of 2 standard deviations around the expected value is shown with blue dashed lines. A slight difference between  $L_k^a$  and  $L_k^d$  can be observed during the period with presence of disturbances mainly due to estimation errors in the Kalman filter.



Figure 5.1.1.2: Desired benchmark for controller selection

Computation results obtained with the theoretical method developed in section 3.2.2.2 and with the Monte Carlo method are shown in Table 5.1.1.1.

Variable $\theta = \{E[L_k^d], std[L_k^d]\}$	$E[L_k^d]$	$std[L_k^d]$
Theoretical method		
$\left(\frac{\sum_{k=1}^{N_k} \theta_{th}}{N_k}\right)$	1.2033	0.0990
Monte Carlo method		
$\left(\frac{\sum_{k=1}^{N_k} \theta_{mc}}{N_k}\right)$ , 100 iterations)	1.2043	0.0978
Mean percentage error		
$\sum_{k=1}^{N_k} \frac{\theta_{th} - \theta_{mc}}{\theta_{th} N_k} x100\%$	0.2	-0.8
Mean absolute percentage error		
$\sum_{k=1}^{N_k} \left  \frac{\theta_{th} - \theta_{mc}}{\theta_{th} N_k} \right  x 100\%$	1	4

 Table 5.1.1.1: Computation results for controller selection benchmark

Indicators such as the mean percentage error and mean absolute percentage error show an agreement between the numerical results obtained with the two methods despite the relatively low number of Monte Carlo iterations. Computation times are summarized in Table 5.1.1.2, showing the benefits of using the theoretical method over Monte Carlo simulations. For the case study, any number of iterations greater than 10 will provide statistically a speed-up in the computations.

Value
(0.5±0.2) ms
(0.10±0.05) ms
(0.2±0.1) x N <sub>mc</sub>

Table 5.1.1.2: Computation times for controller selection benchmark

# 5.1.2. Comparison of different structures

Once the benchmark is computed, different controller structures can be evaluated. For the present study, four structures with different uncertainty parameters will be analyzed as shown in Table 5.1.2.1.

Structure	Uncertainty d
S1	$[\delta c_{d3}  \delta c_{d4}  \delta u_1]$
S2	$[\delta c_{d3}  \delta c_{d4}  \delta u_2]$
S3	$[\delta c_{d3}  \delta u_1  \delta u_2]$
S4	$[\delta c_{d4}  \delta u_1  \delta u_2]$

Table 5.1.2.1: Controller structures for analysis

The controllers are set up as explained in section 4.1.3 with prediction horizons K = 60. Simulation results for each case are shown in Figure 5.1.2.1.



Figure 5.1.2.1: Time response of the 4-Tank system with different structures of multi-stage MPC

It can be observed that after the initial 10 minutes, in which the tank levels are stabilized at their equilibrium point, the performance is deteriorated due to disturbances. The control input bounds are active at times in order to satisfy the hard constraints for tank level  $h_4$  during the events time window. After inspecting the time responses, it may be hard to determine which controller performed better or according to requirements, for which the proper performance measurements should be computed. The achieved performance  $L_k^a$  of each structure is presented along with the benchmark in Figure 5.1.2.2.


Figure 5.1.2.2: Comparison of achieved performance

The abrupt changes in the control inputs lead to drastic peaks in the achieved performance. Once  $L_k^a$  is obtained, the performance indicators can be computed with (46) and (47). Results are shown in Figures 5.1.2.3 and 5.1.2.4 for the  $\alpha^d$  and  $\beta^d$  indices, respectively. For the computation of the indices an averaging window of 45 seconds was used.



**Figure 5.1.2.3**:  $\alpha_k^d$ - Performance index for different control structures



**Figure 5.1.2.4**:  $\beta_k^d$ - Performance index for different control structures

The figures show that during the period before the events time window, all structures perform most of the time better than the benchmark, including the first half of such window. After this time point and until the disturbances disappeared, the performance of the controllers is slightly worse than the benchmark, mainly due to the aggressive input changes. Finally, during the last minutes of simulation, the performance obtained is quite similar to the benchmark. As mentioned in section 3.3.1, indicators such as (48) and (49) can be computed in order to have an overall picture of the controller performance and to facilitate the selection process. Additionally, an indicator that penalizes only poorer performance than the benchmark may be also conceived to complement the comparative analysis. Table 5.1.2.2 shows such an indicator along with the performance specifications  $\varepsilon_{\alpha}$  and  $\varepsilon_{\beta}$  for each case.

Indicator $\theta = \{\alpha, \beta\}$	εα	εβ
$\Delta \theta = \frac{\sum_{k=1}^{N_k} \theta_k^d}{N_k}$	1.0	1.0
$\Delta \theta = 1 - \frac{\sum_{k=1}^{N_k} (1 - \theta_k^d) H (1 - \theta_k^d)}{N_k}$	0.8	0.9

Table 5.1.2.2: Performance specifications for controller selection

Table 5.1.2.3 shows the results of the computations for each structure. In bold it has been indicated the cases that comply with the specifications. According to the performance assessment procedure, structures S1 and S2 allow a better overall performance than the benchmark with less than 20% of performance reduction in average regarding the  $\alpha^d$ -index and less than 10% regarding the  $\beta^d$ -index. Additional considerations should be imposed in order to favor one structure over the other.

Indicator	S1	S2	S3	S4
$\Delta \alpha = \frac{\sum_{k=1}^{N_k} \alpha_k^d}{N_k}$	1.19	1.37	1.77	0.95
$\Delta \alpha = 1 - \frac{\sum_{k=1}^{N_k} (1 - \alpha_k^d) H (1 - \alpha_k^d)}{N_k}$	0.83	0.83	0.79	0.78
$\Delta\beta = \frac{\sum_{k=1}^{N_k} \beta_k^d}{N_k}$	1.12	1.18	1.42	0.96
$\Delta\beta = \overline{1 - \frac{\sum_{k=1}^{N_k} (1 - \beta_k^d) H (1 - \beta_k^d)}{N_k}}$	0.93	0.90	0.87	0.88

Table 5.1.2.3: Aggregate performance for controller structure selection

## 5.1.3. Analysis of dependence on the prediction horizon

The same procedure can be used to analyze the controller performance dependence on the prediction horizon. For this study the structure S1 has been selected and the unconstrained MPC benchmark will be used as before. Table 5.1.3.1 shows the test controllers and Figure 5.1.3.1, the time response for each case.

 Table 5.1.3.1: Controller prediction horizon for analysis

Controller	Prediction horizon K
C1	10
C2	20
C3	40
C4	80



Figure 5.1.3.1: Time response of the 4-Tank system with different prediction horizons

In this case it is rather clear that the performance of the controllers with the shortest prediction horizon is not as good as the other two. This fact can be more clearly seen with the proper performance assessment analysis. Figures 5.1.3.2 and 5.1.3.3 show the respective performance indices.



**Figure 5.1.3.2**:  $\alpha_k^d$ - Performance index for controllers with different prediction horizons



**Figure 5.1.3.3**:  $\beta_k^d$ - Performance index for controllers with different prediction horizons

Controllers C1 and C2 have a significantly poorer performance compared to the benchmark, even though the prediction horizons are quite similar. On the other hand, the performance of C3 is quite similar to the one analyzed in the previous

section with K = 60. Finally, the performance of C4 is much better than the benchmark for significant periods of time, only slightly deteriorated at the end of the events time window. These observations are coherent with the common notion that a longer prediction horizon would lead to a better performance as long as computation delays do not play a major role, which is the case for C4. The aggregate performance assessment is shown in Table 5.1.3.2. In this case, controller C3 is the only one that complies with the performance specifications set in Table 5.1.2.2 for the case study.

Indicator	C1	C2	C3	C4
$\Delta \alpha = \frac{\sum_{k=1}^{N_k} \alpha_k^d}{N_k}$	0.36	0.56	1.05	1.43
$\Delta \alpha = 1 - \frac{\sum_{k=1}^{N_k} (1 - \alpha_k^d) H (1 - \alpha_k^d)}{N_k}$	0.36	0.55	0.82	0.78
$\Delta\beta = \frac{\sum_{k=1}^{N_k} \beta_k^d}{N_k}$	0.44	0.68	1.01	1.23
$\Delta\beta = 1 - \frac{\sum_{k=1}^{N_k} (1 - \beta_k^d) H (1 - \beta_k^d)}{N_k}$	0.44	0.68	0.93	0.86

Table 5.1.3.2: Aggregate performance for controller prediction horizon selection

### 5.2. Performance monitoring on a CSTR

#### 5.2.1. Normal operation

In order to implement a performance monitoring application it is worth having a general grasp of what may be considered as normal operation of the system under study. For this reason, a simulation under ideal conditions, free of disturbances, is performed, which results are shown in Figure 5.2.1. The noisy readings from process measurements are shown in black, whereas the estimated signals are shown in blue and bounds in red.



Figure 5.2.1: Normal operation of the CSTR

Initially, components A and B are too concentrated in the reactor, for which the response of the controller is to reduce the reactor temperature, which would lead to a decrease in  $C_b$ . In order to counteract the resulting increase in  $C_a$  the input flow is also reduced. After this transient stage, it is observed that the system continuously cools down the reactor in order to maintain the temperature  $T_R$  close to 140° C, which would allow suitable reaction rates. Concentrations  $C_a$  and  $C_b$  are kept as close as possible to their set points taking into consideration that the cost for inputs utilization is not significantly increased.

#### 5.2.2. Definition and computation of benchmarks

As explained in section 3.3.2, for performance monitoring it is quite useful to define a so-called historical benchmark. For this case study, such benchmark may be conveniently defined taking into consideration the results obtained for normal operation of the CSTR. From the time response shown in Figure 5.2.1, the achieved cost function  $L_k^a$  can be computed. Results are shown in Figure 5.2.2. This  $L_k^a$  will be the base to define the historical benchmark for the system taking into consideration the time period after the transient stage. The expected value and standard deviation will be used to characterize the band of normal operation for this continuous process. Results are summarized in Table 5.2.2.1.



Figure 5.2.2: Achieved performance during normal operation of the CSTR

Variable	Value
$E[L_k^h]$	160.12
$std[L_k^h]$	25.15
$q_l[L_k^h] = E[L_k^h] - 2std[L_k^h]$	109.83
$q_h[L_k^h] = E[L_k^h] + 2std[L_k^h]$	210.41

Table 5.2.2.1: Historical benchmark for the CSTR

On the other hand, the benchmark  $L_{jk}^{b1}$  for fault detection was computed using the PCET toolbox. Computation results for scenario 5 are shown as an example in Table 5.2.2.2. The results obtained with the PCET toolbox show that there is no significant difference between the Monte Carlo method with 100 iterations and the polynomial chaos expansion of order 1 for this case study. Furthermore, the results from Table 5.2.2.3 show that for the configuration employed, any simulation case with  $N_{mc} \ge 4$  computed with the Galerkin method will be faster than with the Monte Carlo method.

## Table 5.2.2.2: Computation results for online performance

monitoring benchmark $L_{jk}^{b1}$	for the	CSTR
------------------------------------	---------	------

Variable $\theta = \{E[L_k^d], std[L_k^d]\}$	$E[L_{jk}^{b1}]$	$std[L_{jk}^{b1}]$
Theoretical method		
$\left(\frac{\sum_{k=1}^{N_k} \theta_{pcet}}{N_k}\right)$	136.8483	57.2905
Monte Carlo method		
$\left(\frac{\sum_{k=1}^{N_k} \theta_{mc}}{N_k}\right)$ , 100 iterations)	136.9260	57.2919
Mean percentage error		
$\sum_{k=1}^{N_k} \frac{\theta_{pcet} - \theta_{mc}}{\theta_{mc} N_k} x100\%$	-0.04	-0.003
Mean absolute percentage error		
$\sum_{k=1}^{N_k} \left  \frac{\theta_{pcet} - \theta_{mc}}{\theta_{mc} N_k} \right  x 100\%$	0.3	0.09

Table 5.2.2.3: Computation times for online performance

monitoring benchmark  $L_{jk}^{b1}$  for the CSTR

Variable	Value
Initialization	(1.5±0.4) s
$(t_{init}, setup compound model)$	. ,
Theoretical method	(0.30±0.04) s
$(t_{pcet})$	
Monte Carlo method	(0.12±0.04) s
$(t_{mc}, 1 \text{ iteration})$	(
Speed-up gain	
$\eta = \frac{N_{mc} t_{mc}}{t_{pcet}}$	$(0.4\pm0.1) \times N_{mc}$

#### 5.2.3. Fault detection

Once the benchmarks have been defined, the performance monitoring task can take place. Figure 5.2.3.1 shows the time response of the system under the process events from Table 4.2.1.



Figure 5.2.3.1: Operation of the CSTR under process events

During the cooling power fault between time t = 10 min and t = 15 min, the response of the controller is to set  $\dot{Q}$  to the lower bound. This action does not prevent the reactor temperature from rising, for which in order to maintain the concentrations close to their set points, the flux is increased correspondingly. During the flow actuator fault between time t = 20 min and t = 25 min, the mean value of the concentration  $C_a$  is slightly increased, while the reactor temperature is not significantly affected. The flux mean value is also not significantly modified, while some oscillations are observed mainly due to estimation errors. Such errors are also observed during the fault in the inflow concentration controller between time t = 30 min and t = 35 min. In this period,  $\dot{Q}$  is significantly increased in magnitude in order to maintain the reactor temperature level, while *F* is only slightly increased as the controller determines that an increase in the flux would be more detrimental to the system.

The achieved cost function is shown in Figure 5.2.3.2 along with the band that describes normal performance. The performance indices were computed with (52)-(55) considering an averaging window of 10 seconds. Results are shown in Figures 5.2.3.3 and 5.2.3.4.



Figure 5.2.3.2: Achieved performance during abnormal operation of the CSTR



Figure 5.2.3.3: α-Performance indices for performance monitoring on the CSTR



**Figure 5.2.3.4**:  $\beta$ -Performance indices for performance monitoring on the CSTR

From the figures it is clear that even in a case with considerable noise, it is possible to study the health of the system. The  $\alpha^{b}$ -index in Figure 5.2.3.3 clearly shows the estimation errors that occur during the first two minutes of operation and during the last two equipment faults. Furthermore, performance monitoring can be accomplished specially taking into consideration the  $\beta$ -indices as shown in Figure 5.2.3.4. The oscillations around t = 25 min lead to low values of the  $\beta^{h}$ -index based on the historical benchmark, while the model-based  $\beta^{b}$ -index still indicates a normal operation. As explained in section 3.3.2 this may be indicated as a minor fault. On the other hand, the fault in the inflow concentration would be indicated as a major fault as both indicators register a significantly low performance.

#### 5.2.4. Analysis of scenarios

As a corollary from the explanation in section 3.3.3, the degree of agreement of each prediction scenario with the actual achieved values in the process can be measured by (56). Therefore, it is possible to determine which scenarios represent more closely the current state of the process by means of performance indicators. Figures 5.2.4.1 and 5.2.4.2 show the averaged results computed using (56) for the case of normal operation.



**Figure 5.2.4.1**: Analysis of scenarios for normal operation with  $\alpha$ -index



**Figure 5.2.4.2**: Analysis of scenarios for normal operation with  $\beta$ -index

As shown in Table 4.2.4, scenario 5 corresponds to the case free of disturbances. Therefore, it is expected that figures 5.2.4.1 and 5.2.4.2 show relative high values for this scenario. Eventually, other scenarios may also have high values especially considering the randomness in the estimated states and all sources of delay in the system. A low weight based on the  $\beta$ -index may give

clear indication of disagreement between the scenario and the real process as its prediction cannot statistically explain the observations. On the other hand, weights based on  $\alpha$ -index may be more precise but also prone to noise. Tables 5.2.4.1 and 5.2.4.2 summarize the averaged weights for each scenario. The darker gray lines indicate the presence of process events during the respective time window, while bold numbers appear for the scenarios with the best fit for each window. Taking into consideration Tables 4.2.1 and 4.2.4, it can be verified that in most of the cases the weights represent correctly the correlation between the scenarios and the real status of the process as long as the indices  $\alpha_k^b$  and  $\beta_k^b$  are close to 1. If such indices are significantly far from 1, the results may no longer be reliable. An example of this is shown in the rows concerning the time window [30 35] in Table 5.2.4.1, in which a similar weight would be assigned to all scenarios.

t (min)	1	2	3	4	5	6	7	8	9
[0 10]	0.04	0.04	0.06	0.10	0.19	0.21	0.11	0.11	0.15
[10 15]	0.02	0.03	0.03	0.13	0.16	0.23	0.16	0.12	0.13
[15 20]	0.05	0.04	0.07	0.14	0.13	0.19	0.12	0.14	0.13
[20 25]	0.11	0.10	0.09	0.14	0.11	0.09	0.11	0.14	0.12
[25 30]	0.07	0.09	0.08	0.13	0.15	0.16	0.09	0.15	0.10
[30 35]	0.12	0.12	0.11	0.10	0.10	0.10	0.13	0.11	0.11
[35 45]	0.04	0.05	0.04	0.13	0.17	0.18	0.11	0.13	0.15

**Table 5.2.4.1**: Averaged weights with  $\alpha$ -index for CSTR monitoring

**Table 5.2.4.2**: Averaged weights with  $\beta$ -index for CSTR monitoring

t (min)	1	2	3	4	5	6	7	8	9
[0 10]	0.03	0.05	0.06	0.15	0.14	0.14	0.14	0.15	0.15
[10 15]	0.01	0.02	0.03	0.15	0.15	0.14	0.17	0.17	0.17
[15 20]	0.05	0.07	0.06	0.14	0.14	0.13	0.13	0.14	0.14
[20 25]	0.14	0.12	0.13	0.10	0.09	0.09	0.12	0.10	0.10
[25 30]	0.07	0.09	0.09	0.14	0.12	0.11	0.12	0.14	0.13
[30 35]	0.14	0.12	0.12	0.10	0.10	0.10	0.11	0.11	0.11
[35 45]	0.04	0.05	0.06	0.15	0.14	0.14	0.13	0.15	0.15

## 5.3. Performance monitoring on a polymerization reactor

### 5.3.1. Normal operation

A simulation of a batch run under ideal conditions is performed and the results are shown in Figure 5.3.1.



Figure 5.3.1: Normal operation of the polymerization reactor

After an initial peak due to estimation errors, the monomer feed slowly increases until it reaches its maximum rate at time t = 60 min. The input temperatures oscillate permanently in order to maintain the reactor temperature close to the upper soft constraint, as this condition provides the best conversion without incurring in significant penalizations. The sudden feed reductions at times t = 39 min and t = 53 min allow less oscillation in the input temperatures. During the last 20 minutes of operation the benefit for producing polymer exceeds the penalization due to the soft constraints, leading to relative high values of the reactor temperature. The batch time reported was 90 minutes.

### 5.3.2. Definition and computation of benchmarks

From the time response shown in Figure 5.3.1, the achieved cost function  $L_k^a$  can be computed. A smoothened version of the achieved performance in normal operation along with a band to account for possible random events will be considered as the historical benchmark for the polymerization reactor. Figure 5.3.2 shows  $-L_k^a$  for better inspection of the maximization results. Furthermore, the computation time for the benchmark  $L_{jk}^{bK}$  is reported as  $t_{mc} = (0.21 \pm 0.02)s$  for one Monte Carlo simulation.



Figure 5.3.2: Achieved performance during normal operation of the polymerization reactor

### 5.3.3. Fault detection

Figure 5.3.3.1 shows the time response of the system under the process events from Table 4.3.1.



Figure 5.3.3.1: Operation of the polymerization reactor under process events

During the feed system fault, the controller is forced to reduce the feed abruptly. Contrary to the previous case, the input temperatures keep oscillating to maintain the reactor temperature close to the upper soft constraint. Another considerable feed reduction is observed during the input temperature  $T_M^{IN}$  fault, while the fault in  $T_{AWT}^{IN}$  had only a minor effect in the time response of the system. The batch time reported was 94 minutes. The achieved cost function is shown in Figure 5.3.3.2 along with the band that describes normal performance.



Figure 5.3.3.2: Achieved performance during abnormal operation of the polymerization reactor

The inverse of the performance indices computed with (52)-(55) are shown in Figures 5.3.3.3 and 5.3.3.4. As the objective of this case study is to maximize a cost function, the inverse of the performance indices relate better to the interpretation given before. Initial estimation errors lead to low values of the  $\alpha^{h}$ -index for the test case during the first 30 minutes of operation. However, considering the  $\beta^{h}$ -index, this relative low performance can still be considered as normal. From time t = 35 min, the performance of the system is slowly reduced until t = 50 min. After this time, the performance indices relative to the historical benchmark maintain an approximate value of 0.9 until the end of the batch. From time t = 43 min, a minor fault in the system can be indicated considering the readings from the model-based indices.



**Figure 5.3.3.3**: *α*-Performance indices for performance monitoring on the polymerization reactor



**Figure 5.3.3.4**:  $\beta$ -Performance indices for performance monitoring on the polymerization reactor

Close observation of these indices will show, however, that they have a small permanent positive bias, which can be explained by estimation errors, synchronization issues and database access delays. These accumulated delays cause a consistent time difference between the estimated states and the measurements, which leads to an index bias after performing the predictions. This bias is clearly seen in this case as the main component of the cost function, the polymer mass, grows continuously and significantly with time as shown in Figure 5.3.3.5. Furthermore, in order to perform a proper analysis of scenarios, this delay effect should be corrected in the real-time modules taking into consideration the time stamps given by the database.



Figure 5.3.3.5: Effect of delays on the predictions

## 6. Conclusions

In this thesis some methodologies to evaluate systematically the performance of model predictive controllers have been proposed with special consideration for applications based on multi-stage NMPC.

After a thorough literature review, a performance measure based on the optimizer stage cost was chosen. This represents a natural selection in order to conveniently monitor if the controller is doing what it was designed to do. In order to draw conclusions from the monitoring, the achieved performance is compared with a model-based benchmark, which represents a reference of the expected achievable performance of the system by design and which can be computed performing simulations of the system considering a full process model. The resulting benchmark is in general a stochastic variable characterized by its expected value, lower and upper bounds, from which performance indices can be defined in order to simplify the interpretation of results. In order to avoid noisy readings, it is convenient to define time-averaged versions of the performance indices.

In the general case, the design benchmark must be computed by means of Monte Carlo simulations. Nonetheless, for some special cases more computationally efficient techniques can be proposed. A theoretical procedure based on fundamental properties of normal distributions is explained for the case of linear systems with normally-distributed uncertainties. Furthermore, the use of the polynomial chaos expansion theory by means of the PCET toolbox provides an efficient mechanism to solve nonlinear stochastic differential equations with polynomial nonlinearities. Both procedures led to significant computation speed up gains with respect to the conventional Monte Carlo method with similar numerical results. For practical purposes, however, it must be taken into consideration, that the polynomial chaos expansion method is not easily scalable as the resulting system of equations grows significantly fast with the number of uncertain variables, for which only the most relevant ones should be included in the formulation in order to avoid problems related to insufficient computing resources. Moreover, the PCET toolbox requires models to include only polynomial nonlinearities, which may introduce considerable model-plant mismatch if the model conversion is not properly performed.

On the other hand, a procedure based on Markov chains is proposed in order to handle cases with complex distributions such as equipment malfunctions. Incorporating maintenance information, this technique may be used to provide realistic predictions as part of an operators training program. Such technique is envisioned to be used especially on cases with slow dynamics and rather long prediction horizons, so that the effects of equipment failure are meaningful and the computation time requirements of the Monte Carlo method are fulfilled.

The range of possible applications for these methodologies is wide, covering not only control performance monitoring tasks, but also providing support in the controller design stage. As an example, a performance-based procedure for controller selection is presented and illustrated with a quadruple tank system as case study. Given performance specifications, this procedure allows a systematic comparison of controllers in order to validate only those that comply with the desired requirements. Process insight must be provided in order to define a reasonable target performance that can be achieved and a sufficient number of criteria in order to facilitate the selection procedure. The examples presented show cases in which the performance difference between some controllers is significant, while relatively small among others, so that it is difficult to select one univocally.

A performance monitoring technique is illustrated with two case studies regarding a continuous and a batch reactor. Employing a historical and a model-based benchmark, it is possible to compute indices that would allow operators to distinguish between normal and abnormal operation, as well as to classify the type of faults encountered. Considering their robustness to noise, the  $\beta$ -indices are especially suitable for this application. Furthermore, a procedure for dynamic tuning of the optimizer based on performance indices is proposed, for which the  $\alpha$ -indices prove to be quite useful considering their

83

sensitivity to the different scenarios. For this application, however, the monitoring indices  $\alpha_k^b$  and  $\beta_k^b$  are required to be close to 1. Some consideration must be taken to compensate for noise and delay effects in a real-time implementation as these would lead to incorrect predictions, hindering the application of the performance indices for a specific case study.

The performance assessment techniques proposed in this thesis are expected to contribute towards better understanding of the MPC industrial technology, as well as to improve the design and analysis of such advanced controllers.

## 7. Future work

Some progress has been accomplished on the evaluation of model predictive controllers. Nonetheless, some issues remain open for further investigation.

Performance indicators that measure predicted violations of soft constraints may be quite useful to quantify the stability of a system especially in economic applications. Furthermore, an analysis of the computation times of the optimizer may shed some light on the current status of the system.

Further methodologies must be explored in order to solve efficiently stochastic differential equations such as the Euler-Maruyama and the Milstein method. Monte Carlo methods may be further developed by means of parallel computing techniques.

Different technologies of industrial databases and networks should be studied. Proper mechanisms to compensate for communication and access delays must be proposed for each case.

As an extension to the tuning methodology presented, the weights of the scenarios may be adjusted according to the user needs. For this purpose, a configuration parameter may be defined in the range [0 1], so that robustness and performance represent the extreme values. Furthermore, the effects of such online tuning should be tested.

Performance deterioration due to common malfunctions such as sensor faults, sensor drifts and actuators that temporarily do not react to control signals should be studied for industrial case studies.

A natural extension of performance assessment consists in the identification of such faults, therefore achieving a fault diagnosis system. Some techniques have been proposed for linear systems (Botelho, 2016), which may be extended to the nonlinear case.

# 8. Appendix

# A. Configuration parameters of case studies

## A.1. Controller selection for a 4-Tank system

### A.1.1. Process module

Table A.1.1.1: Nominal	parameter values	for 4-Tank system
	paramotor varaoo	ion i ranne oyotonn

Parameter	Description	Value	Units
A <sub>s1</sub>	Cross sectional area of tank 1	2.80x10 <sup>3</sup>	cm <sup>2</sup>
$A_{s2}$	Cross sectional area of tank 2	3.20x10 <sup>3</sup>	cm <sup>2</sup>
A <sub>s3</sub>	Cross sectional area of tank 3	2.80x10 <sup>3</sup>	cm <sup>2</sup>
$A_{s4}$	Cross sectional area of tank 4	3.20x10 <sup>3</sup>	cm <sup>2</sup>
C <sub>d1</sub>	Natural flow coefficient 1	3.15	cm <sup>2.5</sup> / s
C <sub>d2</sub>	Natural flow coefficient 2	2.53	cm <sup>2.5</sup> / s
C <sub>d3</sub>	Natural flow coefficient 3	3.15	cm <sup>2.5</sup> / s
C <sub>d4</sub>	Natural flow coefficient 4	2.23	cm <sup>2.5</sup> / s
<i>k</i> <sub>1</sub>	Ratio flux to voltage in pump 1	12.56	cm <sup>3</sup> / V.s
k <sub>2</sub>	Ratio flux to voltage in pump 2	13.16	cm <sup>3</sup> / V.s
e <sub>p1</sub>	Exponent for natural flow 1	0.50	-
e <sub>p2</sub>	Exponent for natural flow 2	0.50	-
e <sub>p3</sub>	Exponent for natural flow 3	0.50	-
$e_{p4}$	Exponent for natural flow 4	0.50	-

Table A.1.1.2: Initial values for 4-Tank system

Variable	Value	Units
$\Delta x_1$	0.5	cm
$\Delta x_2$	0.8	cm
$\Delta x_3$	-0.4	cm
$\Delta x_4$	-0.7	cm
$\Delta u_1$	0.4	V
$\Delta u_2$	0.3	V

$\Delta u_3$	0.05	-
$\Delta u_4$	-0.05	-

Variable	Value	Units
<i>x</i> <sub>s1</sub>	201.49	cm
<i>x</i> <sub>s2</sub>	214.77	cm
<i>x</i> <sub>s3</sub>	75.68	cm
<i>x</i> <sub><i>s</i>4</sub>	106.02	cm
<i>u</i> <sub>s1</sub>	3.20	V
<i>u</i> <sub>s2</sub>	3.15	V
<i>u</i> <sub>s3</sub>	0.43	-
<i>u</i> <sub>s4</sub>	0.34	-

Table A.1.1.3: Equilibrium point for 4-Tank system

 Table A.1.1.4: Noise distribution for 4-Tank system

Variable	Distribution	Units
<i>w</i> <sub>1</sub>	N(0,0.005 <sup>2</sup> )	cm.s <sup>-1</sup>
<i>w</i> <sub>2</sub>	N(0,0.005 <sup>2</sup> )	cm.s <sup>-1</sup>
<i>W</i> <sub>3</sub>	$N(0, 0.005^2)$	cm.s <sup>-1</sup>
<i>W</i> <sub>4</sub>	N(0,0.005 <sup>2</sup> )	cm.s⁻¹
<i>v</i> <sub>1</sub>	N(0,0.05 <sup>2</sup> )	cm
$v_2$	N(0,0.05 <sup>2</sup> )	cm
<i>v</i> <sub>3</sub>	N(0,0.05 <sup>2</sup> )	cm
$v_4$	$N(0, 0.05^2)$	cm

## A.1.2. Estimator module

	Table A.1.2.1:	Initial values	for estimator	for 4-Tank systen
--	----------------	----------------	---------------	-------------------

Variable	Value	Units
$\Delta \hat{x}_{01}$	0.2	ст
$\Delta \hat{x}_{02}$	-0.1	cm
$\Delta \hat{x}_{03}$	0.1	cm

$\Delta \hat{x}_{04}$	-0.4	cm
<i>P</i> <sub>x01</sub>	0.01	cm <sup>2</sup>
<i>P</i> <sub>x02</sub>	0.01	cm <sup>2</sup>
<i>P</i> <sub>x03</sub>	0.01	cm <sup>2</sup>
<i>P</i> <sub>x04</sub>	0.01	cm <sup>2</sup>

Table A.1.2.2: Noise covariance matrices for estimator for 4-Tank system

Variable	Value	Units
$Q_{w1}$	1.25x10 <sup>-4</sup>	cm <sup>2</sup>
$Q_{w2}$	1.25x10 <sup>-4</sup>	cm <sup>2</sup>
$Q_{w3}$	1.25x10 <sup>-4</sup>	cm <sup>2</sup>
$Q_{w4}$	1.25x10 <sup>-4</sup>	cm <sup>2</sup>
$R_{v1}$	5x10 <sup>-4</sup>	cm <sup>2</sup>
$R_{v2}$	5x10 <sup>-4</sup>	cm <sup>2</sup>
$R_{\nu 3}$	5x10 <sup>-4</sup>	cm <sup>2</sup>
$R_{v4}$	5x10 <sup>-4</sup>	cm <sup>2</sup>

# A.2. Performance monitoring on a CSTR

### A.2.1. Process module

Table A.Z. I. I. NOTITIAL PARAMETER VALUES TO COTR	Table A	<b>A.2.1.1</b> :	Nominal	parameter	values	for CSTR
--	---------	------------------	---------	-----------	--------	----------

Parameter	Description	Value	Units
$\alpha^{ad}$	Corrective factor for $E_A^{ad}$	0.90	-
$\beta^{ab}$	Corrective factor for $K_0^{ab}$	1.00	-
$K_0^{ab}$	Pre-exponential factor for reaction A→B	1.29x10 <sup>12</sup>	h⁻¹
$K_0^{bc}$	Pre-exponential factor for reaction $B \rightarrow C$	1.29x10 <sup>12</sup>	h⁻¹
$K_0^{ad}$	Pre-exponential factor for reaction $2A \rightarrow D$	9.04x10 <sup>9</sup>	mol <sup>-1</sup> h <sup>-1</sup>
$E_A^{ab}$	Activation energy for reaction $A \rightarrow B$	9.76x10 <sup>3</sup>	kJ.mol⁻¹
$E_A^{bc}$	Activation energy for reaction B→C	9.76x10 <sup>3</sup>	kJ.mol⁻¹
$E_A^{ad}$	Activation energy for reaction $2A \rightarrow D$	8.56x10 <sup>3</sup>	kJ.mol⁻¹
$H_R^{ab}$	Enthalpy for reaction A→B	4.20	kJ.mol <sup>-1</sup>
$H_R^{bc}$	Enthalpy for reaction $B \rightarrow C$	-11.00	kJ.mol⁻¹

$H_R^{ad}$	Enthalpy for reaction 2A→D	-41.85	kJ.mol <sup>-1</sup>
ρ	Density	0.93	kg.L⁻¹
Cp	Specific heat capacity	3.01	kJ.kg⁻¹.K⁻¹
C <sub>p,K</sub>	Coolant heat capacity	2.00	kJ.kg <sup>-1</sup> .K <sup>-1</sup>
$A_R$	Area of reactor wall	0.22	m²
V <sub>R</sub>	Volume of reactor	10.01	L
$m_K$	Coolant mass	5.00	kg
T <sub>in</sub>	Inflow temperature	130.00	°C
K <sub>w</sub>	Convection coefficient	4.03x10 <sup>3</sup>	kJ.h <sup>-1</sup> .m <sup>2</sup> .K <sup>-1</sup>
C <sub>a,in</sub>	Inflow concentration of A	5.10	mol/L

### Table A.2.1.2: Initial values for CSTR

Variable	Value	Units
<i>x</i> <sub>1</sub>	2	mol/L
<i>x</i> <sub>2</sub>	0.65	mol/L
<i>x</i> <sub>3</sub>	145	٥C
<i>x</i> <sub>4</sub>	100	٥C
<i>u</i> <sub>1</sub>	200	m <sup>3</sup> .h <sup>-1</sup>
<i>u</i> <sub>2</sub>	-4x10 <sup>3</sup>	kJ.h <sup>-1</sup>

## Table A.2.1.3: Noise distribution for CSTR

Variable	Distribution	Units
<i>W</i> <sub>1</sub>	N(0, 0.4 <sup>2</sup> )	mol.L <sup>-1</sup> .h <sup>-1</sup>
<i>W</i> <sub>2</sub>	N(0, 0.4 <sup>2</sup> )	mol.L <sup>-1</sup> .h <sup>-1</sup>
<i>W</i> <sub>3</sub>	N(0, 40 <sup>2</sup> )	°C.h <sup>-1</sup>
<i>W</i> <sub>4</sub>	N(0, 40 <sup>2</sup> )	°C.h⁻¹
$v_1$	N(0,0.02 <sup>2</sup> )	mol.L <sup>-1</sup>
$v_2$	N(0,0.02 <sup>2</sup> )	mol.L <sup>-1</sup>
v <sub>3</sub>	N(0, 2 <sup>2</sup> )	°C
$v_4$	N(0, 2 <sup>2</sup> )	٥C

### A.2.2. Estimator module

Variable	Value	Units
$\hat{x}_{01}$	1.9	mol/L
$\hat{x}_{02}$	0.7	mol/L
$\hat{x}_{03}$	150	°C
$\hat{x}_{04}$	90	°C
$\hat{x}_{05}$	1	-
$\hat{x}_{06}$	1	-
$P_{x01}$	0.05	mol <sup>2</sup> /L <sup>2</sup>
$P_{x02}$	0.05	mol <sup>2</sup> /L <sup>2</sup>
$P_{x03}$	2	°C <sup>2</sup>
$P_{x04}$	2	°C <sup>2</sup>
$P_{x05}$	0.01	-
$P_{x06}$	0.01	-

Table A.2.2.1: Initial values for estimator for CSTR

Table A.2.2.2: Noise covariance matrices for estimator for CSTR

Variable	Value	Units
$Q_{w1}$	4x10 <sup>-5</sup>	mol <sup>2</sup> /L <sup>2</sup>
$Q_{w2}$	4x10 <sup>-5</sup>	mol <sup>2</sup> /L <sup>2</sup>
$Q_{w3}$	0.4	°C2
$Q_{w4}$	0.4	°C <sup>2</sup>
$Q_{w5}$	4x10 <sup>-5</sup>	-
$Q_{w6}$	4x10 <sup>-5</sup>	-
$R_{v1}$	4x10 <sup>-4</sup>	mol <sup>2</sup> /L <sup>2</sup>
$R_{v2}$	4x10 <sup>-4</sup>	mol <sup>2</sup> /L <sup>2</sup>
$R_{\nu 3}$	4	°C <sup>2</sup>
$R_{v4}$	4	°C <sup>2</sup>

# A.3. Performance monitoring on a polymerization reactor

### A.3.1. Process module

Parameter	Description	Value	Units
R	Gas constant	8.31	kJ.kmol <sup>-</sup> 1.K <sup>-1</sup>
$c_{p,W}$	Specific heat capacity of the coolant	4.20	kJ.kg⁻¹.K⁻¹
C <sub>p,S</sub>	Specific heat capacity of steel	0.47	kJ.kg⁻¹.K⁻¹
C <sub>p,F</sub>	Specific heat capacity of the feed	3.00	kJ.kg <sup>-1</sup> .K <sup>-1</sup>
C <sub>p,R</sub>	Specific heat capacity in the reactor	5.00	kJ.kg⁻¹.K⁻¹
k <sub>WS</sub>	Heat transfer coefficient water-steel	4.80x10 <sup>3</sup>	W.m <sup>-2</sup> .K <sup>-1</sup>
$T_F$	Feed temperature	25.00	°C
A	Heat exchange area in the jacket	65.00	m <sup>2</sup>
$m_{M,KW}$	Mass of coolant in the jacket	5.00x10 <sup>3</sup>	kg
m <sub>s</sub>	Mass of reactor steel	3.90x10 <sup>4</sup>	kg
m <sub>AWT</sub>	Mass of the product in the heat exchanger	2.00x10 <sup>2</sup>	kg
m <sub>AWT,KW</sub>	Mass of the coolant in the heat exchanger	1.00x10 <sup>3</sup>	kg
$\dot{m}_{M,KW}$	Coolant flow in the jacket	3.00x10 <sup>5</sup>	kg.h⁻¹
$\dot{m}_{AWT,KW}$	Coolant flow of the heat exchanger	1.00x10 <sup>5</sup>	kg.h⁻¹
<i>ṁ</i> <sub>AWT</sub>	Product flow to the heat exchanger	2.00x10 <sup>4</sup>	kg.h⁻¹
Ea	Activation energy	8.50x10 <sup>3</sup>	kJ.kmol <sup>-1</sup>
$\Delta H_R$	Specific reaction enthalpy	9.50x10 <sup>2</sup>	kJ.kg⁻¹
$k_0$	Specific reaction rate	7.00	-
k <sub>U2</sub>	Reaction parameter 1	32.00	-
k <sub>U1</sub>	Reaction parameter 2	4.00	-
$\omega_{W,F}$	Mass fraction of water in feed	0.33	-
$\omega_{A,F}$	Mass fraction of monomer in feed	0.67	-
k <sub>AS</sub>	Heat transfer coefficient monomer-steel	1.00x10 <sup>3</sup>	W.m <sup>-2</sup> .K <sup>-1</sup>
k <sub>PS</sub>	Heat transfer coefficient product-steel	100.00	W.m <sup>-2</sup> .K <sup>-1</sup>
α <sup>exp</sup>	Experimental coefficient	3.60x10 <sup>6</sup>	kJ.h <sup>-1</sup> .K <sup>-1</sup>

 Table A.3.1.1: Nominal parameter values for batch polymerization reactor

Variable	Value	Units
<i>x</i> <sub>1</sub>	10 <sup>5</sup>	kg
<i>x</i> <sub>2</sub>	853	kg
<i>x</i> <sub>3</sub>	26.5	kg
<i>x</i> <sub>4</sub>	361.15	°C
<i>x</i> <sub>5</sub>	363.15	°C
<i>x</i> <sub>6</sub>	363.15	°C
x <sub>7</sub>	308.15	°C
<i>x</i> <sub>8</sub>	308.15	°C
<i>x</i> 9	300	°C
<i>x</i> <sub>10</sub>	378.05	kg
<i>u</i> <sub>1</sub>	0	kg.h⁻¹
<i>u</i> <sub>2</sub>	363	°C
<i>u</i> <sub>3</sub>	323	℃

Table A.3.1.2: Initial values for batch polymerization reactor

Table A.3.1.3: Noise distribution for batch polymerization reactor

Variable	Distribution	Units
<i>w</i> <sub>1</sub>	N(0,0.01 <sup>2</sup> )	kg
<i>w</i> <sub>2</sub>	N(0,0.01 <sup>2</sup> )	kg
<i>w</i> <sub>3</sub>	N(0,0.01 <sup>2</sup> )	kg
<i>W</i> <sub>4</sub>	N(0, 1 <sup>2</sup> )	٥C
<i>w</i> <sub>5</sub>	<i>N</i> (0, 3 <sup>2</sup> )	٥C
w <sub>6</sub>	<i>N</i> (0, 3 <sup>2</sup> )	٥C
<i>W</i> <sub>7</sub>	<i>N</i> (0, 3 <sup>2</sup> )	٥C
<i>w</i> <sub>8</sub>	<i>N</i> (0, 3 <sup>2</sup> )	٥C
W9	<i>N</i> (0, 10 <sup>2</sup> )	٥C
W <sub>10</sub>	N(0, 3 <sup>2</sup> )	kg
<i>v</i> <sub>1</sub>	N(0,0.001 <sup>2</sup> )	kg
v <sub>2</sub>	N(0,0.001 <sup>2</sup> )	kg
v <sub>3</sub>	N(0,0.001 <sup>2</sup> )	kg
$v_4$	N(0, 0.1 <sup>2</sup> )	°C

v <sub>5</sub>	N(0, 0.3 <sup>2</sup> )	°C
v <sub>6</sub>	$N(0, 0.3^2)$	°C
v <sub>7</sub>	$N(0, 0.3^2)$	°C
v <sub>8</sub>	N(0, 0.3 <sup>2</sup> )	°C
v <sub>9</sub>	N(0,1 <sup>2</sup> )	°C
v <sub>10</sub>	$N(0, 0.3^2)$	kg

## A.3.2. Estimator module

Table A.3.2.1: Initial values for estimator for batch polymerization reac	tor
---	-----

Variable	Value	Units
$\hat{x}_{01}$	8000	kg
$\hat{x}_{02}$	1000	kg
$\hat{x}_{03}$	50	kg
$\hat{x}_{04}$	365	°C
$\hat{x}_{05}$	350	°C
$\hat{x}_{06}$	350	٥C
$\hat{x}_{07}$	350	°C
$\hat{x}_{08}$	350	٥C
$\hat{x}_{09}$	400	°C
<i>x</i> <sub>010</sub>	380	kg
<i>x</i> <sub>011</sub>	1	-
<i>x</i> <sub>012</sub>	1	-
<i>P</i> <sub>x01</sub>	10 <sup>-5</sup>	kg²
<i>P</i> <sub>x02</sub>	10 <sup>-5</sup>	kg <sup>2</sup>
<i>P</i> <sub>x03</sub>	10 <sup>-5</sup>	kg <sup>2</sup>
$P_{x04}$	0.1	°C2
$P_{x05}$	1	°C2
<i>P</i> <sub>x06</sub>	1	°C2
<i>P</i> <sub>x07</sub>	1	°C2
<i>P</i> <sub>x08</sub>	1	°C2
<i>P</i> <sub>x09</sub>	1	°C²

<i>P</i> <sub>x010</sub>	1	kg <sup>2</sup>
<i>P</i> <sub>x011</sub>	0.1	-
<i>P</i> <sub>x012</sub>	0.1	-

Variable	Value	Units
$Q_{w1}$	10 <sup>-7</sup>	kg <sup>2</sup>
$Q_{w2}$	10 <sup>-7</sup>	kg <sup>2</sup>
$Q_{w3}$	10 <sup>-7</sup>	kg <sup>2</sup>
$Q_{w4}$	0.001	°C2
$Q_{w5}$	0.01	°C2
$Q_{w6}$	0.01	°C2
$Q_{w7}$	0.01	°C <sup>2</sup>
$Q_{w8}$	0.01	°C2
$Q_{w9}$	0.1	°C <sup>2</sup>
$Q_{w10}$	0.01	kg <sup>2</sup>
<i>Q</i> <sub>w11</sub>	0.1	-
$Q_{w12}$	0.1	-
$R_{v1}$	10 <sup>-6</sup>	kg <sup>2</sup>
$R_{v2}$	10 <sup>-6</sup>	kg <sup>2</sup>
$R_{v3}$	10 <sup>-6</sup>	kg <sup>2</sup>
$R_{v4}$	0.01	°C2
$R_{v5}$	0.1	°C2
$R_{v6}$	0.1	°C <sup>2</sup>
$R_{v7}$	0.1	°C <sup>2</sup>
$R_{v8}$	0.1	°C2
$R_{v9}$	1	°C <sup>2</sup>
$R_{v10}$	0.1	kg <sup>2</sup>

# 9. Bibliography

K.J. Åström. *Introduction to Stochastic Control Theory*. Academic Press, San Diego, California, 1970.

V. Botelho, J. O. Trierweiler, M. Farenzena and R. Duraiski. Perspectives and challenges in performance assessment of model predictive control. The Canadian Journal of Chemical Engineering, 94: 1225-1241, 2016.

P. Brandimarte. *Handbook in Monte Carlo Simulation*. John Wiley and Sons, New Jersey, 2014.

C. Dai and S. H. Yang. Controller performance assessment with a LQG benchmark obtained by using the subspace method. Proceedings of Control, 2004.

M. Ellis and P. D. Christofides. Performance monitoring of economic model predictive control systems. Industrial and Engineering Chemistry Research, 53(40): 15406–15413, 2014.

T.J. Harris. Assessment of control loop performance. The Canadian Journal of Chemical Engineering, 67: 856-861, 1989.

T. J. Harris, C.T. Seppala, P.J. Jofriet and L.D. Desborough. A review of performance monitoring and assessment techniques for univariate and multivariate control systems. Journal of Process Control, 9(1): 1-17, 1999.

T. J. Harris. Statistical properties of quadratic-type performance indices. Journal of Process Control, 14(8): 899 – 914, 2004.

B. Huang and S. L. Shah. *Performance Assessment of Control Loops*. Springer-Verlag, London, 1999.

B. Huang and R. Kadali. Estimation of the dynamic matrix and noise model for model predictive control using closed-loop data. Industrial and Engineering Chemistry Research, 41: 842–852, 2002.

B. Huang and R. Kadali. *Dynamic Modeling, Predictive Control and Performance Monitoring*. Springer-Verlag, London, 2008.

M. Kvasnica, M. Herceg, L. Cirka, M. Fikar. Model predictive control of a CSTR. Chemical Papers, 64(3): 301-309, 2010.

C. Liu, B. Huang and Q. Wang. Control performance assessment subject to multi-objective user-specified performance characteristics. IEEE Transactions on Control Systems Technology, 19(3): 682-691, 2011.

S. Lucía, A. Tatulea-Codrean, C. Schoppmeyer, S. Engell. An environment for the efficient testing and implementation of robust NMPC. IEEE Conference on Control Applications, 1843-1848, 2014.

S. Lucía, J. A. E. Andersson, H. Brandt, M. Diehl, S. Engell. Handling uncertainty in economic nonlinear model predictive control: A comparative case study. Journal of Process Control, 24: 1247-1259, 2014.

A. M. Mathai and S. B. Provost. *Quadratic Forms in Random Variables*. Marcel Dekker, New York, 1992.

R. S. Patwardhan. *Studies in Synthesis and Analysis of Model Predictive Controllers*. PhD thesis, University of Alberta, 1999. URL: http://www.collectionscanada.ca/obj/s4/f2/dsk1/tape8/PQDD\_0016/NQ46902.pdf.

F. Petzke, A. Mesbah, S. Streif. Polynomial Chaos Expansion Toolbox for Matlab, 2105. URL: http://www.TU-Chemnitz.de/etit/control/PCET/ Technische Universität Chemnitz, Laboratory for Automatic Control and System Dynamics, Germany.

96

J. Schäfer and A. Cinar. Multivariable MPC performance assessment, monitoring and diagnosis. Proceedings IFAC world congress, 429-434, 2002.

S. Streif, F. Petzke, A. Mesbah, R. Findeisen, R. D. Braatz. Optimal experimental design for probabilistic model discrimination using polynomial chaos. Proceedings IFAC world congress, 4103-4109, 2014.

Y. L. Tong. *The Multivariable Normal Distribution*. Springer-Verlag, New York, 1990.

N. Wiener. The homogeneous chaos. American Journal of Mathematics, 60: 897-936, 1938.

D. Xiu and G. E. Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. SIAM Journal of Scientific Computation, 24: 619-644, 2002.

Q. Yuan, B. Lennox and M. McEwan. Analysis of multivariable control performance assessment techniques. Journal of Process Control, 19(5): 751-760, 2009.

M. A. Zagrobelny. *MPC Performance Monitoring and Disturbance Model Identification*. PhD thesis, University of Wisconsin-Madison, 2014. URL: http://jbrwww.che.wisc.edu/theses/zagrobelny.pdf

Y. Zhang and M. A. Henson. A performance measure for constrained model predictive controllers. European Control Conference, 918-923, 1999.

97