Mirco Lupi

Disturbance Models for Offset-free Nonlinear Predictive Control

Master of Science Thesis in Chemical Engineering

University of Pisa
Department of Civil and Industrial Engineering

10/05/2016

University of Pisa
Department of Civil and Industrial Engineering

Master of Science Degree in Chemical Engineering

Master of Science Thesis

Disturbance Models for Offset-free Nonlinear Predictive Control

Thesis advisor:
Prof. Ing. Gabriele Pannocchia ....

Examiner:
Prof. Ing. Claudio Scali ....

Candidate:
Mirco Lupi ....

Thesis submitted in 05/2016
Academic Year 2014/2015
Consultation allowed

Università di Pisa
Dipartimento di Ingegneria Civile e Industriale

Tesi di Laurea Magistrale in Ingegneria Chimica

Tesi di Laurea Magistrale

Modelli di Disturbo per Controllo Predittivo Non Lineare

Relatore:
Prof. Ing. Gabriele Pannocchia ....

Controrelatore:
Prof. Ing. Claudio Scali ....

Candidato:
Mirco Lupi ....

Tesi presentata nel 05/2016
Anno Accademico 2014/2015
Consultazione consentita

ABSTRACT

Offset-free model predictive control refers to a class of control algorithms able to track asymptotically constant reference signals despite the presence of unmeasured, nonzero mean disturbance acting on the process and/or plant model mismatch. Generally, in these formulations the nominal model of the plant is augmented with integrating disturbances, i.e. with a properly designed disturbance model, and state and disturbance are estimated from output measurements. To date the vast majority of offset-free MPC applications are based on linear models, however, since process dynamics are generally inherently nonlinear, these may perform poorly or even fail in some situations. Better results can be achieved by making use of nonlinear formulations and hence of nonlinear model predictive control (NMPC) technology. However, the obstacles associated with implementing NMPC frameworks are nontrivial. In this work the offset-free tracking problem with nonlinear models is addressed. Firstly some basic concepts related to the observability of nonlinear systems and state estimation are reviewed, focusing on the digital filtering and putting a strong accent on the role of the disturbance model. Thus, a class of disturbance models in which the integrated term is added to model parameters is presented together with an efficient and practical strategy for its design and subsequent implementation in offset-free NMPC frameworks. Extensive simulation results are presented to show the benefits of the resulting NMPC framework over standard ones.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF CODE

NOTATIONS

Upper Case Symbols

F	state transition augmented model
G	augmented input function
H	output augmented model
I	identity matrix
J^{lqr}	linear quadratic regulation problem objective function
K_{LQ}	linear quadratic controller gain
L	disturbance selection matrix
L_{ϕ}	upper bound of the rest function
V_f	terminal cost function
X	sequence of states
Y	sequence of output measurements
A, B, C, G, F, B_d, C_d	linearized system matrices
E^n	set of n standard unit vectors
K	Kalman gain matrix
K^*	gain matrix
M	set of perturbation sizes
N	prediction horizon
P	estimation error covariance matrix
Q	process noise covariance matrix
Q_{min}, Q_{max}	matrices containing the bounds on the process noise covariances
Q_{reg}, R_{reg}	controller tuning matrices
R	sensor noise covariance matrix
T^n	set of perturbation directions
T_i	i-th perturbation direction
U	sequence of inputs
W_O	observability covariance matrix\empirical observability gramian

Lower Case Symbols

c_i	i-th perturbation size
d	disturbance vector of system
Δt	sampling time interval
Δu	input variation between two sampling times
e	output prediction error
e_i	i-th unit vector
f	state transition model
g	input function
h	output model
k	number of sampling step
m	input dimension of system
n	state dimension of system
n_d	disturbance dimension of system
n_{θ}	nominal model parameters vector dimension of system

n_v	sensor noise dimension of system
n_w	process noise dimension of system
p	output dimension of system
$p\left(x\|y\right)$	conditional probability density function of x given y
r	number of matrices for perturbation direction
r_y	controlled output reference function
s	number of different perturbation sizes
ℓ	stage cost function
t	continuous time
u	input vector of system
v	sensor noise vector of system
w	process noise vector of system
x	state vector of system
y	output vector of system
$\bar{y}_c$	controlled output set-point
g_u	input constraint function
g_y	output constraint function
p_c	controlled output dimension of system
q_u	input constraint dimension of system
q_y	output constraint dimension of system
t^*	time instant
$\tilde{u}^{\bullet}$	deterministic signal
$\tilde{u}^{in}$	innovation term
y^{ilm}	output corresponding to a perturbed initial condition
y_{ss}^{ilm}	steady-state value of the output reached after the perturbation
y_s^{jlm}	steady-state value of the output before the perturbation

Greek Letters

α	generic constant
δ	differential
ϵ	estimation error vector
Γ	augmented state vector formed by state and input variations
κ	gain function
λ	eigenvalue vector
μ_i	i-th degree of observability measure
ϕ	rest function
Π	solution of the DARE
ρ	spectral radius
σ	singular value vector
τ	time instant in a given time interval
θ_0	nominal model parameters vector
Θ	observability mapping
ξ	random variable
ζ	augmented state vector formed by state and disturbance

Script Symbols

$\mathcal{C}$	controllability matrix
$\delta\mathcal{G}$	observability codistribution

$\mathcal{F}$	generic function
$\mathcal{G}$	observation space
$\mathcal{K}$	class kappa function
$\mathcal{KL}$	class kappa-ell function
$\mathcal{M}$	n dimensional manifold
$\mathcal{O}$	observability matrix
$\mathcal{U}$	open neighborhood of a given initial state x_0
$\mathcal{V}$	neighborhood of a given initial state x_0
$\mathcal{X}$	generic positive invariant set

Other Symbols

$\mathbb{I}^+$	set of nonnegative integer numbers
$\mathbb{C}_f$	augmented terminal set
$\mathbb{C}^\infty$	set of functions differentiable for all degrees of differentiation
δ_{k-j}	Kronecker delta function
$E[\cdot]$	expected value
$E[x\|y]$	conditional mean of x given y
$I(x, \mathcal{M})$	set of all states in $\mathcal{M}$ that are indistinguishable from x
$L[\cdot]$	Lie derivative operator
$\mathcal{N}(x, y)$	normally distributed random variable with mean x and covariance y
$\mathbb{R}$	set of real numbers
$\mathbb{R}^n$	set of real-valued n-vectors
$\mathbb{R}^{n\times m}$	set of real-valued $n \times m$-matrices
$\mathbb{R}^+$	set of positive real numbers
ℓ^{tun}	stage cost function on the optimal tuning problem
$\mathbb{U}$	set of admissible inputs
$\mathbb{V}$	set of admissible output disturbances
$\mathbb{W}$	set of admissible state disturbances
$\mathbb{X}_f$	terminal set
$\mathbb{X}$	set of admissible states
$\mathbb{Y}$	set of admissible outputs

Subscripts

a	augmented
c	controlled variable
f	final sampling step
k	index of sampling step
$k\|j$	expected value of the variable at sampling step k given data up to sampling step j
max	largest element of a given vector
min	smallest element of a given vector
sp	set point value of the variable

Superscripts

a	auxiliary
x	referred to the variable x
o	actual value
s	steady-state or target value
sim	simulated

$\star$	actual process variable
T	transpose operator

Accents

$\bar{}$	expectation operator
$\dot{}$	first-derivative operator
$\hat{}$	estimated variable
$\tilde{}$	deviation variable

1 INTRODUCTION

1.1 MULTILOOP AND MULTIVARIABLE CONTROL

In order to reduce capital needs and energy costs, modern chemical industries are designed with highly integrated processing units and small surge capacities between the various parts of the installation. This makes them difficult to conduct and gives operators scarce opportunities of intervention to prevent upsets from propagating from one unit to another. Moreover, as a consequence of global competition, rapidly changing economic conditions and more stringent environmental and safety regulations, plants are demanded to satisfy higher performance requirements and tighter product quality specifications. Process control allows manufacturers to keep operations running within a safe operating regime, to maximize plant profitability and to ensure high product quality standards. Additionally, since it contemplates automation, it enables a small staff of operating personnel to manage complex process efficiently.

A generic automatic control unit combines several variable measurements provided by sensors with a priori informations about the plant to drive the controlled outputs to their desired values (set-points). In designing controller for multi-input multi-output (MIMO) systems, it is common to distinguish between two approaches: *multiloop control* and *multivariable control*.

Multiloop control can be regarded as a strategy for directly exploiting single-input single-output (SISO) methods in a MIMO setting: manipulated and controlled variables are paired and then one single-loop controller is assigned to link each input-output pair, resulting in a network of multiple interacting control loops. One advantage of this method is the usage of simple algorithms, which leads to a reasonable computational burden for the analog computing equipment. Another benefit is the ease of both understanding and monitoring of the resulting control structure by plant operating personnel. Indeed, each manipulated variable is coupled with a controlled one and a failed loop, once identified, can be easily maintained as its closure does not affect performances of the other loops. An ulterior advantage is that standard control designs have been developed for common unit operations like furnaces, boilers, compressors [...], therefore, several general structures are in common use and can be enhanced in a straightforward fashion in case of a plant upgrade. However, since individual SISO controllers are usually detuned in order to mitigate the effects of loop interactions, multiloop control performs poorly when there is a broad mutual influence among control loops or when process disturbances have a great effect on the controlled variables. Furthermore, since this approach needs at least an equal number of inputs and outputs to be enforced, it cannot be applied to non-square systems with fewer controls than responses which often arise in chemical industries.

In order to address such issues multivariable control can be used. This term refers to a class of control strategies in which an explicit model of the plant and all the available variables measurements are used jointly to determine the value of the inputs. Multivariable control techniques allow to reduce drastically the magnitude of the interactions between system variables because the effect of each maneuvered on each controlled one is captured by the dynamic model of the process making input signal based not just on the status of one assigned output, but on all plant responses. Moreover, since there are no restrictions on process variables dimension, such approach can be even applied to non-square systems. Many multivariable control techniques have been proposed and sometimes implemented, but only Model Predictive Control (MPC) technologies have been widely applied in the industrial processes [6].

1.2 MPC: MOTIVATION, STRUCTURE AND STRATEGY

In general, each process unit is characterized by a series of peculiarities. These lead to specific performance criteria that may be difficult to express in a unified control framework, requiring additional logic to delineate different operating modes. Moreover, it is well known that the economic operating point of a typical process unit often lies at the intersection of constraints [52]. An efficient control system must therefore be able to handle any desired performance criterion and keep the process as close as possible to constraints without violating them.

The MPC algorithms success is largely due to their ability to take into account constraints on both process inputs and outputs and to satisfy some optimal performance criteria by solving on-line optimization problems. The incorporation of an explicit process model allows controllers to deal straightforwardly with all significant features of the process dynamics while the direct inclusion of constraints into the control algorithm makes constraint violations far less-likely, resulting in tighter control at the optimal constrained steady-state for the process [53].

A schematic of an MPC unit is shown in Figure 1.

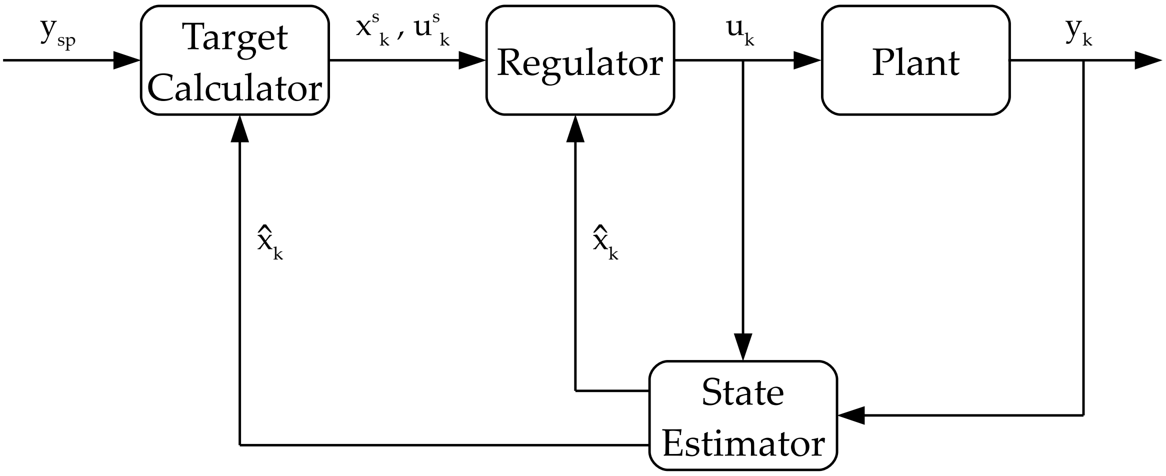


Figure 1: Schematic of an MPC unit.

Typically, an MPC unit is regarded to be composed by three modules: a state estimator, a target calculator and a regulator.

The regulator predicts the dynamic behavior of the system over a *prediction horizon* and determines a future command sequence over a shorter time interval, known as *control horizon*, that brings the process from its current state, x_k, to the current state target x^s_k. The input profile calculation is based on optimizing a performance criterion, generally given as a function of the predicted states and inputs deviations from their target values x^s_k, u^s_k, subject to system dynamics and constraints on both states and controls. Such problem is often called *optimal control problem* (OCP).

If the OCP could be solved for infinite horizons, and if there were no disturbances and no plant-model mismatch, then one could apply the input sequence found at a certain time t_k to the system for all time $t \geqslant t_k$. However, this is possible in a very limited number of special cases. Thus, in order to incorporate some feedback logic a *receding horizon* philosophy is adopted: at time t_k only the first input u_k of the optimal command sequence is actually applied to the plant. The remaining control moves are discarded, and a new OCP is solved at time t_{k+1} when a new value of the current state becomes available.

As stated above the implementation of the receding horizon regulator requires the knowledge of x_k to compute the solution of the OCP. Unfortunately, in most applications, for reasons of convenience or cost-effectiveness the variables that can be measured are a small subset of the state variables. There are also some cases in which the state variables have no physical meaning and hence cannot be measured at all. In addition, measurements and state evolution are corrupted respectively with sensor and process noise. The state estimator module uses all the measured data to produce the optimal estimate of the actual state, $\hat{x}_k$, to be used in both regulator and target calculator. Depending on the cases, the value of $\hat{x}_k$ can arise from the solution of a properly posed constrained optimization problem or from the implementation of recursive algorithms. As feedback in the regulator comes from the update of $\hat{x}_k$, its performances will be highly influenced by the quality of this estimate.

Finally, the target calculator determines new state and input targets defining the closest stationary to the set-point, y_{sp}; this allows the controller to respond to changes in y_{sp}, which may result from various plant-wide necessities. x_k^s and u_k^s are calculated from an optimization embroiling constraints on states and inputs and a steady-state model of the process. Since it involves only the steady-state value of variables and not trajectories of states and inputs through the prediction horizon, the target calculation problem is much smaller than the estimator and regulator ones.

1.3 ISSUES IN NONLINEAR OFFSET-FREE MPC AND THESIS OBJECTIVES

Needing more solutions of optimization problems at each sampling instant, the MPC implementation requires a solution algorithm able to converge reliably to the optimum in no more than a few tens of seconds to be useful in industrial applications. By using linear models (LMPC) the optimization problems take the form of convex linear (LPs) or quadratic programs (QPs), for which a great variety of fast and reliable numerical methods can easily be found in literature. Conversely, with nonlinear models (NMPC) they result in nonlinear programs (NLPs) which may have multiple local optima and will demand a much larger number of computations at each sample, even without providing any hard guarantees that the solution found is a global optimum. Moreover, linear empirical models can be identified in a more straightforward manner from process data than nonlinear models and prove themselves to be suitable for a great variety of operations, especially if high quality feedback measurements are available.

For these reasons the vast majority of MPC applications are based on linear models.

Despite this, there are cases in which the use of NMPC technology is justified. In industrial control system operations, for example, it is often required that the selected controlled variables approach their set-point without offset in presence of unmodeled, nonzero mean disturbances and plant-model mismatch; this problem is known as *offset-free control*. The general approach to the offset-free control problem is to augment the dynamical model of the system with integrating disturbance terms and employ the state estimator to estimate state and disturbance.

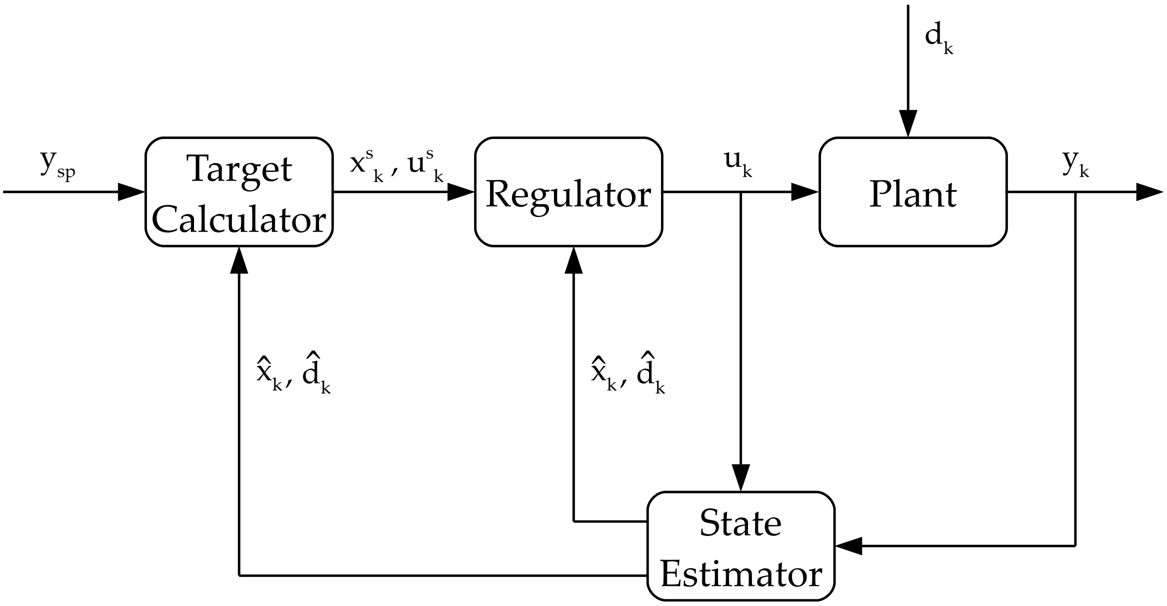


Figure 2: Schematic of an MPC unit for offset-free control achievement.

General conditions and design criteria to obtain steady-state offset-free performances using augmented linear models in both state estimator and regulator implementations are well established in common practice. Nevertheless, since process dynamics are in general inherently nonlinear, even if zero steady-state offset is ensured for nonlinear plants, transient behaviors may be unsatisfactory due to modelling errors, particularly if the nonlinearities are significant. This motivates the usage of augmented nonlinear models and hence of NMPC technology.

However, the obstacles associated with estimating state variables and disturbance terms when implementing an NMPC offset-free framework are nontrivial. First of all, the a priori observability of a dynamic system, that is the possibility to infer the states of the process by knowledge of its outputs, is not a well explored concept as its linear counterpart and can be difficult to check.

Secondly, due to the mathematical complexities introduced by a nonlinear model, nonlinear state estimators are difficult to design, implement and tune. Some of them are developed for certain classes of nonlinear systems only, while other are based on some simplifications of or approximations to the nonlinear model and, consequently, have few limitations.

Finally, while in the linear case the equivalence of different disturbance model choices has been extensively discussed in literature (Pannocchia [42] and Rajamani et al. [54]) there have been no comparable breakthroughs in the nonlinear context. For nonlinear systems the effects of adding integrating disturbances are specific to the application, therefore, the disturbance representation must be properly chosen and the estimator optimally tuned in order to obtain an high performing control system.

1.4 THESIS OVERVIEW

In this work the problem of designing a nonlinear disturbance model for offset-free NMPC frameworks is addressed. The remainder of this thesis is organized as follows.

CHAPTER 2 contains the theoretical basis of this work.

Starting from the formulation of the state estimation problem, conditions under which it particularized to a filtering problem are outlined. Then, the Kalman filter algorithm is derived as the optimal solution of the filtering problem for linear systems and the extended Kalman filter presented as its most straightforward extension to nonlinear systems. Subsequently, a comprehensive survey of the theory of nonlinear observability is provided with focus on its implications in the estimator performances. Furthermore, methods to check observability of nonlinear systems are reviewed in detail together with various measures of the degree of observability.

Finally, the offset-free control problem is addressed, and general design criteria on how to build an offset-free NMPC framework are exposed.

CHAPTER 3 starts with a description of the modules constituting the offset-free NMPC unit adopted, the general augmented model employed and the design choices made. Then it deepens in the characterization of the proposed design strategy, showing all facilities provided by this method in determining the optimal nonlinear disturbance model to be implemented. At last it focus on the peculiar theoretical and computational aspects of the proposed approach outlining how the second phase of the procedure can even be used in other contexts to improve the state estimator performances.

CHAPTER 4 concerns the application of the proposed method for the design of an offset-free NMPC framework for the control of a non-isothermal CSTR reactor. The technique effectiveness is proved with several examples which show how the NMPC algorithm synthesized gives better performances if compared with the others commonly adopted in the relevant literature.

CHAPTER 5 contains a brief review of the main results and some conclusions specifying where and how the proposed approach can be useful in future research and which possible improvements can be easily implemented.

2 PRELIMINARIES AND LITERATURE REVIEW

In this chapter the theoretical basis of this work are presented together with references to the relevant literature. The detailed outline is as follows. Section 2.1 contains a brief overview of different model classes for nonlinear systems modeling with a focus on the ODE models adopted in this thesis. The basic concepts of state estimation are discussed later in section 2.2: after recalling conditions under which a state estimation problem particularized to a filtering one, it is shown that the Kalman filter is the optimal state estimator for linear systems and how the extended Kalman filter is its natural nonlinear counterpart. The final part of the section is devoted to the tuning of such filters. As observability of nonlinear systems plays an important role in nonlinear state estimators performances, it is addressed from a differential-geometric viewpoint in section 2.3. Finally, all the concepts introduced in the previous sections are recalled in section 2.4 in order to give general design criteria on how to build an offset-free NMPC framework.

2.1 DYNAMICS MODELS

In this thesis an ordinary differential equations (ODEs) model in state-space form is employed to represent the dynamic behavior of the real plant.
Although there exist a variety of models suitable for this purpose like neural networks, fuzzy models and polynomial models, first-principle models are the most popular tools in literature for dynamic process modeling [68]. These representations, which include not only ordinary differential equations models but also partial differential equations (PDEs), and differential algebraic equations (DAEs) models, have two major benefits with respect to the other ones. The first one is that they are based on theoretical foundations, therefore, exhibit the same type of behavior as the plant in a wider range of operating conditions. The second one is linked to the fact that fewer model parameters need to be fit to the real process resulting in easier and faster identification procedures. However, ODEs models may result computationally expensive especially for high dimensional systems as they must be numerically integrated to predict the plant behavior. An ulterior disadvantage is that, depending on the complexity of the physical system to be modeled, model equations cannot be derived or require expertise in the area of knowledge at the advanced level for their development.

2.1.1 Continuous-time models

The following time-invariant nonlinear state-space representation will be considered:

$$\begin{aligned}\frac{d}{dt}x(t) &= f(x(t), u(t), w(t)) \\ y(t) &= h(x(t), v(t))\end{aligned} \tag{2.1}$$

where $x(t) \in \mathbb{R}^n$ denotes the state, $u(t) \in \mathbb{R}^m$ the input, $y(t) \in \mathbb{R}^p$ the output and

$w(t) \in \mathbb{R}^{n_w}$ and $v(t) \in \mathbb{R}^{n_v}$ respectively the process and the sensor noise.
The following shorthand notation for (2.1) is introduced:

$$\begin{aligned} \dot{x} &= f(x, u, w) \\ y &= h(x, v) \end{aligned} \tag{2.2}$$

where $\dot{x}$ denotes the time derivative of $x(t)$.

2.1.2 Discrete-time models

Often real sensors provide measurement at the times $t_k = k\Delta t$, where $k \in \mathbb{I}^+$ represents the number of the sampling step and Δt denotes a fixed duration between two sampling step, termed *sampling time*. These situations require a discrete-time representation of the process dynamics which can be performed by discretizing the continuous-time nonlinear model (2.2) with zero-order hold on the input, $u(\cdot)$, and the process noise, $w(\cdot)$, to give:

$$\begin{aligned} x_{k+1} &= f\left(x_k, u_k, w_k\right) \\ y_k &= h\left(x_k, v_k\right) \end{aligned} \tag{2.3}$$

Note that in order to avoid an unnecessary and heavy notation, the functions $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{n_w} \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^{n_v}$ have been denoted in the same manner in both the continuous (2.2) and discrete (2.3) time models even though they are different.

2.2 STATE ESTIMATION

State estimation is the problem of inferring the values of the state variables at any given time t_k based on the knowledge of the system dynamics and the availability of inputs and noisy outputs measurements.
Consider the discrete-time system (2.3) and let $\hat{x}_{k|j}$ represent the state estimate at sample instant k given data up to sample instant j. Depending on the amount of information that is available to be employed in the state estimate it is common to distinguish between three classes of estimation problems: *smoothing*, *filtering* and *prediction*.
The smoothing problem consists in estimating x_k from data collected both before and after time t_k. Thus, $\hat{x}_{k|j}$ is a smoothed estimate when $k < j$. Since smoothing does not provide current values of the state variable but only at some time in the past, it cannot be used in real-time control applications but in process analysis and diagnosis only.
Conversely, if the aim is to find the optimal prediction of x_k more than one time step ahead of the available measurements, a predicted estimate, $\hat{x}_{k|j}$ with $k > j$, must be formed. Predictions are obtained by using the process model to extrapolate the filtered estimates into the future and are typically used to compute the objective function in MPC regulator unit formulations or when there are lab analysis delays in measurements of outputs (e.g. in chromatographic analysis).
Finally, $\hat{x}_{k|j}$ is a filtered estimate when $k = j$, i.e. when the state estimate at time t_k is made from data up to time t_k, but not beyond. According to Ray [56], the filtering technique is the most common employed in control applications because the most up-to-date state estimates are provided in a sequential fashion. For that reason, only the filtering problem will be threated in this work.

2.2.1 Filtering problem

At any given time t_k let an initial estimate $\hat{x}_0$ of x_0, a sequence of noisy measurements $Y_k = \{y_1, y_2, \ldots, y_k\}$, a sequence of inputs $U_{k-1} = \{u_0, u_1, \ldots, u_{k-1}\}$, and the representation of the process dynamics (2.3) be available and consider the conditional probability density function of x_k given such information, $p\left(x_k | Y_k, U_{k-1}\right)$.

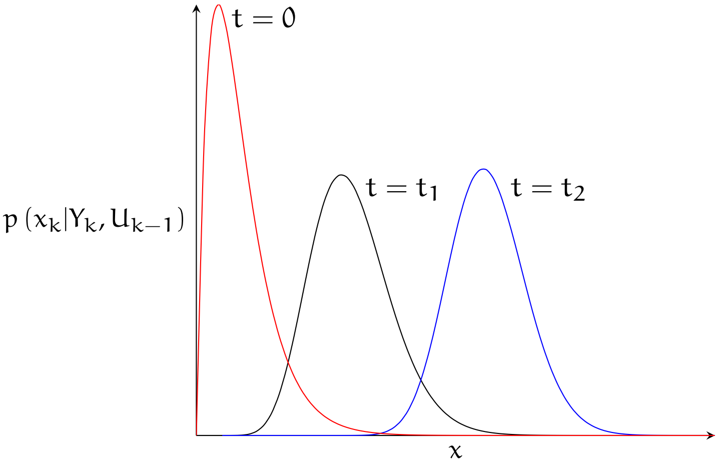


Figure 3: Evolution of probability distributions of system state conditional on the data.

$p\left(x_k | Y_k, U_{k-1}\right)$, plotted in Figure 3, is the main statistical function of interest, since it conveys all the information about x_k up to and including time t_k.
It can be noted that $p\left(x_k | Y_k, U_{k-1}\right)$ changes with time as new measurements are collected: the smaller sensor and process noise covariances are, the narrower the dispersion is. Contractions in the distribution lead to an improvement in the estimate quality.
Aiming at obtaining the optimal value of $\hat{x}_k$, a filter propagates the conditional probability density function with time and at any given time t_k produces the value of $\hat{x}_{k|k}$ that optimizes a properly chosen objective function; depending on such criterion, different state estimates may arise. The *minimum mean-square error (MMSE)* and the *maximum a posteriori (MAP)* criteria are the most commonly used in filtering literature.
Defined the *state estimation error* as:

$$\epsilon_k = x_k - \hat{x}_k \tag{2.4}$$

the MMSE estimation consists in finding the state estimate, $\hat{x}_k$, that minimizes the conditional mean-square value of ϵ_k, that is:

$$\min_{\hat{x}_k} E\left[\epsilon_k^T \epsilon_k | Y_k, U_{k-1}\right] \tag{2.5}$$

From Equation 2.4 the cost function can be rewritten as:

$$\begin{aligned}
E\left[\epsilon_k^T \epsilon_k | Y_k, U_{k-1}\right] &= E\left[\left(x_k - \hat{x}_k\right)^T \left(x_k - \hat{x}_k\right) | Y_k, U_{k-1}\right] \\
&= E\left[x_k^T x_k - x_k^T \hat{x}_k - \hat{x}_k^T x_k + \hat{x}_k^T \hat{x}_k | Y_k, U_{k-1}\right] \\
&= E\left[x_k^T x_k | Y_k, U_{k-1}\right] - E\left[x_k^T | Y_k, U_{k-1}\right] E\left[\hat{x}_k | Y_k, U_{k-1}\right] - \\
&\quad - E\left[\hat{x}_k^T | Y_k, U_{k-1}\right] E\left[x_k | Y_k, U_{k-1}\right] + E\left[\hat{x}_k^T \hat{x}_k | Y_k, U_{k-1}\right] \\
&= E\left[x_k^T x_k | Y_k, U_{k-1}\right] - E\left[x_k^T | Y_k, U_{k-1}\right] \hat{x}_k - \\
&\quad - \hat{x}_k^T E\left[x_k | Y_k, U_{k-1}\right] + E\left[\hat{x}_k^T \hat{x}_k | Y_k, U_{k-1}\right]
\end{aligned} \tag{2.6}$$

Adding and subtracting $\mathrm{E}\left[x_k^T|Y_k, U_{k-1}\right]\mathrm{E}\left[x_k|Y_k, U_{k-1}\right]$ to Equation 2.6 and substituting the resulting expression in (2.5) yields:

$$\min_{\hat{x}_k} \mathrm{E}\left[x_k^T x_k|Y_k, U_{k-1}\right] - \mathrm{E}\left[x_k^T|Y_k, U_{k-1}\right]\mathrm{E}\left[x_k|Y_k, U_{k-1}\right] + \\ + \left\{\hat{x}_k - \mathrm{E}\left[x_k|Y_k, U_{k-1}\right]\right\}^T \left\{\hat{x}_k - \mathrm{E}\left[x_k|Y_k, U_{k-1}\right]\right\} \tag{2.7}$$

Since the first two terms in (2.7) do not depend on $\hat{x}_k$, the objective function achieves a global minimum when the quadratic term is equal to zero, that is:

$$\hat{x}_k = \mathrm{E}\left[x_k|Y_k, U_{k-1}\right] \tag{2.8}$$

Therefore, the adoption of the MMSE criterion provides a value of $\hat{x}_k$ which is equal to the mean of the distribution $p\left(x_k|Y_k, U_{k-1}\right)$.
The MAP estimation, conversely, can be formalized in the following manner:

$$\max_{\hat{x}_k} p\left(x_k|Y_k, U_{k-1}\right) \tag{2.9}$$

Thus, the MAP criterion leads to a value of $\hat{x}_k$ which is equal to the maximum (i.e. the mode) of the dispersion $p\left(x_k|Y_k, U_{k-1}\right)$.
Unfortunately, the conditional probability density function may have a complex expression and many local maxima, therefore, finding an optimal state estimate in the MAP sense can be a major computational challenge. Analogous considerations can be made relatively to the MMSE criterion. Under a set of particular conditions related with the linearity of the system dynamics and the normality of the random variables involved, things get better. In particular, the conditional probability density function propagated by the filter at any given time t_k is a Gaussian for which a solution to the problems (2.9) and (2.5) can be easily computed. Since the mode coincides with the mean and the median for a normal dispersion, the MMSE and the MAP criteria lead to the same value of $\hat{x}_k$ in such case. Moreover, as a Gaussian is completely characterize by its mean and covariance matrix, the filter is required to propagate only the first and the second moment of $p\left(x_k|Y_k, U_{k-1}\right)$ and not the entire distribution, with a strong reduction of the computational burden.
Under the assumptions made above the general optimal filter simplifies to what is known with the name of *Kalman filter (KF)*.

2.2.2 Kalman filter

Suppose that the actual plant has a linear (possibly time-varying) dynamics and consider the following particularization of (2.3):

$$x_{k+1} = A_k x_k + B_k u_k + w_k \tag{2.10a}$$

$$y_k = C_k x_k + v_k \tag{2.10b}$$

where $A_k \in \mathbb{R}^{n \times n}$, $B_k \in \mathbb{R}^{n \times m}$ and $C_k \in \mathbb{R}^{p \times n}$ represent the process matrices.
Assume that w_k and v_k are *white noises*, i.e. discrete signal whose samples can be regarded as a sequence of serially uncorrelated random variables with zero mean and finite variance (see subsection A.1.5 for further insights).

In addition, assume that they are normally distributed with known covariance matrices $Q_k \in \mathbb{R}^{n \times n}$ and $R_k \in \mathbb{R}^{p \times p}$, respectively:

$$\begin{aligned} w_k &\sim \mathcal{N}(0, Q_k) \\ v_k &\sim \mathcal{N}(0, R_k) \\ E\left[w_k w_j^T\right] &= Q_k \delta_{k-j} \\ E\left[v_k v_j^T\right] &= R_k \delta_{k-j} \\ E\left[v_k w_j^T\right] &= 0 \end{aligned} \tag{2.11}$$

where δ_{k-j} is the Kronecker delta function defined as:

$$\delta_{k-j} = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k \neq j \end{cases} \tag{2.12}$$

The choice of a zero mean for w_k and v_k is made for simplicity of exposition. However, in the general case of $w_k \sim \mathcal{N}(\bar{w}_k, R_k)$ and $v_k \sim \mathcal{N}(\bar{v}_k, Q_k)$ is sufficient to consider two constant additive terms equals to $\bar{w}_k$ and $\bar{v}_k$ in the state transition and output equations respectively to move back to the said case.

Initially, since there are no data available to estimate x_0, the following assumption is made:

$$x_0 \sim \mathcal{N}\left(\bar{x}_0, P_{0|0}\right) \tag{2.13}$$

and $\hat{x}_{0|0}$ is set equal to the expected value of the initial state, that is:

$$\hat{x}_{0|0} = \bar{x}_0 \tag{2.14}$$

In general $P_{0|0}$ represents the uncertainty in the initial estimate of x_0: the larger $P_{0|0}$ is and the less is known about x_0. In situations where there is absolutely no idea about the value of the initial state a common choice, referred to in statistic literature as *noninformative prior*, is to set $\bar{x}_0 = 0$ and choose a large value of $P_{0|0}$.

Since x_0 is assumed to be normally distributed, w_k and v_k are white and Gaussian and the state and observation dynamics are linear, the conditional probability density functions $p(x_k|Y_k, U_{k-1})$ are Gaussian for any k. Under the assumptions made on the mean of w_k and v_k the expected value evolution with time of such distributions can be derived by taking the expected value of both side of Equation 2.10a:

$$\begin{aligned} \bar{x}_{k+1} &= E\left[x_{k+1}\right] \\ &= A_k \bar{x}_k + B_k u_k \end{aligned} \tag{2.15}$$

Combining Equation 2.15 and Equation 2.10a yields:

$$\begin{aligned} (x_{k+1} - \bar{x}_{k+1})(\cdots)^T &= (A_k x_k + B_k u_k + w_k - \bar{x}_k)(\cdots)^T \\ &= [A_k (x_k - \bar{x}_k) + w_k] + [\cdots]^T \\ &= A_k (x_k - \bar{x}_k)(x_k - \bar{x}_k)^T A_k^T + w_k w_k^T + \\ &\qquad A_k (x_k - \bar{x}_k) w_k^T + w_k (x_k - \bar{x}_k)^T A_k^T \end{aligned} \tag{2.16}$$

thus, the distribution covariance time dependency can be obtained by explicating the mean of the above expression:

$$\begin{aligned}
P_{k+1} &= E\left[\left(x_{k+1}-\bar{x}_{k+1}\right)\left(x_{k+1}-\bar{x}_{k+1}\right)^T\right] \\
&= E\Big[A_k\left(x_k-\bar{x}_k\right)\left(x_k-\bar{x}_k\right)^T A_k^T + w_k w_k^T + \\
&\qquad A_k\left(x_k-\bar{x}_k\right) w_k^T + w_k\left(x_k-\bar{x}_k\right)^T A_k^T\Big] \\
&= A_k E\left[\left(x_k-\bar{x}_k\right)\left(x_k-\bar{x}_k\right)^T\right] A_k^T + E\left[w_k w_k^T\right] + \\
&\qquad \cancel{A_k E\left[\left(x_k-\bar{x}_k\right) w_k^T\right]} + \cancel{E\left[\left(x_k-\bar{x}_k\right) w_k^T\right] A_k^T} \\
&= A_k P_k A_k^T + Q_k \qquad (2.17)
\end{aligned}$$

Since the first measurement is taken at sample $k=1$, there are no additional information available to update $\hat{x}_{0|0}$ between sample $k=0$ and $k=1$. Therefore, the best that can be done right now is an update of the state estimate and of its related covariance based on the knowledge of their dynamics:

$$\hat{x}_{1|0} = A_0\hat{x}_{0|0} + B_0 u_0 \qquad (2.18)$$

$$P_{1|0} = A_0 P_{0|0} A_0^T + Q_0 \qquad (2.19)$$

By extending the reasoning to each generic sample time t_k the following more general equations are obtained:

$$\hat{x}_{k+1|k} = A_k\hat{x}_{k|k} + B_k u_k \qquad (2.20)$$

$$P_{k+1|k} = A_k P_{k|k} A_k^T + Q_k \qquad (2.21)$$

Equation 2.20 and Equation 2.21 constitute the so called *time update step* of the filter. During the time update step $p\left(x_k|Y_k, U_{k-1}\right)$ is propagated through a linear transformation to give $p\left(x_{k+1}|Y_k, U_k\right)$, which conveys all that can be said about x_{k+1} before making the observation y_{k+1}. When the measure is collected there is a need of a practically realizable and computationally efficient method for the recursive improvement of the state estimate. To accomplish this end Kalman [26] proposed an update of the conditional probability density function mean based on a linear combination of the state prediction and the most up-to-date observed measure:

$$\hat{x}_{k+1|k+1} = K^*_{k+1}\hat{x}_{k+1|k} + K_{k+1} y_{k+1} \qquad (2.22)$$

where $K^*_{k+1} \in \mathbb{R}^{n\times n}$ and $K_{k+1} \in \mathbb{R}^{n\times p}$ are weighting or *gain* matrices that must be properly chosen in order to form the optimal filtered estimate. A first desirable property of the estimator (2.22) is that, on average, the state estimate is equal to the actual value of the state. Such requirement is known as *unbiasedness condition* and can be formalized as follows:

$$E\left[\hat{x}_{k+1|k+1}\right] = E\left[x_{k+1}\right] \qquad (2.23)$$

Substituting Equation 2.10b into Equation 2.22 and taking expectations provides:

$$\begin{aligned}
E\left[\hat{x}_{k+1|k+1}\right] &= E\left[K^*_{k+1}\hat{x}_{k+1|k} + K_{k+1}C_{k+1}x_{k+1} + K_{k+1}v_{k+1}\right] \\
&= K^*_{k+1}E\left[\hat{x}_{k+1|k}\right] + K_{k+1}C_{k+1}E\left[x_{k+1}\right] + \cancel{K_{k+1}E\left[v_{k+1}\right]} \\
&= K^*_{k+1}E\left[\hat{x}_{k+1|k}\right] + K_{k+1}C_{k+1}E\left[x_{k+1}\right] \qquad (2.24)
\end{aligned}$$

Assuming that $\hat{x}_{k|k}$ is an unbiased estimate yields:

$$\begin{aligned} E\left[\hat{x}_{k+1|k}\right] &= E\left[A_k \hat{x}_{k|k} + B_k u_k\right] \\ &= A_k E\left[\hat{x}_{k|k}\right] + B_k u_k \\ &= E\left[x_{k+1}\right] \end{aligned} \tag{2.25}$$

Hence, by combining Equation 2.24 and Equation 2.25 it follows:

$$\begin{aligned} E\left[\hat{x}_{k+1|k+1}\right] &= K^*_{k+1} E\left[x_{k+1}\right] + K_{k+1} C_{k+1} E\left[x_{k+1}\right] \\ &= \left(K^*_{k+1} + K_{k+1} C_{k+1}\right) E\left[x_{k+1}\right] \end{aligned} \tag{2.26}$$

and the condition (2.23) reduces the requirement to:

$$K^*_{k+1} = I - K_{k+1} C_{k+1} \tag{2.27}$$

Substituting Equation 2.27 into Equation 2.22 and rearranging yield:

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}\left(y_{k+1} - C_{k+1} \hat{x}_{k+1|k}\right) \tag{2.28}$$

where K_{k+1} is known as the *Kalman gain* and the expression inside parentheses as *innovation term*.
Since the estimator (2.28) is unbiased regardless of the value of K_{k+1} used, the Kalman gain can be determined in order to ensure that $\hat{x}_{k+1|k+1}$ is the optimal estimate in accordance with some optimality criterion. By choosing the MMSE criterion follows:

$$\begin{aligned} \ell &= \min_{K_{k+1}} E\left[\epsilon^T_{k+1} \epsilon_{k+1}\right] \\ &= \min_{K_{k+1}} E\left[\text{trace}\left(\epsilon_{k+1} \epsilon^T_{k+1}\right)\right] \\ &= \min_{K_{k+1}} \text{trace}\left(P_{k+1|k+1}\right) \end{aligned} \tag{2.29}$$

where $P_{k+1|k+1}$ can be derived as:

$$\begin{aligned} P_{k+1|k+1} &= E\left[\epsilon_{k+1} \epsilon^T_{k+1}\right] \\ &= E\left\{\left[\left(I - K_{k+1} C_{k+1}\right) \epsilon_k - K_{k+1} \nu_{k+1}\right] [\cdots]^T\right\} \\ &= \left(I - K_{k+1} C_{k+1}\right) E\left[\epsilon_k \epsilon^T_k\right] \left(I - K_{k+1} C_{k+1}\right)^T - \\ &\qquad - K_{k+1} \cancel{E\left[\nu_{k+1} \epsilon^T_k\right] \left(I - K_{k+1} C_{k+1}\right)^T} - \\ &\quad - \cancel{\left(I - K_{k+1} C_{k+1}\right) E\left[\epsilon_k \nu^T_{k+1}\right]} K^T_{k+1} + K_{k+1} E\left[\nu_{k+1} \nu^T_{k+1}\right] K^T_{k+1} \\ &= \left(I - K_{k+1} C_{k+1}\right) P_{k+1|k} \left(I - K_{k+1} C_{k+1}\right)^T + K_{k+1} R_{k+1} K^T_{k+1} \end{aligned} \tag{2.30}$$

Equation 2.28 and Equation 2.30 constitute what is known with the name of *measurements update step*. During the measurements update step $p\left(x_{k+1}|Y_k, U_k\right)$ is propagated through a linear transformation to give $p\left(x_{k+1}|Y_{k+1}, U_k\right)$ and the filtering recursion is complete.
Keeping in mind that for any matrix A and a symmetric matrix B:

$$\frac{\partial \, \text{trace}\left(ABA^T\right)}{\partial A} = 2AB \tag{2.31}$$

combining Equation 2.29 and Equation 2.30 and differentiating with respect to the gain matrix yield:

$$\frac{\partial \ell}{\partial K_{k+1}} = 2\left(I - K_{k+1} C_{k+1}\right) P_{k+1|k} \left(-C^T_{k+1}\right) + 2K_{k+1} R_{k+1} \tag{2.32}$$

Finally, setting the above derivative equal to zero and solving provide the value of K_{k+1} that minimize ℓ:

$$\begin{aligned} K_{k+1}R_{k+1} &= \left(I - K_{k+1}C_{k+1}\right)P_{k+1|k}C_{k+1}^{T} \\ K_{k+1}\left(R_{k+1} + C_{k+1}P_{k+1|k}C_{k+1}^{T}\right) &= P_{k+1|k}C_{k+1}^{T} \\ K_{k+1} &= P_{k+1|k}C_{k+1}^{T}\left(C_{k+1}P_{k+1|k}C_{k+1}^{T} + R_{k+1}\right)^{-1} \qquad (2.33) \end{aligned}$$

Algorithm 1 reports a resume of the key equations which underly the KF algorithm.

Algorithm 1 Kalman Filter

```
1: At $k = 0$, initializations:
       $\hat{x}_{0|0} = E\left[x_0\right]$
       $P_{0|0} = E\left[\left(x_0 - \hat{x}_{0|0}\right)\left(x_0 - \hat{x}_{0|0}\right)^T\right]$
2: Time update:
       $\hat{x}_{k+1|k} = A_k\hat{x}_{k|k} + B_k u_k$
       $P_{k+1|k} = A_k P_{k|k} A_k^T + Q_k$
3: Kalman gain:
       $K_{k+1} = P_{k+1|k}C_{k+1}^T\left(C_{k+1}P_{k+1|k}C_{k+1}^T + R_{k+1}\right)^{-1}$
4: Measurement update:
       $\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}\left(y_{k+1} - C_{k+1}\hat{x}_{k+1|k}\right)$
       $P_{k+1|k+1} = \left(I - K_{k+1}C_{k+1}\right)P_{k+1|k}$
5: Replace $k+1$ with $k$ and go to Step 2
```

Note that even when x_0, w_k and v_k are not assumed to be normally distributed the Kalman filter still provides the optimal state estimate in the MMSE sense which can be achieved with a linear filter. However, there may be a nonlinear filter that gives a better solution to the filtering problem.

2.2.3 Extended Kalman filter

Originally proposed by Stanley Schmidt for spacecraft navigation problems the extended Kalman filter (EKF) is undoubtedly the most widely used nonlinear state estimation technique that has been applied in the past few decades [62]. This filter relaxes the linearity assumption and makes use of a nonlinear discrete-time process model like (2.3):

$$x_{k+1} = f(x_k, u_k, w_k) \qquad (2.34a)$$

$$y_k = h(x_k, v_k) \qquad (2.34b)$$

where acceptances (2.11) continue to be respected.
Since there are no data available at time t_0, x_0 is assumed to be normally distributed with mean $\bar{x}_0$ and covariance $P_{0|0}$. Even in this case $\hat{x}_{0|0}$ is set equal to $\bar{x}_0$.
When a normally distributed random variable is passed through a nonlinear dynamics the resulting conditional probability density functions propagated by the filter at any time t_k are no longer gaussian in shape and hence all the related benefits provided by the normal

distributions are lost. In order to avoid this, the EKF calculates a Gaussian approximation of the real dispersion by carrying out a linearization of the nonlinear transformation around the mean of the dispersion.
A graphical representation of the previous derivations is reported in Figure 4.

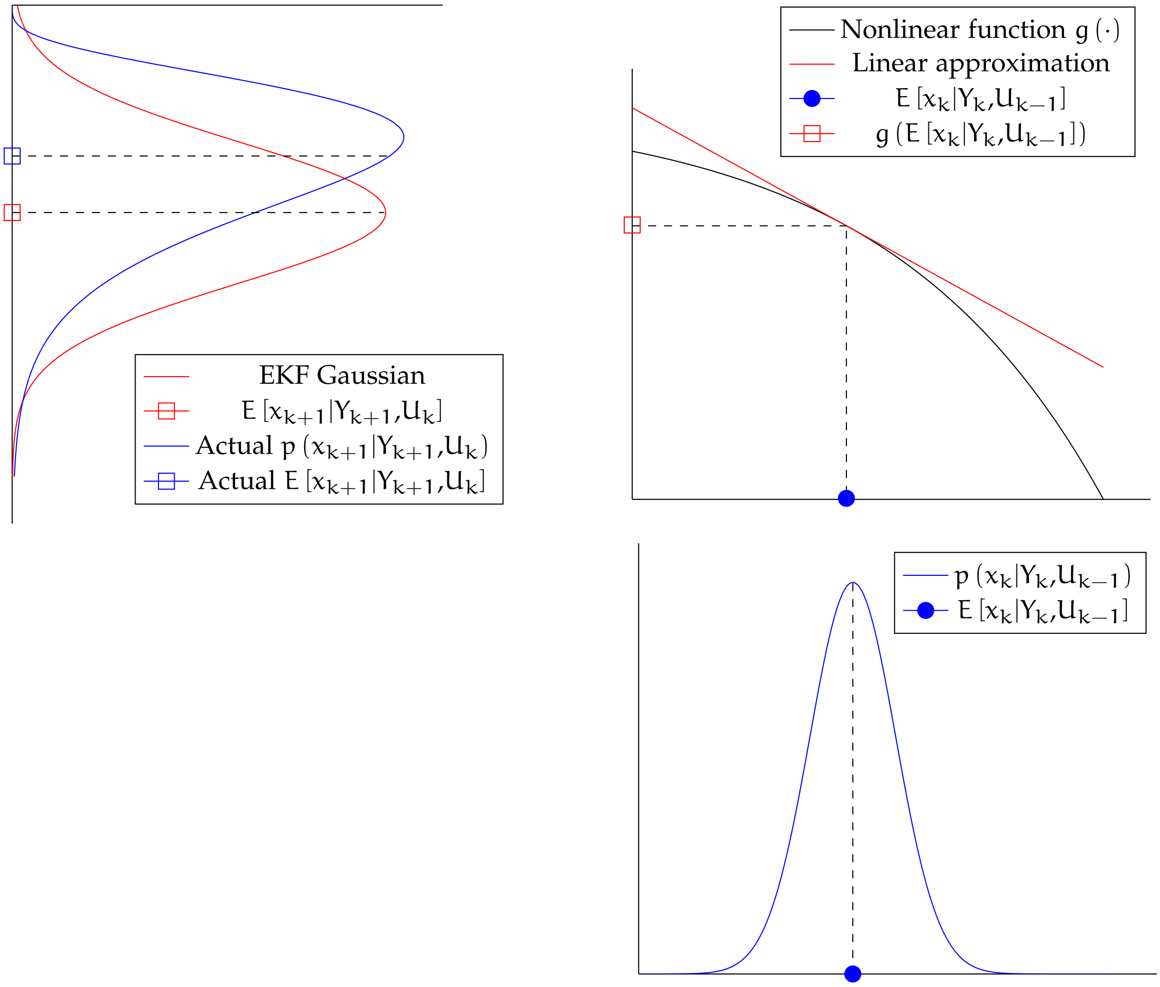


Figure 4: Effect of nonlinear transformation of Gaussian random variable and EKF approximation.

Thus, performing a fist-order Taylor series expansions of Equation 2.34a around the mean of $p\left(x_k|Y_k,U_{k-1}\right)$ and of Equation 2.34b around the mean of $p\left(x_{k+1}|Y_k,U_k\right)$ provides:

$$x_{k+1} \approx f\left(\hat{x}_{k|k},u_k,0\right) + \left.\frac{df}{dx}\right|_{\substack{x=\hat{x}_{k|k}\\u=u_k\\w=0}} \left(x_k - \hat{x}_{k|k}\right) + \left.\frac{df}{dw}\right|_{\substack{x=\hat{x}_{k|k}\\u=u_k\\w=0}} w_k \tag{2.35a}$$

$$y_k \approx h\left(\hat{x}_{k|k-1},0\right) + \left.\frac{dh}{dx}\right|_{\substack{x=\hat{x}_{k|k-1}\\v=0}} \left(x_k - \hat{x}_{k|k-1}\right) + \left.\frac{dh}{dv}\right|_{\substack{x=\hat{x}_{k|k-1}\\v=0}} v_k \tag{2.35b}$$

By taking the expected value of both side of (2.34) the time dependencies of mean and observation result:

$$\hat{x}_{k+1|k} = f\left(\hat{x}_{k|k},u_k,0\right) \tag{2.36a}$$

$$\hat{y}_{k+1|k} = h\left(\hat{x}_{k+1|k},0\right) \tag{2.36b}$$

and substituting them into Equation 2.35 yields:

$$
\begin{aligned}
x_{k+1} &\approx \hat{x}_{k+1|k} + \left.\frac{df}{dx}\right|_{\substack{x=\hat{x}_{k|k} \\ u=u_k \\ w=0}} \left(x_k - \hat{x}_{k|k}\right) + \left.\frac{df}{dw}\right|_{\substack{x=\hat{x}_{k|k} \\ u=u_k \\ w=0}} w_k \\
y_k &\approx \hat{y}_{k+1|k} + \left.\frac{dh}{dx}\right|_{\substack{x=\hat{x}_{k|k-1} \\ v=0}} \left(x_k - \hat{x}_{k|k-1}\right) + \left.\frac{dh}{dv}\right|_{\substack{x=\hat{x}_{k|k-1} \\ v=0}} v_k
\end{aligned}
\tag{2.37}
$$

Then, by defining the system Jacobian matrices as:

$$
\begin{aligned}
A_k &= \left.\frac{df}{dx}\right|_{\substack{x=\hat{x}_{k|k} \\ u=u_k \\ w=0}} \\
C_k &= \left.\frac{dh}{dx}\right|_{\substack{x=\hat{x}_{k|k-1} \\ v=0}} \\
G_k &= \left.\frac{df}{dw}\right|_{\substack{x=\hat{x}_{k|k} \\ u=u_k \\ w=0}} \\
F_k &= \left.\frac{dh}{dv}\right|_{\substack{x=\hat{x}_{k|k-1} \\ v=0}}
\end{aligned}
\tag{2.38}
$$

and the signals:

$$
\begin{aligned}
\tilde{u}_k^{\bullet} &= \hat{x}_{k+1|k} - A_k \hat{x}_{k|k} \\
\tilde{u}_k^{in} &= \hat{y}_{k|k-1} - C_k \hat{x}_{k|k-1} \\
\tilde{w}_k &\sim \left(0, G_k Q_k G_k^T\right) \\
\tilde{v}_k &\sim \left(0, F_k R_k F_k^T\right)
\end{aligned}
\tag{2.39}
$$

the following linear state-space model is obtained:

$$x_{k+1} = A_k x_k + \tilde{u}_k^{\bullet} + \tilde{w}_k \tag{2.40a}$$

$$y_k = C_k x_k + \tilde{u}_k^{in} + \tilde{v}_k \tag{2.40b}$$

Therefore, the covariance matrix time dependency, the Kalman gain value and the measurement update equations can be derived in an analogous manner to the linear KF and result:

$$P_{k+1|k} = A_k P_{k|k} A_k^T + Q_k \tag{2.41}$$

$$K_{k+1} = P_{k+1|k} C_{k+1}^T \left(C_{k+1} P_{k+1|k} C_{k+1}^T + F_{k+1} R_{k+1} F_{k+1}^T\right)^{-1} \tag{2.42}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \left[y_{k+1} - h\left(\hat{x}_{k+1|k}, 0\right)\right] \tag{2.43}$$

$$P_{k+1|k+1} = \left(I - K_{k+1} C_{k+1}\right) P_{k+1|k} \tag{2.44}$$

Note that the EKF requires Jacobians evaluations at each time instant, therefore, it is expected to have a grater computational burden than its linear counterpart especially when implemented on high dimensional systems.

A resume of the extended Kalman filter, obtained by combining all the above equations into a single algorithm, is reported in Algorithm 2.

Because EKF is obtained using a linear approximation of the nonlinear system and making various stochastic assumptions on noise and disturbance that are rarely met in practice, it gives no guarantees of optimality and, at worst, may led to unacceptable performances.

Algorithm 2 Extended Kalman Filter

```
1: Initializations $(k = 0)$:
       $\hat{x}_{0|0} = E\left[x_0\right]$
       $P_{0|0} = E\left[\left(x_0 - \hat{x}_{0|0}\right)\left(x_0 - \hat{x}_{0|0}\right)^T\right]$
2: State equation Jacobian matrices:
       $A_k = \left.\frac{df}{dx}\right|_{\substack{x=\hat{x}_{k|k}\\ u=u_k\\ w=0}}$
       $G_k = \left.\frac{df}{dw}\right|_{\substack{x=\hat{x}_{k|k}\\ u=u_k\\ w=0}}$
3: Time updates:
       $\hat{x}_{k+1|k} = f\left(\hat{x}_{k|k}, u_k, 0\right)$
       $P_{k+1|k} = A_k P_{k|k} A_k^T + G_k Q_k G_k^T$
4: Measurement equation Jacobian matrices:
       $C_{k+1} = \left.\frac{dh}{dx}\right|_{\substack{x=\hat{x}_{k+1|k}\\ v=0}}$
       $F_{k+1} = \left.\frac{dh}{dv}\right|_{\substack{x=\hat{x}_{k+1|k}\\ v=0}}$
5: Filter gain:
       $K_{k+1} = P_{k+1|k} C_{k+1}^T \left(C_{k+1} P_{k+1|k} C_{k+1}^T + F_{k+1} R_{k+1} F_{k+1}^T\right)^{-1}$
6: Measurement updates:
       $\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}\left[y_{k+1} - h\left(\hat{x}_{k+1|k}, 0\right)\right]$
       $P_{k+1|k+1} = \left(I - K_{k+1} C_{k+1}\right) P_{k+1|k}$
7: Replace $k + 1$ with $k$ and go to Step 2
```

More refined linearization techniques can be used to reduce the linearization error and provide better estimation results, however, they do so at the price of higher complexity and computational expense.

For these reasons the use of this technique is justified if there exists a sufficiently large neighborhood in which the linearized model is a good representation of the nonlinear system and if the disturbances are well represented by zero mean state and measurement noise. In this case the optimal estimate for the linearized system provided by the EKF should be a reasonable approximation to the optimal estimate for the nonlinear system.

2.2.4 Tuning

As seen before, the computation of the Kalman gain requires an a priori knowledge about the process and the measurements noise covariances. If such statistics are known exactly, then the Kalman gain matrix takes its optimal value and the resulting state estimate is the better that can be achieved by the filter, otherwise things gradually get worse.

Concerning this, consider the Kalman gain expression:

$$K_k = P_{k|k-1} C_k^T \left(C_k P_{k|k-1} C_k^T + R_k \right)^{-1} \tag{2.45}$$

where $P_{k|k-1}$ depends on Q_k according to Equation 2.21.

Looking at Equation 2.45 it can be noted that as R_k tends to zero K_{k+1} increases, weighting the innovation term more heavily. Specifically:

$$\lim_{R_k \to 0} K_k = P_{k|k-1} C_k^T \left(C_k P_{k|k-1} C_k^T \right)^{-1} \tag{2.46}$$

Therefore, if the assumed noise covariances are such that R_k is negligible in Equation 2.45, then K_k will be too large. This leads to a deterioration of the filter performances and in the worst case to unstable or biased state estimates due to the measurement noise having too great an influence on them.

Conversely, as $P_{k|k-1}$ tends to zero K_k has limit zero and hence it weights the innovation term less heavily.

$$\lim_{P_{k|k-1} \to 0} K_k = 0 \tag{2.47}$$

Thus, if Q_k and R_k adopted make $P_{k|k-1}$ too small the Kalman gain will be small in turn and the state estimates will converge too slowly to their true values.

In most practical applications, while R_k might be determined from the a priori knowledge of the sensor characteristics, Q_k cannot be selected in an analogous deterministic way. The main reasons of such inability are that it might be affected by unknown modelling errors and the impossibility to measure the state vector directly.

Therefore, tuning the filter, i.e. choosing the covariances that provide the optimal value of the filter gain, is a challenging task.

Often Q_k and R_k are supposed to be time invariant and the tuning is performed by trial and error. In this case, initial values for the matrices are randomly picked from a set of candidates, which may result from the process knowledge or from information available in literature, and adjusted until the filter went to convergence. Then, the rest of the procedure consists in varying by trial and error the initial covariances found in order to improve the filter performances according to a predefined criterion. This tuning approach represents a considerable burden for the designer and often results very time consuming especially for

high dimensional systems.
Although several techniques have been proposed to overcome this issue (see e.g. Saha et al. [60] for an overview) the most interesting for the purposes of this thesis are the optimization based ones. These methods consist in formalizing the tuning problem as:

$$\begin{aligned} &\min_{Q,R} \textstyle\sum_k \ell^{tun} \\ \text{subject to: } &Q \geqslant 0 \\ &R \geqslant 0 \end{aligned} \tag{2.48}$$

and then in applying an efficient numerical algorithm for its resolution.
Several algorithms have been successfully implemented in literature, ranging from genetic [69] to downhill simplex [51], however, their usage must be carefully evaluated in dependence of the particular problem addressed.
The cost function must be properly selected in order to achieve the desired performances; in this sense well performing solutions for linear systems are the ones proposed by Odelson et al. [40] and Pannocchia et al. [45].
These methods provide the benefit of selecting tuning parameters in an automatic and optimal manner, however, they have the drawback of requiring an accurate selection of the objective function by the designer and may result computationally intractable for high dimensional nonlinear systems.

2.3 NONLINEAR OBSERVABILITY

A necessary condition for the state estimator convergence is that the system for which the states are to be estimated is *observable*. The basic idea of observability is that any two different states can be distinguished by applying some input and observing the two system outputs over some finite time interval [55].
Several equivalent definitions of observability have been developed for linear time-invariant (LTI) systems. Each one of them can be easily verified with different tests and lends itself to an explicit state reconstruction procedure. Researchers have tried to establish similar results for nonlinear systems, however, apart from a few particular generalizations, a completely parallel theory on nonlinear observability has reveal itself not feasible. Therefore, various weaker definitions of nonlinear observability have been developed, all with an emphasis on computational characterization, and their implication on the system structure [24].

2.3.1 Definitions

Since process and measurement noises do not affect the observability, they are set to zero and the following particularization of the continuous-time nonlinear system (2.2) is considered:

$$\dot{x} = f(x, u) \tag{2.49a}$$

$$y = h(x) \tag{2.49b}$$

where x takes values in a connected n-dimensional manifold $\mathcal{M}$, u in some open subset $\mathbb{U}$ of $\mathbb{R}^m$ and y in some open subset $\mathbb{Y}$ of $\mathbb{R}^p$. Functions $f : \mathcal{M} \times \mathbb{U} \times \mathbb{R}^+ \to \mathcal{M}$ and

$h : \mathcal{M} \times \mathbb{R}^+ \to \mathbb{Y}$ are assumed to be $\mathbb{C}^\infty$ with respect to their arguments, and input functions $u(\cdot)$ to be locally essentially bounded and measurable functions in the set $\mathbb{U}$. Equations which do not have solution for some valid inputs and initial conditions do not define system behavior completely and hence are inappropriate for use in observability analysis. For this reason it is also assumed that the system is *complete*, that is, for every admissible input u and every $x_0 \in \mathcal{M}$ there exists a solution to Equation 2.49a such that $x(0) = x_0$ and $x \in \mathcal{M}$ for all $t \in \mathbb{R}^+$.
According to what stated at the beginning of this section, observability is characterized by the fact that one cannot admit indistinguishable states; such requirements can be formalized as follows.

Definition 1 (Indistinguishable states)**.** Let $y(t, t_0, x_0, u(t))$ denote the output trajectory from an initial state x_0, initial time t_0, and input $u(t)$ for the continuous-time nonlinear system (2.49). Two states x_0 and x_1 in $\mathcal{M}$ are called *indistinguishable* if for every admissible input $u(t)$ defined on the interval $[t_0, t_f]$ identical outputs result: $y(t, t_0, x_0, u(t)) \equiv y(t, t_0, x_1, u(t))\ \forall t \in [t_0, t_f]$.

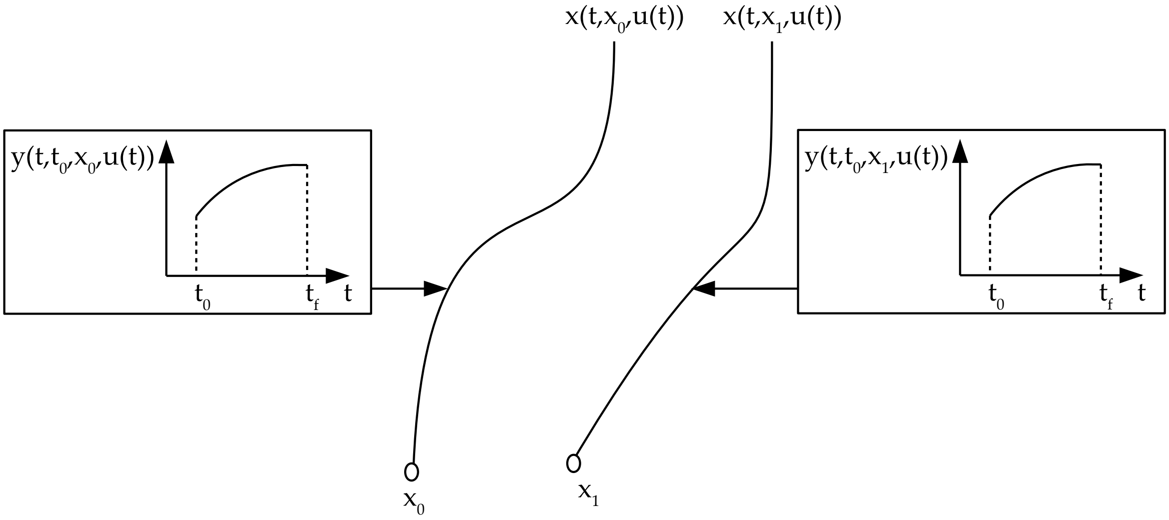


Figure 5: Indistinguishable states.

From this, denoted as $I(x_0, \mathcal{M})$ the set of all states in $\mathcal{M}$ that are indistinguishable from x_0, observability can be defined.

Definition 2 (Observability)**.** The nonlinear system (2.49) is *observable at* $x_0 \in \mathcal{M}$ if $I(x_0, \mathcal{M}) = x_0$.
The same nonlinear system is *globally observable* if it is observable at x for all $x \in \mathcal{M}$.

Note that observability of a nonlinear system does not exclude the possibility of states indistinguishable by certain inputs $u(t)$ in the interval $[t_0, t_f]$. To avoid any such eventuality there have to be additional constraints on the inputs.

Definition 3 (Universal input)**.** An input $u \in \mathbb{U}$ is *universal* on $[t_0, t_f]$ if for every pair of initial condition x_0 and x_1 there exists $t^* \in [t_0, t_f]$ such that $y(t^*, t_0, x_0, u(t)) \not\equiv y(t^*, t_0, x_1, u(t))$.

A non-universal input is called a *singular input*. If the nonlinear system (2.49) is globally observable and all inputs $u(t)$ are universal the reconstruction of x from measurement data can be possible for all inputs $u(t)$ in the time interval $[t_0, t_f]$. Such a system is said to be *uniformly observable*.

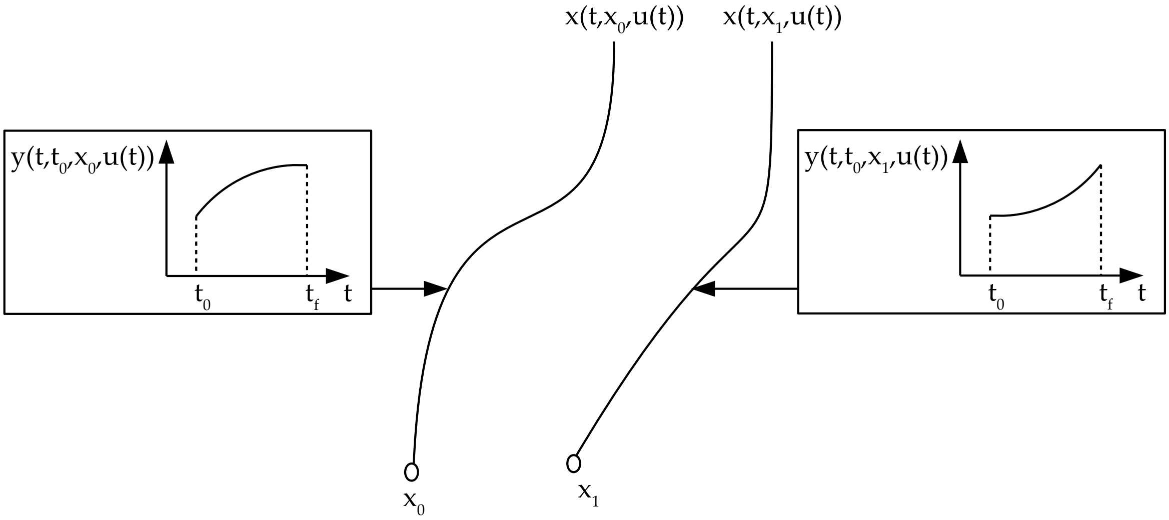


Figure 6: Observable system at x_0.

Definition 4 (Uniform observability)**.** The nonlinear system (2.49) is *uniformly observable* if it is globally observable and all inputs $u(t)$ are universal.

Observability, according to the above definitions, is a global concept; it might be necessary to compute state trajectories that are very far from $x(t_0)$ or in which t_f is very large to distinguish between two states. Therefore, a local concept of observability is introduced which restricts state trajectories to a neighborhood of $x(t_0)$ and implicitly sets a limit to t_f as well.

Definition 5 (Local observability)**.** The nonlinear system (2.49) is *locally observable at* $x_0 \in \mathcal{M}$ if for every open neighborhood $\mathcal{U}$ of x_0 and for every solution $x(t)$ completely in $\mathcal{U}$ it is verified that $I(x_0, \mathcal{U}) = x_0$.
The same nonlinear system is *locally observable* if it is locally observable at x for all $x \in \mathcal{M}$.

Remark 1. In other words, the system is locally observable if every state can be distinguished from every other state in $\mathcal{U}$ by using system trajectories remaining close to the state.

Both definitions above ensure that a state $x_0 \in \mathcal{M}$ can be distinguished from every other state in $\mathcal{M}$ but for practical purposes it is often enough to be able to distinguish between neighbors in $\mathcal{M}$. This leads us to the following weaken concept of observability:

Definition 6 (Weak observability)**.** The nonlinear system (2.49) is *weakly observable at* $x_0 \in \mathcal{M}$ if there is some neighborhood $\mathcal{V}$ of x_0 such that $I(x_0, \mathcal{M}) \cap \mathcal{V} = x_0$.
The same nonlinear system is *weakly observable* if such a neighborhood $\mathcal{V}$ exists for all $x \in \mathcal{M}$.

Once again it might be necessary to compute state trajectories that are very far from $\mathcal{V}$ to distinguish states of $\mathcal{V}$. Therefore, even in this case, a local concept of weak observability is introduced.

Definition 7 (Local weak observability)**.** The nonlinear system (2.49) is *locally weakly observable at* $x_0 \in \mathcal{M}$ if there is some neighborhood $\mathcal{V}$ of x_0 such that $I(x_0, \mathcal{M}) \cap \mathcal{V} = x_0$ for all solutions $x(t)$ completely in any neighborhood $\mathcal{U}$ of x_0.
The same nonlinear system is *locally weakly observable* if it is locally weakly observable at x for all $x \in \mathcal{M}$.

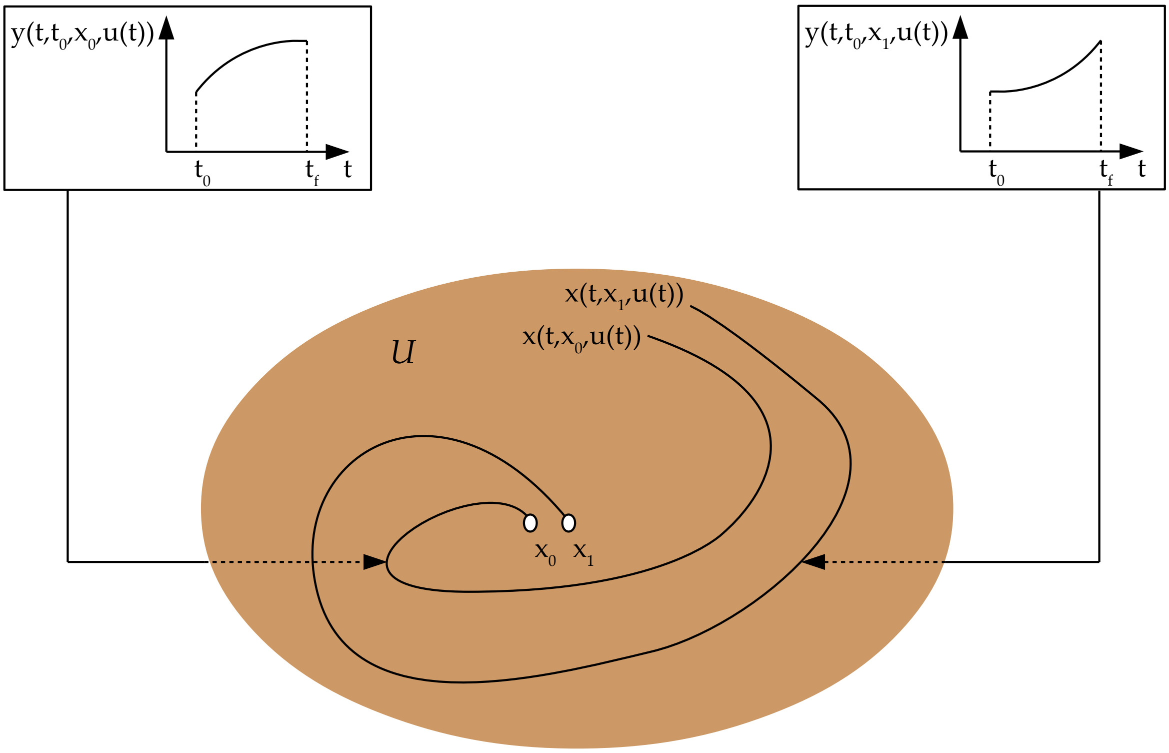


Figure 7: Locally observable system at x_0.

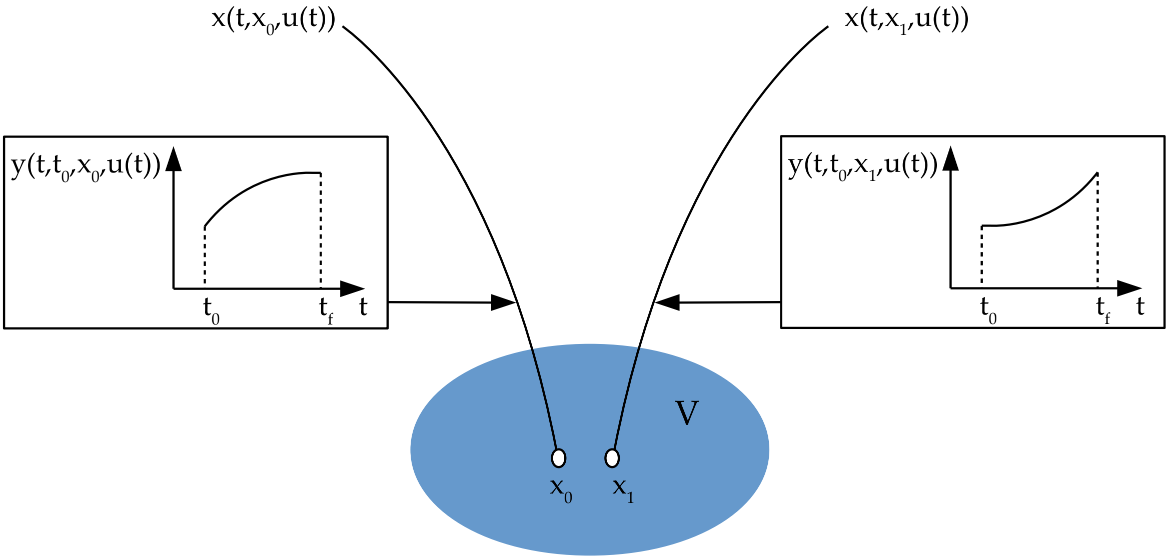


Figure 8: Weakly observable system at x_0.

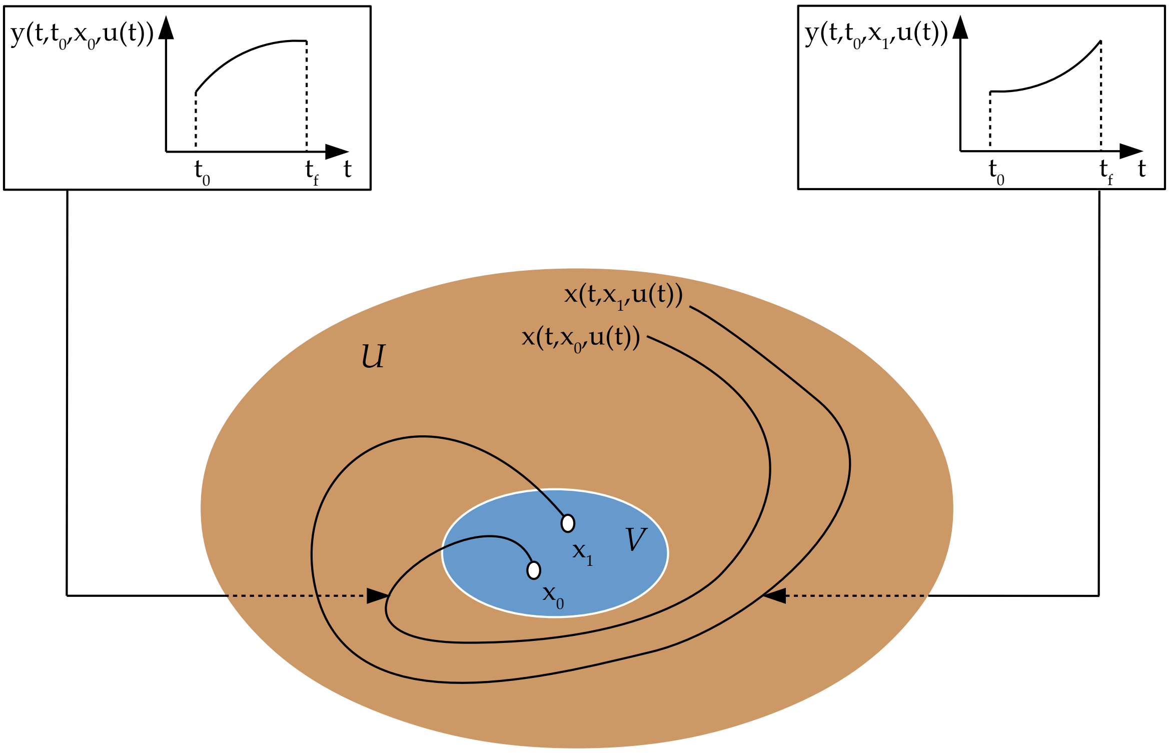


Figure 9: Locally weakly observable system at x_0.

$$\begin{tikzcd}
\text{Local observability} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{Global observability} \arrow[d, Rightarrow] \\
\text{Local weak observability} \arrow[r, Rightarrow] & \text{Weak observability}
\end{tikzcd}$$

Figure 10: Relationships between the various form of observability.

Remark 2. Roughly speaking, the system is locally weakly observable if every state can be distinguished from its neighbors by using system trajectories remaining close to the state.

By setting $\mathcal{U}$ in Definition 5 equal to $\mathcal{M}$ it can be verified that local observability implies global observability. Furthermore, by placing $\mathcal{V}$ in Definition 7 equal to $\mathcal{M}$ it is possible to demonstrate that it involves also local weak observability. Therefore, local observability is a stronger property than global and local weak observability.
Finally, by posing $\mathcal{V}$ in Definition 6 and $\mathcal{U}$ in Definition 7 equal to $\mathcal{M}$ it can be shown that both global and local weak observability implicate weak observability. These findings are summarized in Figure 10.

2.3.2 Observability tests

In order to establish a priori whether or not a nonlinear system is observable there is the need to characterize from a mathematical viewpoint the previous definitions.
Concerning this, the local weak observability, in addition to being of more practical interest with respect to the other forms of observability, presents the advantage of admitting several computation characterizations. Most notable ones are reviewed in this section.

Observability rank condition

The observability test presented here will be developed in terms of linear combinations of vector fields on a smooth manifold $\mathcal{M}$. Therefore, the formulation is limited to a particular class of nonlinear systems known as *control affine systems*. Note that this is not a strong restriction since many real-world nonlinear systems are well described by this type of models.
Consider the following control affine system obtained as a particularization of the general nonlinear system (2.49):

$$\dot{x} = f(x) + g(x)u \tag{2.50a}$$

$$y = h(x) \tag{2.50b}$$

and let $x_0 = x(0)$ be its initial condition.
Assuming that f, g and h are analytic functions it is possible to compute the Taylor series expansion of the output at $t = 0$ as:

$$\begin{aligned} y &= \sum_{k=0}^{\infty} \left.\frac{d^k y}{dt^k}\right|_{t=0} \frac{t^k}{k!} \\ &= h(x)|_{t=0} + \left.\frac{dy}{dt}\right|_{t=0} t + \left.\frac{d^2 y}{dt^2}\right|_{t=0} \frac{t^2}{2} + \text{H.O.T.} \end{aligned} \tag{2.51}$$

where H.O.T. stands for higher order terms.
Recalling from subsection A.2.3 the notion of Lie derivative of a vector function, it is straightforward to verify that:

$$\begin{aligned} \left.\frac{dy}{dt}\right|_{t=0} &= \left[\frac{\partial h(x)}{\partial x}\frac{dx}{dt}\right]_{t=0} \\ &= \left[\frac{\partial h(x)}{\partial x}(f(x) + g(x)u)\right]_{t=0} \\ &= L_f[h(x_0)] + L_g[h(x_0)]u(0) \end{aligned} \tag{2.52}$$

and so on for the higher order derivatives.
Substituting such expressions in Equation 2.51 yields:

$$y = h(x_0) + \left\{ L_f\left[h(x_0)\right] + L_g\left[h(x_0)\right]u(0) \right\} t + \left\{ L_f^2\left[h(x_0)\right] + L_f\left\{L_g\left[h(x_0)\right]\right\}u(0) + \right. \\ \left. + L_g\left\{L_f\left[h(x_0)\right]\right\}u(0) + L_g^2\left[h(x_0)\right]u^2(0) + L_g\left[h(x_0)\right]\left.\frac{du}{dt}\right|_{t=0} \right\}\frac{t^2}{2} + \text{H.O.T.} \tag{2.53}$$

All the output functions as well as all their derivatives along the system trajectories are contained in the so called *observation space*.

Definition 8 (Observation space)**.** The observation space for a nonlinear system (2.50) is defined as the smallest real vector space (denoted by $\mathcal{G}$) of $\mathbb{C}^\infty$ functions containing the components of $h(x)$ and closed under Lie derivation along system trajectories for any constant $u \in \mathbb{R}^m$.

By referring to Equation 2.53:

$$\mathcal{G} = \left\{ h(x_0), L_f\left[h(x_0)\right], L_g\left[h(x_0)\right], L_f\left\{L_g\left[h(x_0)\right]\right\}, L_g\left\{L_f\left[h(x_0)\right]\right\}, L_f^2\left[h(x_0)\right], \right. \\ \left. , L_g^2\left[h(x_0)\right], \dots \right\} \tag{2.54}$$

Consider now a point $x_0 + \delta x$ in open neighborhood $\mathcal{V}$ of x_0 and let it be the new initial condition. By computing the Taylor series expansion of the new output at $t = 0$ and subtracting Equation 2.53 from the resulting expression it is possible to quantify the output variation δy due to the change in the initial condition:

$$\delta y = dh(x_0)\delta x + dL_f\left[h(x_0)\right]\delta x t + dL_g\left[h(x_0)\right]\delta x t + dL_f\left\{L_g\left[h(x_0)\right]\right\}u_0\delta x\frac{t^2}{2} + \text{H.O.T.} \tag{2.55}$$

Thus, according to Definition 1 on the indistinguishability of two initial states, x_0 and $x_0 + \delta x$ will be indistinguishable if for every $u \in \mathbb{U}$ defined on the time interval $[0, t_f]$ results:

$$\delta y = 0 \tag{2.56}$$

from which follows:

$$\Theta(x_0)\delta x = 0 \tag{2.57}$$

where:

$$\Theta(x) = \begin{bmatrix} dh(x) \\ dL_f\left[h(x)\right] \\ dL_g\left[h(x)\right] \\ dL_f\left\{L_g\left[h(x)\right]\right\} \\ \vdots \end{bmatrix} \tag{2.58}$$

The space spanned by the rows of $\Theta(x)$ takes the name of *observability codistribution*, $d\mathcal{G}$.

Theorem 1 (Observability rank condition)**.** *If $\Theta(x_0)$ contains n linear independent row vectors, i.e.:*

$$\text{rank}\,\Theta(x_0) = n \tag{2.59}$$

then the nonlinear continuous time system (2.50) is locally weakly observable at x_0.
More generally, the same nonlinear system satisfying:

$$\text{rank}\,\Theta(x) = n \tag{2.60}$$

is locally weakly observable.

In other words, if $\Theta(x)$ has rank equal to n there exist no nontrivial variations of the initial condition δx such that the output and its derivatives along the system trajectories remain unchanged independently on the input applied.
Notice finally that when $f(x)$ and $h(x)$ are linear functions and g is constant with respect to x the observability rank condition particularized in the so called *Kalman rank condition*.

Kalman rank condition on the linearized system

Consider the continuous-time nonlinear system (2.49) and let (x^s, u^s) be a steady state operating point, i.e.:

$$f(x^s, u^s) = 0 \tag{2.61}$$
$$h(x^s) = y^s \tag{2.62}$$

Performing the Taylor series expansion of (2.49) around (x^s, u^s) yields:

$$\dot{x} = \cancel{f\left(x^s, u^s\right)} + \left.\frac{df}{dx}\right|_{\substack{x=x^s\\u=u^s}} \left(x - x^s\right) + \left.\frac{df}{du}\right|_{\substack{x=x^s\\u=u^s}} \left(u - u^s\right) + \text{H.O.T.} \tag{2.63}$$
$$y = h\left(x^s\right) + \left.\frac{dh}{dx}\right|_{x=x_s} \left(x - x^s\right) + \text{H.O.T.} \tag{2.64}$$

For x sufficiently close to x^s, the higher order terms will be very close to zero, and hence can be neglected to obtain:

$$\dot{x} \approx A\delta x + B\delta u \tag{2.65}$$
$$\delta y \approx C\delta x \tag{2.66}$$

where:

$$A = \left.\frac{\partial f}{\partial x}\right|_{\substack{x=x^s\\u=u^s}} \tag{2.67}$$
$$B = \left.\frac{\partial f}{\partial u}\right|_{\substack{x=x^s\\u=u^s}} \tag{2.68}$$
$$C = \left.\frac{\partial h}{\partial x}\right|_{x=x^s} \tag{2.69}$$
$$\delta x = x - x^s \tag{2.70}$$
$$\delta u = u - u^s \tag{2.71}$$
$$\delta y = y - y^s \tag{2.72}$$

Then, by recalling Equation 2.61 and by using the fact that:

$$\delta\dot{x} = \dot{x} \tag{2.73}$$

the following linearized system results:

$$\delta\dot{x} = A\delta x + B\delta u \tag{2.74a}$$
$$\delta y = C\delta x \tag{2.74b}$$

Now, let $\delta x_0 = x_0 - x^s$ be the initial state to be determined. Since δx_0 has n unknown components and the system (2.74) is linear, it is expected that n measurements y_k taken at different sample times t_k are sufficient to figure out the initial state value:

$$\begin{aligned}
\delta y_0 &= C\delta x_0 \\
\delta y_1 &= CA\delta x_0 + CB\delta u_0 \\
&\vdots \\
\delta y_{n-1} &= CA^{n-1}\delta x_0 + \sum_{j=1}^{n-2} \left(CA^j B\delta u_{n-2-j}\right)
\end{aligned}$$

or in matrix form:

$$Y = \mathcal{O}\delta x_0 \tag{2.75}$$

where:

$$Y = \begin{bmatrix} \delta y_0 \\ \delta y_1 - CB\delta u_0 \\ \vdots \\ \delta y_{n-1} - \sum_{j=1}^{n-2} \left(CA^j B\delta u_{n-2-j}\right) \end{bmatrix} \tag{2.76}$$

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{2.77}$$

From the fundamental theorem of linear algebra the system of linear algebraic equations with n unknowns (2.75) has a unique solution if and only if the columns of the system matrix are linearly independent. Thus, the initial condition x_0 is completely determined if the so-called *observability matrix* $\mathcal{O} \in \mathbb{R}^{(n\cdot p)\times n}$ is full rank.
The previous derivations can be summarized in the following theorem:

Theorem 2 (Kalman rank condition)**.** *The LTI system* (2.74) *is observable if and only if*

$$\mathrm{rank}\,(\mathcal{O}) = n \tag{2.78}$$

If $\mathcal{O}$ is not full rank, any initial state with a component in the nullspace of $\mathcal{O}$ cannot be uniquely determined from the output measurements. In this case the linearized system (2.74) is not locally weakly observable at x_0 and the null space of $\mathcal{O}$ is called the *unobservable subspace*.
According to Ray [56], observability of nonlinear systems is often determined by the structure of the system and is not dependent on the state in a complex manner. Consequently, such linearized observability test is usually adequate for many practical applications.
However, it must be noted that local weak observability of the linearized system (2.74) may not imply local weak observability of the original nonlinear system (2.49) as $\mathcal{O}$ is an approximation to the matrix $\Theta(x)$ defined in Equation 2.58.

Observability covariance matrix rank condition

As every real process has a region of operation, it is important for the state estimator design to take into account information about the observability of the system over such

region. Since the Kalman rank condition can be checked only for those points which lie in a sufficiently small neighborhood of the operating point it reveals itself unsuitable for this purpose. At the same time it is not currently feasible to compute the Lie-algebra based observability matrix for systems of medium or even large scale and which does not belong to the control-affine class. Both of this issues have been addressed by Hahn and Edgar [17] who developed the so called *observability covariance matrix rank condition*.
Observability covariance matrix is a powerful tool for analyzing observability of nonlinear systems as it is naturally computed from experimental or simulation data, collected within a region of operation where the process has to be controlled. At the same time, it has been successfully applied to system with dozen or even hundreds of states and extracting information from observability covariance matrix is no more challenging than it is for linear systems.
The following sets need to be defined for observability covariance matrix calculation:

$$\begin{aligned} T^n &= \{T_1, \dots, T_n; \quad T_i \in \mathbb{R}^{n\times n},\ T_i^T T_i = I,\ i = 1, \dots, r\} \\ M &= \{c_1, \dots, c_s; \quad c_i \in \mathbb{R},\ c_i > 0,\ i = 1, \dots, s\} \\ E^n &= \{e_1, \dots, e_n; \quad \text{standard unit vectors in } \mathbb{R}^n\} \end{aligned}$$

where r is the number of matrices for perturbation directions, s is the number of different perturbation sizes for each direction and n is the number of states of the system.
Usually, T is chosen to be $\{I, -I\}$, where the submatrix $\{I\}$ refers to positive perturbations in the state variables while $\{-I\}$ corresponds to decreases in the value of the state variables. More complicated choice for T are possible for some nonlinear systems as long as they satisfy the conditions stated above. At the same time, the elements of M must be chosen in such a way as to capture the nonlinear behavior of the system at a certain distance from the nominal steady-state operating point (x^s, u^s).

Definition 9 (Observability covariance matrix)**.** Let T^n, E^n and M be given sets as described above. The *observability covariance matrix* is defined by:

$$W_O = \sum_{l=1}^{r} \sum_{m=1}^{s} \frac{1}{rsc_m^2} \int_0^\infty T_l \Psi^{lm}(t) T_l^T \, dt \tag{2.79}$$

where $\Psi^{lm}(t) \in \mathbb{R}^{n\times n}$ is given by:

$$\Psi_{ij}^{lm}(t) = \left(y^{ilm}(t) - y_{ss}^{ilm}\right)^T \left(y^{jlm}(t) - y_{ss}^{jlm}\right) \tag{2.80}$$

$y^{ilm}(t)$ is the output of the nonlinear system corresponding to the perturbed initial condition $x(0) = x^s + c_m T_l e_i$ with input $u = u^s$ and y_{ss}^{ilm} is the steady-state of the output that the system will reach after this perturbation.

It is now possible to formulate the following test.

Theorem 3 (Observability covariance matrix rank condition)**.** *The nonlinear continous time system* (2.49) *is observable at a certain distance from a nominal steady-state operating point if the associated observability covariance matrix, W_O, has full rank.*

Remark 3. The output trajectories generated by simulation are computed only for nominal value of the input. Hence, observability analysis by the observability covariance matrix is applicable only for nominal value of the input.

Suppose that the initial state $x(0)$ is within the *region of attraction* of the operating point, defined as follows.

Definition 10 (Region of attraction)**.** The region of attraction of a system is the set of initial states $x(0)$ from which the system converges to the equilibrium point (x^s, u^s).

In this case, the observability covariance matrix reduces to the *empirical observability gramian*.

Definition 11 (Empirical observability gramian)**.** Let T^n, E^n and M be given sets as described above. The *empirical observability gramian* is defined by:

$$W_O = \sum_{l=1}^{r} \sum_{m=1}^{s} \frac{1}{rsc_m^2} \int_0^{\infty} T_l \Psi^{lm}(t) T_l^T \, dt \qquad (2.81)$$

where $\Psi^{lm}(t) \in \mathbb{R}^{n \times n}$ is given by:

$$\Psi_{ij}^{lm}(t) = \left(y^{ilm}(t) - y_s^{ilm}\right)^T \left(y^{jlm}(t) - y_s^{jlm}\right) \qquad (2.82)$$

and $y^{ilm}(t)$ is the output of the nonlinear system corresponding to the perturbed initial condition $x(0) = x^s + c_m T_l e_i$ with input $u = u^s$. The y_s^{ilm} refers to the steady-state value of the output of the system before this perturbation.

If the system under investigation is linear and stable, any initial condition will remain within the region of attraction of x^s and the observability covariance matrix is identical to the empirical observability gramian. Because it has been proven (see Lall et al. [30]) that empirical observability gramian reduces to the observability gramian of the linearized system for small perturbations around the operating point, it can be ensured that the observability results from Theorem 3 will locally match the ones derived for a linearized system.

2.3.3 Degree of observability

The tests presented in the previous sections can be used for checking observability of nonlinear systems but they only gives a yes or no answer; they do not quantify how observable or unobservable the system is. In order to accomplish this end measures for the *degree of observability* of nonlinear systems have been derived. Singh and Hahn [64] make an overview of several of these that is reported below.
The first presented is:

$$\mu_1 = \lambda_{min}(W_O) \qquad (2.83)$$

where the eigenvector corresponding to the smallest eigenvalue λ_{min} refers to the least observable direction of the system and the eigenvalue corresponds to the degree of observability of this worst direction. Higher values of this measure imply higher degree of observability of the least observable direction.
Another measure commonly used is based on the trace of the observability covariance matrix inverse:

$$\mu_2 = \frac{n}{\operatorname{trace}\left(W_O^{-1}\right)} \qquad (2.84)$$

where n refers to the number of states of the system.
Recalling from the fundamentals of linear algebra that the trace of a matrix is the sum of

its eigenvalues and that the eigenvalues of W_O^{-1} are the reciprocals of the eigenvalues of W_O results:

$$\mu_2 = \left(\frac{\sum_{i=1}^{n} \frac{1}{\lambda_i}}{n} \right)^{-1} \tag{2.85}$$

Thus, μ_2 can also be viewed as the inverse arithmetic mean of the reciprocals of the eigenvalues of W_O. Larger values of μ_2 indicate an higher degree of observability.
An ulterior alternative is to use the determinant of the observability covariance matrix:

$$\mu_3 = \left[\det\left(W_O\right)\right]^{\frac{1}{n}} \tag{2.86}$$

since the determinant is the product of the eigenvalues of a matrix, Equation 2.86 can be rewritten as follows:

$$\mu_3 = \sqrt[n]{\prod_{i=1}^{n} \lambda_i} \tag{2.87}$$

From Equation 2.87 can be seen that μ_3 is equal to the geometric mean of the eigenvalues of W_O. A value of the determinant of zero indicates that the matrix is rank deficient, corresponding to an unobservable system. Similarly, larger value of the determinant are an indicator of the increased observability.
A further measure that can be used is the condition number (CN) of the observability covariance matrix, defined as:

$$CN = \frac{\sigma_{max}\left(W_O\right)}{\sigma_{min}\left(W_O\right)} \tag{2.88}$$

where σ_{min} and σ_{max} refer to the smallest and largest singular values of the observability covariance matrix. Since W_O is symmetric, σ_{min} and σ_{max} are identical to the smallest and largest eigenvalues respectively.
A smaller value of the condition number generally implies increased observability of a system.
One may also think to make use only of the smallest singular value of the observability covariance matrix:

$$NS = \sigma_{min}\left(W_O\right) \tag{2.89}$$

NS is similar to the measures based on the smallest eigenvalues given in Equation 2.83, Equation 2.84 and Equation 2.86 as it serves as an indicator of how far the system is from being unobservable. Higher values of this criterion imply an increased degree of observability.
All the above measures are strongly influenced by the smallest singular value or the smallest eigenvalues. While it can be useful to ensure that a minimum degree of observability exists even for the worst directions, this may not always be the best measures. The reason for this is that in some application (e.g. sensor placement) it is unlikely that every single state of the system needs to be observed; instead, the main focus should be on ensuring that the most important states can be observed easily, e.g., measures which are strongly influenced by the smallest eigenvalue of singular value can return misleading information if some states that are not important for plant operation may be unobservable [64].
A measure based upon these ideas is for example the spectral radius of the covariance observability matrix:

$$\rho\left(W_O\right) = \lambda_{max}\left(W_O\right) = \sigma_{max}\left(W_O\right) \tag{2.90}$$

Large values of $\rho\left(W_O\right)$ indicate that the dominant direction in the observability covariance matrix can be easily observed.
The final measure reported is the trace of the observability covariance matrix:

$$\operatorname{trace}\left(W_O\right) = \sum_{i=1}^{n} \sigma_i\left(W_O\right) = \sum_{i=1}^{n} \lambda_i\left(W_O\right) \tag{2.91}$$

A large value of the trace corresponds to an increase in the overall observability of the system.

Table 1: Measures for degree of observability

Measure	Benchmark	Equation
μ_1	Smallest Eigenvalue	(2.83)
μ_2	Smallest Eigenvalue	(2.84)
μ_3	Smallest Eigenvalue	(2.86)
CN	Smallest Eigenvalue	(2.88)
NS	Smallest Eigenvalue	(2.89)
$\rho\left(W_O\right)$	Largest Eigenvalue	(2.90)
$\operatorname{trace}\left(W_O\right)$	Largest Eigenvalue	(2.91)

Summing up, the presented measures can be divided into two main categories: measures which are mainly based upon the least observable direction in state space, and measures which are predominantly influenced by the largest eigenvalue of the observability covariance matrix. These findings are summarized in Table 1.

2.4 OFFSET-FREE CONTROL

Suppose to have to control a time-invariant dynamical system of the form:

$$x^\star_{k+1} = f^\star\left(x^\star_k, u_k, w^\star_k\right) \tag{2.92a}$$

$$y^\star_k = h^\star\left(x^\star_k, v^\star_k\right) \tag{2.92b}$$

where $x^\star_k \in \mathbb{R}^n$ is the plant state, $u^\star_k \in \mathbb{R}^m$ is the control input, $y^\star_k \in \mathbb{R}^p$ is the plant output, $w^\star_k \in \mathbb{R}^{n_w}$ and $v^\star_k \in \mathbb{R}^{n_v}$ denote plant states and output disturbances. The plant output is measured at each sample time t_k and the functions $f^\star : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{n_w} \to \mathbb{R}^n$ and $h^\star : \mathbb{R}^n \times \mathbb{R}^{n_v} \to \mathbb{R}^p$ are assumed to be continuous.
The following particularization of the discrete-time nonlinear system (2.3) is taken as a representation of the plant dynamics:

$$x_{k+1} = f\left(x_k, u_k\right) \tag{2.93a}$$

$$y_k = h\left(x_k\right) \tag{2.93b}$$

Disturbances at any sample time t_k are defined as:

$$w^\star_k = f^\star\left(x^\star_k, u_k, w^\star_k\right) - f\left(x_k, u_k\right) \tag{2.94}$$

$$v^\star_k = h^\star\left(x^\star_k, v^\star_k\right) - h\left(x_k\right) \tag{2.95}$$

and are assumed to be asymptotically constant and bounded in compact sets: $w_k^* \in \mathbb{W}$, $v_k^* \in \mathbb{V}$.
Moreover, input and output are required to satisfy the following constraints at all times:

$$g_u(u_k) \leqslant 0 \tag{2.96}$$

$$g_y(y_k) \leqslant 0 \tag{2.97}$$

where $g_u : \mathbb{R}^m \to \mathbb{R}^{q_u}$, and $g_y : \mathbb{R}^p \to \mathbb{R}^{q_y}$ are convex functions defining the following compact convex sets:

$$\mathbb{U} := \{u_k \in \mathbb{R}^m \mid g_u(u_k) \leqslant 0\} \tag{2.98}$$

$$\mathbb{Y} := \{y_k \in \mathbb{R}^p \mid g_y(y_k) \leqslant 0\} \tag{2.99}$$

Induced by the process model and the output constraint set, the following state constraint set results:

$$\mathbb{X} := \{x_k \in \mathbb{R}^n \mid g_y(h(x_k)) \leqslant 0\} \tag{2.100}$$

Define the controlled output $y_c \in \mathbb{R}^{p_c}$ as a function of the measured output:

$$y_c = r_y(y_k) \tag{2.101}$$

and denote with $\bar{y}_c \in \mathbb{Y}$ its desired set-point.
The offset-free control problem consists in planning an output feedback MPC law:

$$u_k = \kappa(y_k) \tag{2.102}$$

such that:

1. Input and output constraints are satisfied at all times.
2. The closed-loop system reaches an equilibrium.
3. The following condition holds true:

$$\lim_{k\to\infty} y_c = \bar{y}_c \tag{2.103}$$

2.4.1 Choice of the disturbance model and state augmentation

The general approach to achieve offset-free performance, motivated by the works of Davison and Smith [11], Kwakernaak and Sivan [28] and the Internal Model Principle of Francis and Wonham [15], is to augment the system state with an integrating disturbance term and design a controller that can remove asymptotically constant, nonzero disturbances.
To this end, the representation of the process dynamics (2.93) is augmented with a disturbance model to give an *augmented model* whose general form can be written as:

$$x_{k+1} = F(x_k, u_k, d_k) \tag{2.104a}$$

$$d_{k+1} = d_k \tag{2.104b}$$

$$y_k = H(x_k, d_k) \tag{2.104c}$$

where $d \in \mathbb{R}^{n_d}$ is the so-called *disturbance state* or simply disturbance and $F : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{n_d} \to \mathbb{R}^n$ and $H : \mathbb{R}^n \times \mathbb{R}^{n_d} \to \mathbb{R}^p$ are supposed to be continuous, and consistent with the nominal model, i.e. for all $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ there holds:

$$\begin{aligned} F(x_k, u_k, 0) &= f(x_k, u_k) \\ H(x_k, 0) &= h(x_k) \end{aligned} \tag{2.105}$$

Moreover, it is assumed that the augmented system is uniformly observable.

Note that the disturbance model does not necessarily have to represent the real disturbances acting on the plant for offset-free control: under the assumptions made its presence is sufficient to move the state of the original system onto a manifold that cancels disturbance effects on the controlled variables [47]. For this reason the disturbance model design is sometimes done using standard solutions; most notable ones are the *pure Output Disturbance Model (ODM)* and the *pure Input Disturbance Model (IDM)*.

The pure output disturbance model adds one disturbance state for every output variable:

$$x_{k+1} = f(x_k, u_k) \tag{2.106a}$$

$$d_{k+1} = d_k \tag{2.106b}$$

$$y_k = h(x_k) + d_k \tag{2.106c}$$

Although simplistic, that solution offers the benefits to accurately model set-point variations, which often enter feedback loops as step disturbances, and to provide zero offset for step changes in reference signal. Moreover, since errors in the model can appear as slowly varying output disturbances, it provides robustness to model error. In spite of that, the framework suffers from observability issues for plant containing integrators, because the effects of the plant integrating mode and of the output disturbance cannot be distinguished. In addition, since it models poorly actual disturbance dynamics for many practical systems, closed-loop dynamic performances are often unsatisfactory [37].

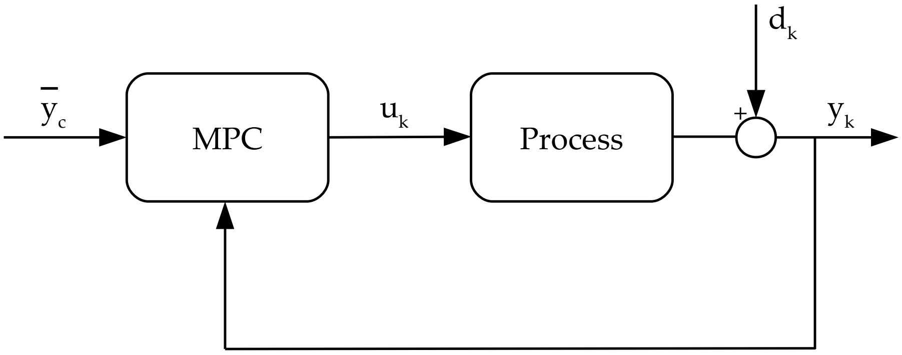


Figure 11: Output disturbance model.

An equally simple method is the pure input disturbance model:

$$x_{k+1} = f(x_k, u_k + d_k) \tag{2.107a}$$

$$d_{k+1} = d_k \tag{2.107b}$$

$$y_k = h(x_k) \tag{2.107c}$$

With that choice the difference between the predicted and the actual plant output is assumed to be caused by a step disturbance on the input, which remains constant in the future. Since in this model disturbances enter upstream of the process, they are filtered by the plant dynamics and resemble real disturbances in many practical applications. Furthermore, the input disturbance model is very effective in rejecting disturbances with slow dynamics and in rendering the MPC regulator not sensitive to input uncertainty with an increment of the controller robustness for ill-conditioned process [48].

The main deficiency of this representation is that it can be used only for systems with the same number of inputs and outputs.

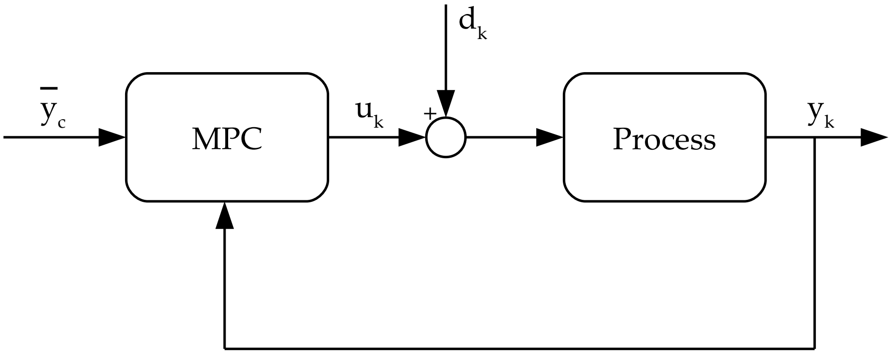


Figure 12: Input disturbance model.

2.4.2 State estimator design

In order to treat this topic there is a need of some preliminary definitions.

Definition 12 ($\mathcal{K}$ function)**.** A function $\mathcal{F} : \mathbb{R}^+ \to \mathbb{R}^+$ belongs to the class $\mathcal{K}$ if it is continuous, zero at zero, and strictly increasing.

Definition 13 ($\mathcal{KL}$ function)**.** A function $\mathcal{F} : \mathbb{R}^+ \times \mathbb{I}^+ \to \mathbb{R}^+$ belongs to the class $\mathcal{KL}$ if for each $k \geqslant 0$ the function $\mathcal{F}(\cdot, k) \in \mathcal{K}$, and for each $\alpha \in \mathbb{R}^+$ there holds:

$$\lim_{k\to\infty} \mathcal{F}(\alpha, k) = 0 \tag{2.108}$$

Definition 14 (Asymptotic stability of the estimate)**.** Let $\hat{x}_k$ be an estimate of x_k obtained for the nonlinear system (2.93), given a sequence of output measurements $Y = \{y_0, \dots, y_k\}$ and a (prior) estimate of the initial state $\hat{x}_0$. The estimate is asymptotically stable if there exists $\mathcal{F} \in \mathcal{KL}$ such that for all initial state x_0, prior estimate $\hat{x}_0$ and $k \in \mathbb{I}^+$ there holds:

$$\left|x_k - \hat{x}_k\right| \leqslant \mathcal{F}\left(\|x_0 - \hat{x}_0\|, k\right) \tag{2.109}$$

Without loss of generality, it is assumed that the augmented state $\left[x_k^T, d_k^T\right]^T$ is estimated from the output measurement y_k at each time t_k by using an observer of the following form:

$$\hat{x}_{k|k} = F\left(\hat{x}_{k-1|k-1}, u_{k-1}, \hat{d}_{k-1|k-1}\right) + \kappa^x\left(y_k - H\left(\hat{x}_{k|k-1}, \hat{d}_{k|k-1}\right)\right) \tag{2.110a}$$

$$\hat{d}_{k|k} = \hat{d}_{k|k-1} + \kappa^d\left(y_k - H\left(\hat{x}_{k|k-1}, \hat{d}_{k|k-1}\right)\right) \tag{2.110b}$$

Having defined the predicted estimates:

$$\hat{x}_{k|k-1} = F\left(\hat{x}_{k-1|k-1}, u_{k-1}, \hat{d}_{k-1|k-1}\right) \tag{2.111a}$$

$$\hat{d}_{k|k-1} = \hat{d}_{k-1|k-1} \tag{2.111b}$$

$$\hat{y}_{k|k-1} = H\left(\hat{x}_{k|k-1}, \hat{d}_{k|k-1}\right) \tag{2.111c}$$

and the output prediction error as:

$$e_k = y_k - H\left(\hat{x}_{k|k-1}, \hat{d}_{k|k-1}\right) \tag{2.112}$$

Equation 2.110 can be rewritten as follows:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \kappa^x(e_k) \tag{2.113a}$$

$$\hat{d}_{k|k} = \hat{d}_{k|k-1} + \kappa^d(e_k) \tag{2.113b}$$

where the functions $\kappa^x : \mathbb{R}^p \to \mathbb{R}^n$ and $\kappa^d : \mathbb{R}^p \to \mathbb{R}^{n_d}$ are supposed to be continuous and to satisfy:

$$\kappa^x(0) = 0 \tag{2.114a}$$

$$\kappa^d(e_k) = 0 \Longleftrightarrow e_k = 0 \tag{2.114b}$$

Moreover, Equation 2.111 and Equation 2.113 form an asymptotically stable observer for the augmented system (2.104).

Note that condition (2.114) implies $n_d \geqslant p$. In general there is no advantage of choosing a disturbance model where $n_d > p$ as it provides more degrees of freedom than required and increases the complexity of observer, target calculator and regulator. Therefore, in the remainder it will be assumed that:

$$n_d = p \tag{2.115}$$

2.4.3 Target calculator and controller design

Obtained an estimate of the augmented state from the observer, the target calculator compute the values of state, input and output required at steady-state to ensure exact tracking of the controlled variable. Such values result from the solution at each generic sample time t_k of an optimization problem which can be formulated in general terms as follows:

$$\begin{aligned} \min_{x_k, u_k, y_k} \quad & \ell^s(y_k, u_k) \qquad (2.116) \\ \text{subject to: } \quad & x_k = F\left(x_k, u_k, \hat{d}_{k|k}\right) \\ & y_k = H\left(x_k, \hat{d}_{k|k}\right) \\ & r_y(y_k) = \bar{y}_c \\ & y_k \in \mathbb{Y} \\ & u_k \in \mathbb{U} \end{aligned}$$

where $\ell^s : \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}$ is the *steady-state cost function*.

Although (2.116) is an NLP problem and then may admits many local minima in dependence of input/output dimensions and system dynamics, in the remainder it will be assumed that (2.116) is feasible and that its solution $\left(x_k^s, u_k^s, y_k^s\right)$ is unique. Consider now a state sequence $X = \{x_0, x_1, \dots, x_N\}$ and an input sequence $U = \{u_0, u_1, \dots, u_{N-1}\}$ and let:

$$y_i = H\left(x_i, \hat{d}_k\right) \qquad \text{for } i = 0, \dots, N-1 \tag{2.117}$$

be the model output corresponding to a state x_i and a disturbance estimate $\hat{d}_{k|k}$.

As $\left(x_k^s, u_k^s, y_k^s\right)$ is available from the target calculation, it is possible to define the deviation variables:

$$\tilde{x}_i = x_i - x_k^s \qquad \text{for } i = 0, \dots, N \tag{2.118}$$

$$\tilde{u}_i = u_i - u_k^s \qquad \text{for } i = 0, \dots, N-1 \tag{2.119}$$

and formulate the following finite-horizon OCP:

$$\min_{X,U} \sum_{i=0}^{N-1} \ell^{ocp}(\tilde{x}_i, \tilde{u}_i) + V_f(\tilde{x}_N) \tag{2.120}$$

$$\begin{aligned} \text{subject to: } & x_0 = \hat{x}_{k|k} \\ & x_{i+1} = F\left(x_i, u_i, \hat{d}_{k|k}\right) \\ & y_i \in \mathbb{Y} \\ & u_i \in \mathbb{U} \\ & \tilde{x}_N \in \mathbb{X}_f \end{aligned}$$

in which $\ell^{ocp} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^+$ is a strictly positive definite convex function. $\mathbb{X}_f \subseteq \mathbb{X}$ and $V_f : \mathbb{R}^n \to \mathbb{R}^+$ are, respectively, a *terminal set* and a *terminal cost function*, properly designed in order to achieve closed loop stability.
Assuming that problem (2.120) is feasible, its solution is denoted by $\left(X_k^o, U_k^o\right)$ and the associated receding horizon implementation is given by:

$$u_k = U_0^o \tag{2.121}$$

Terminal set and terminal cost function design

The terminal set and the terminal cost function must be properly designed to ensure that the origin of the closed-loop system is an *asymptotically stable* equilibrium point.

Definition 15 (Stability of an equilibrium point)**.** Consider an autonomous nonlinear dynamical system $x_{k+1} = f(x_k)$ with an initial condition x_0. Suppose that $f : \mathcal{M} \to \mathbb{R}^n$ has an equilibrium at x^s so that $f\left(x^s\right) = 0$ then:

1. This equilibrium is said to be *stable*, if, for every constant $\alpha_1 > 0$, there exists a constant $\alpha_2 > 0$ such that, if $\|x_0 - x^s\| < \alpha_2$, then:

$$\lim_{k\to\infty} \|x_k - x^s\| < \alpha_1 \qquad \forall\, t_k > 0 \tag{2.122}$$

2. This equilibrium is said to be *asymptotically stable* if it is stable and the constant α_2 is such that if $\|x_0 - x^s\| < \alpha_2$, then:

$$\lim_{k\to\infty} \|x_k - x^s\| = 0 \tag{2.123}$$

3. This equilibrium is said to be *unstable* if it is not stable.

Note that asymptotic stability implies that a state trajectory starting close enough to an equilibrium point (within a distance α_2 from it) remains close to x^s (within a distance α_1) and eventually converge to it.
In order to outline conditions for the asymptotic stability of the closed-loop system there is the need to introduce the notion of *positively invariant set*.

Definition 16 (Positively invariant set)**.** Given a dynamical system $x_{k+1} = f(x_k)$ and the state trajectory $x(k, x_0)$ where x_0 is the initial condition, let $\mathcal{X} = \{x_k \in \mathbb{R}^n \,|\, \mathcal{F}\left(x_k\right) = 0\}$ where $\mathcal{F}$ ia a real valued function. The set $\mathcal{X}$ is said to be positively invariant if $x_0 \in \mathcal{X}$ implies that:

$$x(k, x_0) \in \mathcal{X} \qquad \forall t_k \geqslant 0 \tag{2.124}$$

Roughly speaking, this means that if the set $\mathcal{X}$ is positively invariant, then once a state trajectory of the system enters $\mathcal{X}$, it will never leave it again.
Assume now to know an auxiliary control law:

$$u = \kappa^a\left(x\right) \tag{2.125}$$

and a positively invariant set $\mathbb{X}_f \subset \mathbb{X}$ containing the origin such that, for the closed loop system:

$$x_{i+1} = f\left(x_i, \kappa^a\left(x_i\right)\right) \tag{2.126}$$

and for any $x_j \in \mathbb{X}_f$ the following conditions hold:

$$x_i \in \mathbb{X}_f \qquad \text{for } i \geqslant j \tag{2.127}$$

$$u_i = \kappa^a\left(x_i\right) \in \mathbb{U} \qquad \text{for } i \geqslant j \tag{2.128}$$

Suppose also that the stage cost $\ell^{ocp}\left(\cdot\right)$ in the objective function (2.120) is defined by:

$$\ell^{ocp}\left(x_i, u_i\right) = \|x_i\|^2_{Q_{reg}} + \|u_i\|^2_{R_{reg}} \tag{2.129}$$

where the tuning matrices $Q_{reg} \in \mathbb{R}^{n\times n}$ and $R_{reg} \in \mathbb{R}^{m\times m}$ are real-valued, symmetric and positive definite. Large values of Q_{reg} in comparison to R_{reg} are adopted when there is the intent to drive the state to the origin quickly even employing large control action. Conversely, penalizing the control action through large values of R_{reg} relative to Q_{reg} is the way to reduce the control action and slow down the rate at which the state approach the origin.

The deviation variables $\tilde{x}_i$ and $\tilde{u}_i$ appearing in Equation 2.120 have been replaced in Equation 2.129 with x_i and u_i for simplicity of notation, however, the handling which will follow is almost identical.

Under the assumptions made, the conditions for the asymptotic stability of the closed-loop system are established by the following theorem [61].

Theorem 4. *Let $\mathbb{X}^0_f$ be the set of states where a solution of the optimization problem (2.120) exists. If for any $x_i \in \mathbb{X}_f$ the condition:*

$$\Sigma_i = V_f\left(f\left(x_i, \kappa^a\left(x_i\right)\right)\right) - V_f\left(f\left(x_i\right)\right) + \|x_i\|^2_{Q_{reg}} + \|u_i\|^2_{R_{reg}} \leqslant 0 \tag{2.130}$$

is fulfilled and:

$$V_f\left(f\left(x_i\right)\right) \leqslant \mathcal{F}\left(\|x_i\|\right) \tag{2.131}$$

where $\mathcal{F}\left(\|x_i\|\right)$ is a class $\mathcal{K}$ function, then the origin of the closed-loop system with the receding horizon control law implicitly defined by (2.121) is an asymptotically stable equilibrium point with region of attraction $\mathbb{X}^0_f$.

A terminal set and a terminal cost function complying with the assumptions of the Theorem (4) can be obtained by linearization of the state transition equation (2.93a) at the origin:

$$f\left(x_i, u_i\right) \approx Ax_i + Bu_i + \phi\left(x_i, u_i\right) \tag{2.132}$$

where:

$$A = \left.\frac{\partial f}{\partial x_i}\right|_{\substack{x_i=0\\u_i=0}} \tag{2.133}$$

$$B = \left.\frac{\partial f}{\partial u_i}\right|_{\substack{x_i=0\\u_i=0}} \tag{2.134}$$

$$\lim_{\|(x_i,u_i)\|\to 0} \sup \frac{\|\phi\left(x_i, u_i\right)\|}{\|(x_i, u_i)\|} = 0 \tag{2.135}$$

The goal is to design a linear state feedback control law $u = -K^a x$ in a way such that the nominal nonlinear system (2.93) is asymptotically stabilized in a neighborhood of the

origin.
This neighborhood will be taken as the terminal set $\mathbb{X}_f$ in the optimization problem (2.120). The optimal control approach to this design is to define the performance index:

$$J^{lqr} = \frac{1}{2} \sum_{i=0}^{\infty} \|x_i\|^2_{Q_{reg}} + \|u_i\|^2_{R_{reg}} \tag{2.136}$$

and search for the control that minimizes such index. This problem is commonly denoted in linear systems theory as *Linear Quadratic Regulation (LQR) problem*.
As J^{lqr} is a monotonically increasing function with respect to i and is defined over an infinite interval, in order to ensure its convergence to a finite value there is the need to introduce the further assumption that the pair (A, B) is *completely controllable*.

Definition 17 (Complete controllability)**.** A system is completely controllable if it is possible to go from any initial state to any final state in a finite time.
For a linear system this is equivalent to the rank fullness of the so called *controllability matrix* $\mathcal{C} \in \mathbb{R}^{n \times (n \cdot m)}$, defined as:

$$\mathcal{C} = \begin{bmatrix} B & AB \cdots A^{n-1}B \end{bmatrix} \tag{2.137}$$

Thus, by dynamic programming arguments (see e.g. Rawlings and Mayne [55]), it is straightforward to show that, under the assumptions made, the optimal stabilizing control that minimizes (2.136) is given by:

$$u_i = -K_{LQ} x_i \tag{2.138}$$

in which:

$$K_{LQ} = \left(B^T \Pi B + R_{reg}\right)^{-1} B^T \Pi A \tag{2.139}$$

and Π is a positive semidefinite matrix, defined as:

$$\Pi = Q_{reg} + A^T \Pi A - A^T \Pi B \left(B^T \Pi B + R_{reg}\right)^{-1} B^T \Pi A \tag{2.140}$$

Equation 2.140 is referred to in literature as the *Discrete-time Algebraic Riccati Equation (DARE)*. Furthermore, the optimal value of the cost function from any point x_i is:

$$\ell^{lqr} = \frac{1}{2} \|x_i\|^2_{\Pi} \tag{2.141}$$

Substituting Equation 2.138 into Equation 2.132 yields:

$$f(x_i, u_i) \approx \left(A - BK_{LQ}\right) x_i + \phi\left(x_i, -K_{LQ} x_i\right) \tag{2.142}$$

where:

$$\lim_{\|x_i\| \to 0} \sup \frac{\|\phi\left(x_i, -K_{LQ} x_i\right)\|}{\|\left(x_i, -K_{LQ} x_i\right)\|} = 0 \tag{2.143}$$

Moreover, the objective function (2.136) becomes:

$$\begin{aligned} J^{lqr} &= \frac{1}{2} \sum_{i=0}^{\infty} \|x_i\|^2_{Q_{reg}} + \|u_i\|^2_{R_{reg}} \\ &= \frac{1}{2} \sum_{i=0}^{\infty} \|x_i\|^2_{Q_{reg}} + \|K_{LQ} x_i\|^2_{R_{reg}} \\ &= \frac{1}{2} \sum_{i=0}^{\infty} \|x_i\|^2_{Q^*_{reg}} \end{aligned} \tag{2.144}$$

In which:

$$Q_{reg}^{*} = Q_{reg} + K_{LQ}^{T} R_{reg} K_{LQ} \tag{2.145}$$

Equation 2.144 and Equation 2.142 form an LQR problem without the control, whose optimal cost in analogy with the previous case, results:

$$\ell^{lqr} = \frac{1}{2} \|x_i\|_{\Pi^{*}}^{2} \tag{2.146}$$

and Π^{*} is computed by the DARE:

$$\left(A - BK_{LQ}\right)^{T} \Pi^{*} \left(A - BK_{LQ}\right) = -\alpha_3 Q_{reg}^{*} \tag{2.147}$$

in which a generic constant $\alpha_3 > 1$ is added to take into account the effects of the linearization errors.

Let the terminal cost function $V_f\left(x_i\right)$ of the finite horizon controller (2.120) be defined by Equation 2.146.

In the neighborhood of the origin:

$$\begin{aligned}
\Sigma_i &= V_f\left(f\left(x_i, -K_{LQ} x_i\right)\right) - V_f\left(x_i\right) + \left(\|x_i\|_{Q_{reg}}^{2} + \|\kappa_a\left(x_i\right)\|_{R_{reg}}^{2}\right) \\
&= f\left(x_i, -K_{LQ} x_i\right)^{T} \Pi^{*} f\left(x_i, -K_{LQ} x_i\right) - x_i^{T} \Pi^{*} x_i + x_i^{T} Q_{reg} x_i + \\
&\qquad + x_i^{T} K_{LQ}^{T} R_{reg} K_{LQ} x_i \\
&= x_i^{T} \left(A - BK_{LQ}\right)^{T} \Pi^{*} \left(A - BK_{LQ}\right) x_i + 2 x_i^{T} \Pi^{*} \phi\left(x_i, -K_{LQ} x_i\right) - \\
&\qquad - x_i^{T} \Pi^{*} x_i + \phi^{T}\left(x_i, -K_{LQ} x_i\right) \Pi^{*} \phi\left(x_i, -K_{LQ} x_i\right) + x_i^{T} Q_{reg}^{*} x_i \\
&= x_i^{T} \left\{ \left(A - BK_{LQ}\right)^{T} \Pi^{*} \left(A - BK_{LQ}\right) - \Pi^{*} + Q_{reg}^{*} \right\} x_i + \\
&\qquad + 2 x_i^{T} \Pi^{*} \phi\left(x_i, -K_{LQ} x_i\right) + \phi^{T}\left(x_i, -K_{LQ} x_i\right) \Pi^{*} \phi\left(x_i, -K_{LQ} x_i\right) \\
&= x_i^{T} \left(1 - \alpha_3\right) Q_{reg}^{*} x_i + 2 x_i^{T} \Pi^{*} \phi\left(x_i, -K_{LQ} x_i\right) + \\
&\qquad + \phi^{T}\left(x_i, -K_{LQ} x_i\right) \Pi^{*} \phi\left(x_i, -K_{LQ} x_i\right)
\end{aligned} \tag{2.148}$$

Letting:

$$L_{\phi} = \sup \frac{\|\phi\left(x_i, -K_{LQ} x_i\right)\|}{\|x_i\|} \tag{2.149}$$

one has:

$$2 x_i^{T} \Pi^{*} \phi\left(x_i, -K_{LQ} x_i\right) \leqslant 2 \|\Pi^{*}\| L_{\phi} \|x_i\|^{2} \tag{2.150}$$

and:

$$\phi^{T}\left(x_i, -K_{LQ} x_i\right) \Pi^{*} \phi\left(x_i, -K_{LQ} x_i\right) \leqslant \|\Pi^{*}\| L_{\phi}^{2} \|x_i\|^{2} \tag{2.151}$$

Since:

$$\lim_{\|x_i\| \to 0} L_{\phi} = 0 \tag{2.152}$$

then $\Sigma_i \leqslant 0$ in a sufficiently small neighborhood of the origin, so that the decreasing condition (2.131) of Theorem 4 is satisfied.

In addition, the following choice of the terminal set:

$$\mathbb{X}_f = \left\{ x_i \,|\, x_i^{T} \Pi^{*} x_i \leqslant \alpha_4 \right\} \subset \mathbb{X} \tag{2.153}$$

for some suitably chosen constant α_4, guarantees that $\mathbb{X}_f$ is positively invariant for the LQ control law (2.138).

Summing up, the adoption of the stage cost (2.129), of the terminal set (2.153) and of the terminal cost function (2.146) in the optimal control problem (2.120) ensures the asymptotic stability of the closed-loop system origin.

2.4.4 Conditions for offset-free control

According to Pannocchia and Rawlings [47], a necessary condition for the offset-free control achievement is that the closed-loop system reaches an asymptotically stable equilibrium. Establishing general conditions under which this occurs is a very difficult task because it would involve state estimator, target optimizer and regulator together.
Therefore, a common approach for offset-free MPC (see e.g. Pannocchia et al. [46]) is to assume that an asymptotically stable equilibrium has been reached and then to demonstrate that offset-free performances are achieved at such equilibrium.
Indeed, under the assumption that the closed-loop system reaches a steady-state, the stability of the observer implies that the augmented state estimate also reaches a steady-state $\left(\hat{x}_\infty, \hat{d}_\infty\right)$. Thus, from Equation 2.111 and Equation 2.113 follows:

$$\hat{d}^*_\infty = \hat{d}_\infty \tag{2.154}$$

$$= \hat{d}^*_\infty + \kappa^d \left(y_\infty - \hat{y}^*_\infty\right) \tag{2.155}$$

where $\hat{d}^*_\infty$ and y_∞ are the predicted disturbance estimate and the process output at steady-state respectively and $\hat{y}^*_\infty = H\left(\hat{x}_\infty, \hat{d}_\infty\right)$.
It is straightforward to see that Equation 2.154 implies:

$$\kappa^d \left(y_\infty - \hat{y}^*_\infty\right) = 0 \tag{2.156}$$

and then from Equation 2.114 results:

$$y_\infty = \hat{y}^*_\infty = \hat{y}_\infty \tag{2.157}$$

Consider now the target calculation problem and denote its solution as $(x^s_\infty, u^s_\infty, y^s_\infty)$. Closed-loop stability of the equilibrium and positive definiteness of the cost function ℓ^s imply:

$$(\hat{x}_\infty, u_\infty) = (x^s_\infty, u^s_\infty) \tag{2.158}$$

where u_∞ is the first element of the optimal control profile and $\hat{x}_\infty$ is the corresponding first element of the optimal state sequence.
Equation 2.158 coupled with output constraints of (2.116) and the previous relations, imply offset-free control i.e.:

$$r_y\left(y_\infty\right) = r_y\left(\hat{y}_\infty\right) = r_y\left(y^s_\infty\right) = \bar{y}_c \tag{2.159}$$

All the previous reasonings can be summarized in the following theorem [46].

Theorem 5. *Assume that the target calculation problem* (2.116) *and the regulation problem are feasible at all times, and that the closed loop system reaches an equilibrium with input* u_∞ *and output* y_∞*. It follows that:*

$$r_y\left(y_\infty\right) = \bar{y}_c \tag{2.160}$$

2.5 REVIEW OF RELATED LITERATURE

The earliest industrial implementations of offset-free model predictive control as Dynamic Matrix Control (DMC) (Cutler and Ramaker [10]) or Identification and Command (IDCOM) (Richalet et al. [59]) date back to the '70. These control algorithms used a particular class of linear models called finite-impulse or step-response models to predict the future plant behavior and an output disturbance model to achieve offset-free performances.

However they were applicable only to square, open-loop, stable systems. State-space MPC formulations based on more general disturbance models and observer have been discussed more recently. In this context Pannocchia and Rawlings [47] derived general conditions and design criteria which apply to linear MPC for square and nonsquare, open-loop stable, integrating, and unstable systems and allow to achieve offset-free performances. Later, effective linear formulations based on disturbance models and observers have been matter of investigation (Pannocchia and Bemporad [44], Maeder et al. [31]) until Rajamani et al. [54] presented an important result on the disturbance models equivalence. Apparently different linear offset-free approaches are the *velocity form approach* [49] and the *state disturbance observer approach* [67]. In the first the input and (possibly) the state are replaced by their rate-of-change and the offset-free control is achieved without the need of defining and estimating disturbances while the other method is characterized by fact that it does not require state augmentation. However, newly Pannocchia [42] has demonstrated that these two supposedly alternative approaches are actually particular cases of the disturbance model formulation.

The earliest state-space offset-free NMPC algorithm is probably the one proposed by Meadows and Rawlings [34] where an output disturbance, computed at each sampling instant as the difference between the measured plant output and the model prediction, is added into the objective function for the entire predictive horizon. Later, Srinivasarao et al. [66] proposed a formulation that integrates both the state and output disturbances from the EKF and has the advantage to be applied to open-loop unstable systems. In order to incorporate constraints handling abilities in the state estimator unit Huang et al. [23] proposed an offset-free NMPC based on nonlinear moving horizon estimation (MHE). When there are no information about the plant-model mismatch or about the disturbances acting on the system this framework implements a disturbance model which affects both state and output dynamics. The method has been successfully implemented on a large scale air separation unit, where reveals himself to be very effective in removing offset in the controlled variables at steady-state and, when the set-point is not feasible, in minimizing the difference between the reference signal and the steady-state response. The major disadvantage of this framework is that it gives no guarantees on the observability of the augmented nonlinear system, therefore, it may be unimplementable in some practical applications. Few years later, Morari and Maeder [37] have extended the concepts of linear offset-free MPC to nonlinear MPC. The work presents a generalization of the concepts in Maeder et al. [31], and Pannocchia and Rawlings [47] and at the same time offers a much simpler exposition of the ideas behind offset-free linear MPC. Recently, those concepts have been reviewed by Pannocchia et al. [46] who proposed an NMPC offset-free framework which implements a linear disturbance model affecting the state evolution whose dynamics is described by a properly chosen matrix.

3 PROPOSED METHOD

In this chapter a procedure aimed at designing the most suitable disturbance model and associated observer to be implemented in an NMPC framework is developed. The chapter is organized as follows. In section 3.1 the equations describing the plant behavior and the nominal process model are introduced together with the basic assumptions on the disturbance dynamics and the constraints on input, output and state. Then, the next two sections are devoted to the description of the units constituting the offset-free NMPC framework adopted in this work and of the design choices made. In particular, section 3.2 concerns state augmentation and estimator with a deepening on two disturbance model employed in common practice, while section 3.3 regards target calculator and regulator. Eventually, section 3.4 provides a detailed description of the actual procedure highlighting its peculiar theoretical and computational aspects and showing all the facilities provided by this method in determining the optimal nonlinear disturbance model to be implemented in the offset-free NMPC framework designed.

3.1 PLANT, NOMINAL MODEL AND CONSTRAINTS

Assume that the plant to be controlled is described by a set of nonlinear, time-invariant and discrete equations like (2.92) and that its output is measured at each sample time t_k. The following discrete-time nonlinear system is taken as a representation of the plant dynamics:

$$x_{k+1} = f\left(x_k, u_k, \theta_0\right) \tag{3.1a}$$

$$y_k = h\left(x_k, \theta_0\right) \tag{3.1b}$$

where $\theta_0 \in \mathbb{R}^{n_\theta}$ denotes the nominal model parameters. The functions $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{n_\theta} \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R}^{n_\theta} \to \mathbb{R}^p$ are assumed to be continuous.
Disturbances at any sample time t_k are defined as:

$$w_k^* = f^\star\left(x_k^\star, u_k, w_k^\star\right) - f\left(x_k, u_k, \theta_0\right) \tag{3.2}$$

$$v_k^* = h^\star\left(x_k^\star, v_k^\star\right) - h\left(x_k, \theta_0\right) \tag{3.3}$$

and are assumed to be asymptotically constant and bounded in compact sets: $w_k^* \in \mathbb{W}$, $v_k^* \in \mathbb{V}$.
Input and output are required to satisfy constraints of type (2.96) and (2.97) respectively and therefore the related sets of admissible inputs and outputs are defined as in (2.98) and (2.99).
The state constraints set results instead:

$$\mathbb{X} := \left\{x_k \in \mathbb{R}^n \,|\, g_y\left(h\left(x_k, \theta_0\right)\right) \leqslant 0\right\} \tag{3.4}$$

3.2 AUGMENTED MODEL AND STATE ESTIMATOR DESIGN

In order to achieve offset-free control in presence of unmodeled, nonzero mean disturbances and/or plant-model mismatch the model (3.1), used by the controller, is augmented with an integrated disturbance term:

$$\begin{aligned} x_{k+1} &= F\left(x_k, u_k, \theta_0 + L\, d_k\right) && (3.5a)\\ d_{k+1} &= d_k && (3.5b)\\ y_k &= H\left(x_k, \theta_0 + L\, d_k\right) && (3.5c) \end{aligned}$$

where L is an $\mathbb{R}^{n_\theta \times n_d}$ matrix whose columns are distinct standard unit vectors; the form of L determines to which process parameters the integrated terms are added. The choice of L is the key-issue of this thesis and, therefore, it is widely discussed in the sections ahead. The functions $F : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{n_\theta} \to \mathbb{R}^n$ and $H : \mathbb{R}^p \times \mathbb{R}^{n_\theta} \to \mathbb{R}^p$ are assumed to be continuous and consistent with the nominal model (3.1), i.e. for all $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ there holds:

$$\begin{aligned} F\left(x_k, u_k, \theta_0\right) &= f\left(x_k, u_k, \theta_0\right) && (3.6a)\\ H\left(x_k, \theta_0\right) &= h\left(x_k, \theta_0\right) && (3.6b) \end{aligned}$$

The evolution of the augmented state $\zeta_k = \left[x_k^T\ d_k^T\right]^T$ can be rewritten as:

$$\begin{aligned} \zeta_{k+1} &= F_a\left(\zeta_k, u_k, \theta_0 + L_a \zeta_k\right) && (3.7a)\\ y_k &= H_a\left(\zeta_k, \theta_0 + L_a \zeta_k\right) && (3.7b) \end{aligned}$$

in which $L_a \in \mathbb{R}^{n_\theta \times (n+n_d)}$ is a matrix of the form $[0\ L]$. Even functions $F_a : \mathbb{R}^{n+n_d} \times \mathbb{R}^m \times \mathbb{R}^{n_\theta} \to \mathbb{R}^{n+n_d}$ and $H_a : \mathbb{R}^{n+n_d} \times \mathbb{R}^{n_\theta} \to \mathbb{R}^p$ are assumed to be continuous and consistent with (3.1).

An extended Kalman filter is employed as a state estimator for (3.7) to estimate the augmented state from the plant measurements.

The filter uses the following representation of the augmented system dynamics:

$$\begin{aligned} \zeta_{k+1} &= F_a\left(\zeta_k, u_k, \theta_0 + L_a \zeta_k, w_k\right) && (3.8a)\\ y_k &= H_a\left(\zeta_k, \theta_0 + L_a \zeta_k, v_k\right) && (3.8b) \end{aligned}$$

in which w_k and v_k are white, normal, zero-mean and uncorrelated noises with covariance matrices $Q_k \in \mathbb{R}^{(n+n_d)\times(n+n_d)}$ and $R_k \in \mathbb{R}^{p \times p}$ respectively.

Thus, the predicted estimates are defined as:

$$\begin{aligned} \hat{\zeta}_{k|k-1} &= F_a\left(\hat{\zeta}_{k-1|k-1}, u_{k-1}, \theta_0 + L_a \hat{\zeta}_{k-1|k-1}\right) && (3.9a)\\ \hat{y}_{k|k-1} &= H_a\left(\hat{\zeta}_{k|k-1}, \theta_0 + L_a \hat{\zeta}_{k-1|k-1}\right) && (3.9b) \end{aligned}$$

and the filtering equation is linear in the form:

$$\hat{\zeta}_{k|k} = \hat{\zeta}_{k|k-1} + K_k\left(y_k - \hat{y}_{k|k-1}\right) \qquad (3.10)$$

where $K_k \in \mathbb{R}^{(n+n_d)\times p}$ is the Kalman gain matrix introduced in Equation 2.42.

Equation 3.9 and Equation 3.10 are assumed to form an asymptotically stable observer for the augmented system (3.5).

Equation 3.10 can also be written as:

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k^x\left(y_k - \hat{y}_{k|k-1}\right) && (3.11a)\\ \hat{d}_{k|k} &= \hat{d}_{k|k-1} + K_k^d\left(y_k - \hat{y}_{k|k-1}\right) && (3.11b) \end{aligned}$$

where $K_k^x \in \mathbb{R}^{n \times p}$ and $K_k^d \in \mathbb{R}^{n_d \times p}$ are related to K_k by the following expression:

$$K_k = \begin{bmatrix} K_k^x \\ K_k^d \end{bmatrix} \tag{3.12}$$

From (3.11) is straightforward to see that taking a number of disturbances equal to the number of outputs, i.e.:

$$n_d = p \tag{3.13}$$

and the assumption that K^d is full rank are sufficient to ensure that not only condition (2.114a) is obviously satisfied but also condition (2.114b). Indeed, under the assumptions made K_k^d is a full rank square matrix and hence according to the rank–nullity theorem only the zero output prediction error vector lies in its nullspace.

3.3 TARGET CALCULATOR AND CONTROLLER DESIGN

Given the current augmented state estimate from the EKF, the target calculator determine the equilibrium targets that ensure exact tracking of the controlled variables by solving the following nonlinear program:

$$\min_{x_k, u_k, y_k} \ell^s \left(y_k, u_k\right) \tag{3.14}$$

$$\begin{aligned} \text{subject to: } x_k &= F\left(x_k, u_k, \theta_0 + L\,\hat{d}_{k|k}\right) \\ y_k &= H\left(x_k, \theta_0 + L\,\hat{d}_{k|k}\right) \\ r_y\left(y_k\right) &= \bar{y}_c \\ y_k &\in \mathbb{Y} \\ u_k &\in \mathbb{U} \end{aligned}$$

(3.14) is assumed to be feasible at all times and its (unique) solution is denoted as $\left(x_k^s, u_k^s, y_k^s\right)$. Let $U = \{u_0, u_1 \ldots, u_{N-1}\}$, $X = \{x_0, x_1 \ldots, x_N\}$ and $Y = \{y_0, y_1 \ldots, y_N\}$ be, respectively, an input profile, a state trajectory and an output sequence.
Denote as Δu_i the difference between a manipulated variable u_i and its former.

$$\Delta u_i = u_i - u_{i-1} \qquad \text{for } i = 0, \ldots, N-1 \tag{3.15}$$

Furthermore, let:

$$\tilde{y}_i = y_i - \bar{y}_c \qquad \text{for } i = 0, \ldots, N \tag{3.16}$$

$$\tilde{r}_i = \begin{bmatrix} x_i - x_k^s \\ u_{i-1} - u_k^s \end{bmatrix} \qquad \text{for } i = 0, \ldots, N \tag{3.17}$$

The optimal control profile is chosen by solving the following finite-horizon nonlinear optimization problem:

$$\min_{X,U} \sum_{i=0}^{N-1} \ell^{ocp}\left(\tilde{y}_i, \Delta u_i\right) + V_f\left(\tilde{r}_N\right) \tag{3.18}$$

$$
\begin{aligned}
\text{subject to: } x_0 &= \hat{x}_{k|k} \\
x_{i+1} &= F\left(x_i, u_i, \theta_0 + L\,\hat{d}_{k|k}\right) \\
y_i &= H\left(x_i, \theta_0 + L\,\hat{d}_{k|k}\right) \\
y_i &\in \mathbb{Y} \\
u_i &\in \mathbb{U} \\
\tilde{\Gamma}_N &\in \mathbb{C}_f
\end{aligned}
$$

in which $V_f : \mathbb{R}^{n+m} \to \mathbb{R}^+$ and $\mathbb{C}_f \subseteq \mathbb{X} \cup \mathbb{U}$ are properly designed terminal set and terminal cost function, respectively, and $\ell^{ocp} : \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}^+$ is still a strictly positive definite convex function.

Assuming that problem (3.18) is feasible, its solution is denoted by $\left(X_k^o, U_k^o\right)$ and the associated receding horizon implementation is given by:

$$u_k = U_0^o \tag{3.19}$$

Terminal set and terminal cost function design

The stage cost ℓ^{ocp} of the objective function (3.18) is taken as follows:

$$\ell^{ocp}(y_i, \Delta u_i) = \|y_i\|^2_{Q_{reg}} + \|\Delta u_i\|^2_{R_{reg}} \tag{3.20}$$

where $Q_{req} \in \mathbb{R}^{p \times p}$ and $R_{reg} \in \mathbb{R}^{m \times m}$ and the deviation variables have been replaced by y_i and Δu_i in order to avoid heavy notation. In this formulation Q_{reg} is the matrix containing the weights on the squared deviations of the output from the origin, while R_{reg} is the matrix of weights on the square of the inputs change called *move suppression factors*. The larger R_{reg} in comparison to Q_{reg}, the more the optimizer tends to minimize the associated input changes and the smoother the control action is. Conversely, employing large values of Q_{reg} with respect to R_{reg} the output are driven to the origin quickly even at the expense of nervous control action.

Consider the following linearization of the state transition equation at the origin:

$$
\begin{aligned}
f\left(x_i, u_i\right) &\approx A\delta x_i + B\delta u_{i-1} + B\Delta u_i + \phi\left(\delta x_i, \delta u_i\right) &(3.21a)\\
\delta u_i &= \delta u_{i-1} + \Delta u_i &(3.21b)
\end{aligned}
$$

from which:

$$
\begin{aligned}
\begin{bmatrix} f\left(x_i, u_i\right) \\ \delta u_{i-1} \end{bmatrix} &\approx \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \delta\Gamma_i + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u_i + \begin{bmatrix} \phi\left(\delta x_i, \delta u_i\right) \\ 0 \end{bmatrix} \\
&\approx A_a \Gamma_i + B_a \Delta u_i + \phi_a\left(\delta x_i, \delta u_i\right)
\end{aligned} \tag{3.22}
$$

Thus, in analogy to the general case treated in subsection 2.4.3 the LQR problem objective function (2.136) has the form:

$$J^{lqr} = \frac{1}{2} \sum_{i=0}^{\infty} \|\Gamma_i\|^2_{Q^{\Gamma}_{reg}} + \|\Delta u_i\|^2_{R_{reg}} \tag{3.23}$$

where:

$$Q^{\Gamma}_{reg} = \begin{bmatrix} Q_{reg} & 0 \\ 0 & 0 \end{bmatrix} \tag{3.24}$$

Proceeding in an analogous manner to what was done in subsection 2.4.3 is straightforward to demonstrate that the choices:

$$V_f(\Gamma_i) = \frac{1}{2}\|\Gamma_i\|^2_{\Pi^*} \tag{3.25}$$

$$\mathbb{C}_f = \left\{\Gamma_i \,|\, \Gamma_i \Pi^* \Gamma_i^T \leqslant \alpha_5\right\} \subset \mathbb{X} \cup \mathbb{U} \tag{3.26}$$

comply with the assumption of Theorem 4 and hence ensure the asymptotic stability of the closed-loop system origin.

3.4 DESIGN OF THE DISTURBANCE MODEL

Although output and (mostly) input disturbance models, discussed in subsection 2.4.1, often perform adequately, process knowledge could be used to augment the nominal model with integrating terms more effectively. For instance, if disturbances are known to enter a process through specific parameters, it seems reasonable to add the integrators to these parameters. However, when the number of disturbances which are known to enter the process through specific parameters is greater than the number of outputs there is the problem of establishing on which parameters to put the integrated terms.

In this section a systematic and structured procedure to overcome this issue and implement a very effective nonlinear disturbance model is proposed.

3.4.1 Observability limitation

In order to update the augmented state estimate with Equation 3.10 and hence achieve offset-free control performances the augmented system (3.5) has to be observable at least in a neighborhood of the steady-states (x^s, u^s) where the plant has to be controlled.

Thus, the first step of the disturbance design procedure is to determine which ones of all the candidate augmented systems of type (3.5) are observable.

Different approaches must be adopted in dependence of the operating point stability

Introduced the preliminary definitions above it is now possible to proceed with the detailed description of the first phase of the disturbance model designon procedure.

Stable operating point

According to the following definition, suppose that (x^s, u^s) is an *exponentially stable* steady-state.

Definition 18 (Exponentially stable equilibrium point)**.** Let x^s be an equilibrium point of the continuous-time nonlinear system (2.49) with constant input u^s.

x^s is called an *exponentially stable equilibrium point* of (2.49) if there exist some constants α_6, α_7 and $\alpha_8 > 0$ such that the solution x satisfies

$$|x| \leqslant \alpha_6 |x_0| e^{-\alpha_8 (t-t_0)} \quad \forall |x_0| < \alpha_7 \text{ and } t \geqslant t_0 \geqslant 0 \tag{3.27}$$

α_8 is called the *rate of convergence*.

In this case the observability of the augmented system (3.5) is verified by checking the observability covariance matrix rank condition (3).

Note that in order to obtain reliable results the sizes of the perturbations for each direction c_m must be chosen in a way such that:

$$x(0) = x^s + c_m T_l e_i \tag{3.28}$$

remains in the region of attraction of x^s during the test.

The observability covariance matrix of the augmented system (3.5) can be decomposed into:

$$W_O = \begin{bmatrix} W_{O_{xx}} & W_{O_{xd}} \\ W_{O_{dx}} & W_{O_{dd}} \end{bmatrix} \tag{3.29}$$

where $W_{O_{xx}}$ is the observability covariance matrix of the nonaugmented system (3.1), $W_{O_{dd}}$ represents the covariances of the outputs with respect to disturbances and $W_{O_{xd}}$ and $W_{O_{dx}}$, which are one the transposed of the other, represent the covariance of the outputs resulting from changes in the state variables and disturbances.

From Equation 3.29 it is straightforward to see that different disturbance models, each one corresponding to a specific choice of L, will lead to a distinct W_O. Thus, measures of the degree of observability can be used to rank all candidate augmented systems from the most observable in the region of interest to the least one. As in control applications it is important that a minimum degree of observability exists even in the worst system direction, measure (2.83) is used.

In general it is impossible to establish a limit value for $\lambda_{min}(W_O)$ under which an augmented model can be considered practically unobservable. Thus, defined the following sequences of asymptotically constant disturbances:

$$W^{sim} = \{w_0^{sim}, w_1^{sim}, \ldots, w_{N_{sim}}^{sim}\} \tag{3.30}$$

$$V^{sim} = \{v_0^{sim}, v_1^{sim}, \ldots, v_{N_{sim}}^{sim}\} \tag{3.31}$$

where $w_i^{sim} \in \mathbb{W}$ and $v_i^{sim} \in \mathbb{V}$, the augmented systems behavior in presence of disturbances is simulated for a time interval of length $t_{N_{sim}}$ and the candidate disturbance models which result unobservable are discarded. The augmented models must be tested in descending order of degree of observability. In this way if during the simulations a candidate augmented system results unobservable then all the other characterized by a lower degree of observability should not be analyzed and can be discarded.

The procedure is summarized in Figure 13.

Under the assumptions made each candidate augmented model that meets the observability requirement allows the regulator to remove offset in the controlled variables if implemented in an NMPC framework. Therefore, if high performances are not required then one may think to skip the second phase of the procedure and take directly the most observable augmented model.

In this case it is possible to automate the disturbance model design by formulating and solving the following optimization problem:

$$\min_{a} \min_{i} \lambda_{min}(W_{0,i}) \tag{3.32}$$

$$\begin{aligned} \text{subject to: } x_{k+1} &= F\left(x_k, u_k, \theta_0 + a^T d_k\right) \\ y_k &= H\left(x_k, \theta_0 + a^T d_k\right) \\ \sum_{i=0}^{n_a} a_i &= p \\ a_i &\in \{0,1\} \quad \text{for } i = 1, \ldots, n_a \end{aligned}$$

where $a \in \mathbb{R}^{n_a}$ is a constant vector and $W_{0,i}$ is the observability covariance matrix relative to a certain distance from the operating point.

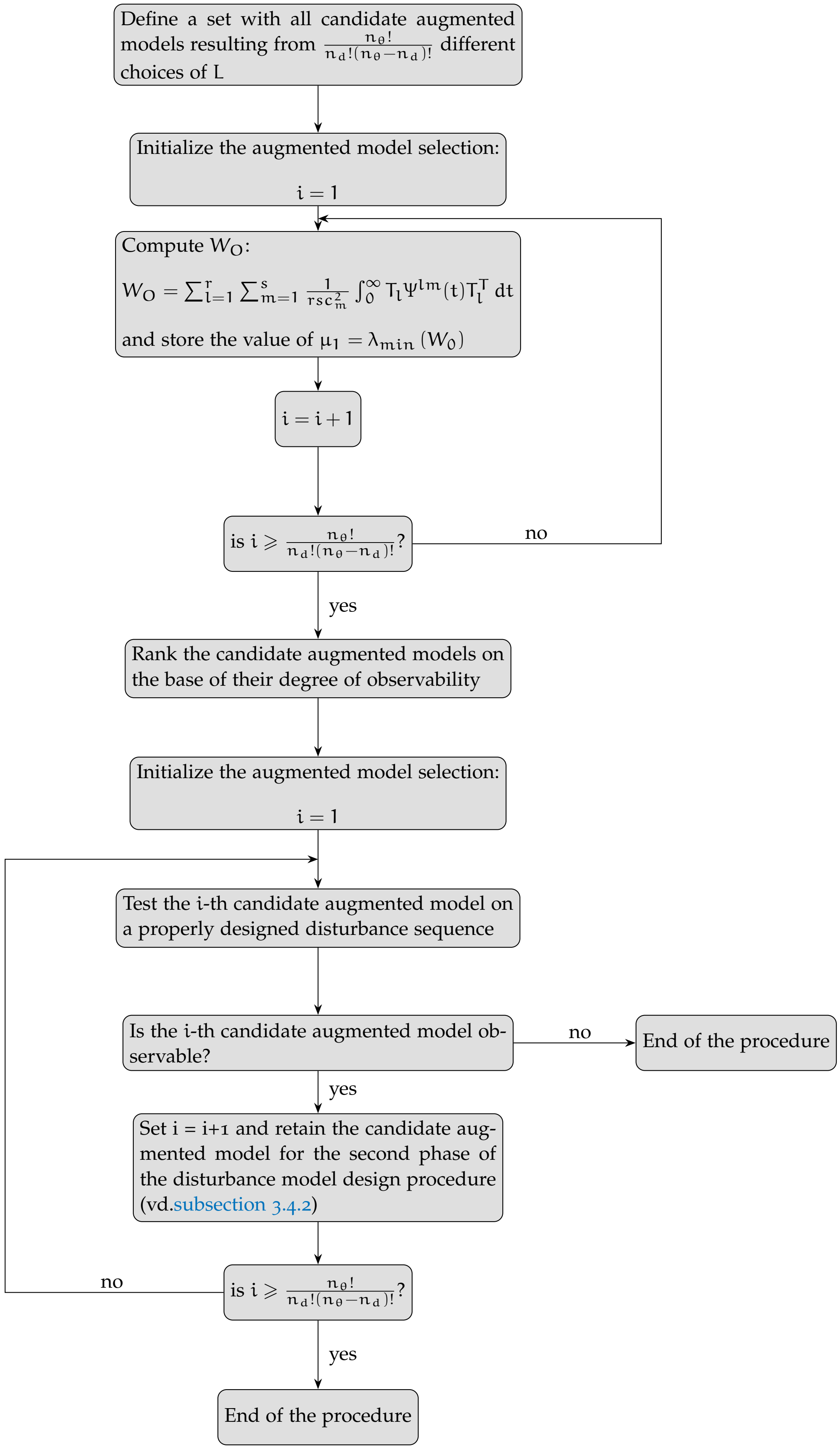

Define a set with all candidate augmented models resulting from $\frac{n_\theta!}{n_d!(n_\theta - n_d)!}$ different choices of L
Initialize the augmented model selection: $i = 1$
Compute W_O:
$W_O = \sum_{l=1}^{r} \sum_{m=1}^{s} \frac{1}{rsc_m^2} \int_0^\infty T_l \Psi^{lm}(t) T_l^T \, dt$
and store the value of $\mu_1 = \lambda_{min}(W_0)$
$i = i + 1$
is $i \geqslant \frac{n_\theta!}{n_d!(n_\theta - n_d)!}$?
no
yes
Rank the candidate augmented models on the base of their degree of observability
Initialize the augmented model selection: $i = 1$
Test the i-th candidate augmented model on a properly designed disturbance sequence
Is the i-th candidate augmented model observable?
no
End of the procedure
yes
Set i = i+1 and retain the candidate augmented model for the second phase of the disturbance model design procedure (vd.subsection 3.4.2)
no
is $i \geqslant \frac{n_\theta!}{n_d!(n_\theta - n_d)!}$?
yes
End of the procedure

Figure 13: Observable augmented models design procedure for systems running in the region of attraction of stable operating points.

Unstable operating point

The observability covariance matrix cannot be computed for unstable operating points, therefore, in dependence of the structure of the system there is the need of employing the Kalman rank test (2) on the linearized model or the observability rank test (2.60). According to this criteria, they will be retained only augmented models for which:

$$\mathrm{rank}\,\Theta\,(\zeta) = n + n_d \tag{3.33}$$

or

$$\mathrm{rank}\,\mathcal{O} = n + n_d \tag{3.34}$$

while the other are discarded.
In this case it is not possible to rank the retained augmented models on the base of their degree of observability. Therefore, the augmented models behavior must be simulated on the disturbance sequences W^*_{sim} and V^*_{sim} in order to establish which of them are observable in the region of interest and discard the unobservables.
The whole procedure is summarized in Figure 14.

3.4.2 Design of the optimal disturbance model

As stated above, each observable augmented model in principle is able to ensure offset-free control if implemented in an NMPC framework. However, depending on the disturbance model which is used the robust performances of the offset-free controller will be quite different as the effects of adding integrating disturbances are specific to the application in the nonlinear context.
The aim of this second part of the disturbance model design procedure is to determine which one of the observable augmented models resulting from the previous test performs better in closed-loop.
In order to accomplish this end there is the need to compare the closed-loop performances of each NMPC algorithm linked to each candidate disturbance model. However, as the regulator performances are highly dependent from the state estimator ones, the extended Kalman filter employed for each augmented model must be properly tuned so that the performances of each regulator are the better achievable.
As frequently the measurement error statistics remain constant during time, R_k is assumed to be time-invariant, i.e.:

$$R_k = R \tag{3.35}$$

and the value of R is evaluated by taking some off-line sample measurements and computing the measurement error variance.
Q_k cannot be determined in an analogous deterministic way, therefore, to simplify the tuning operations it assumed to be time-invariant like R:

$$Q_k = Q \tag{3.36}$$

and diagonal.
Note that the latter assumption is equivalent to consider that the process noises acting on the augmented state are independent from each other.
Thus, taking an optimization based approach, the extended Kalman filter tuning problem can be formalized as follows:

$$\min_{Q}\; J^{tun} \tag{3.37}$$

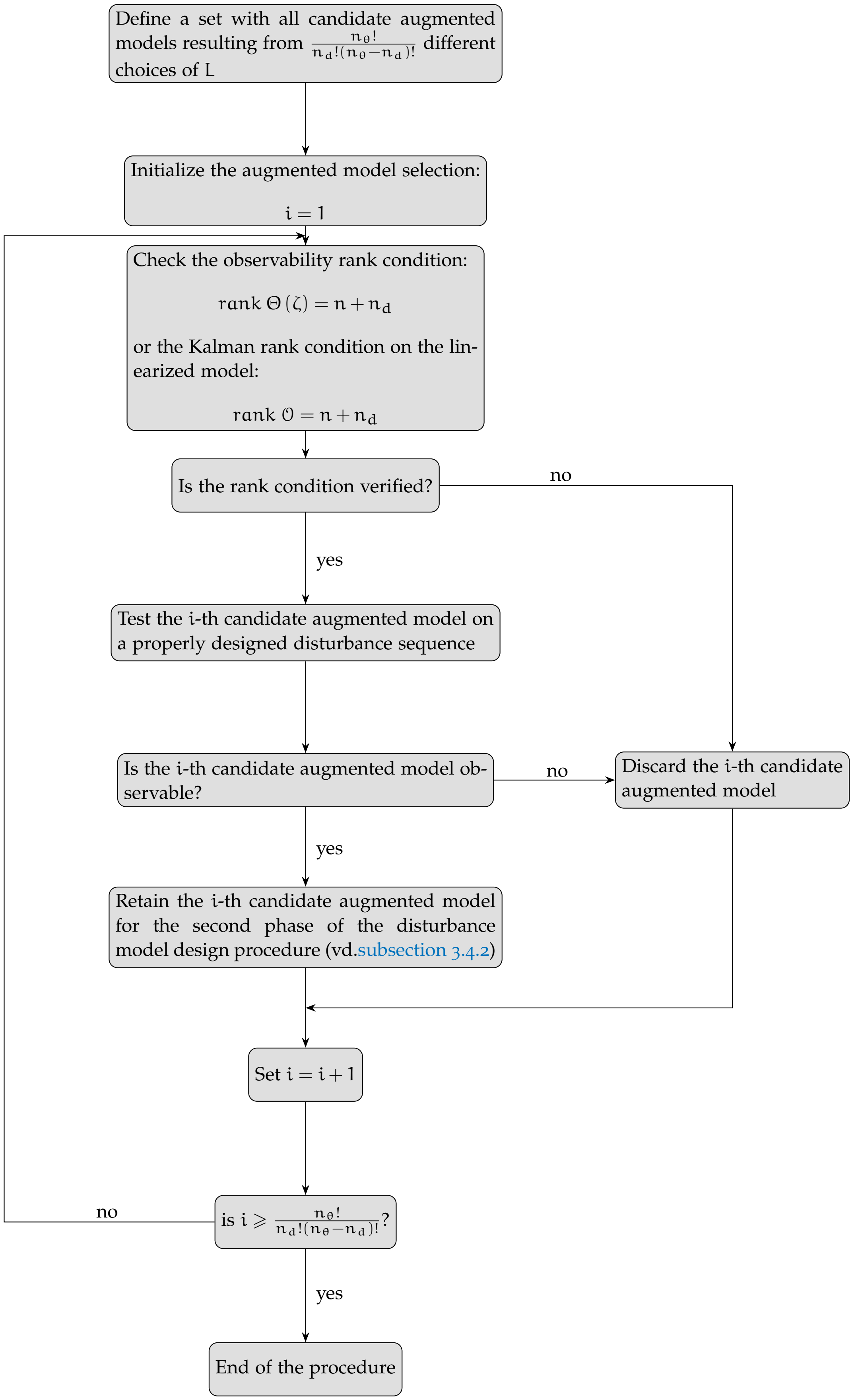

Define a set with all candidate augmented models resulting from $\frac{n_\theta!}{n_d!(n_\theta - n_d)!}$ different choices of L
Initialize the augmented model selection: $i = 1$
Check the observability rank condition: $\operatorname{rank} \Theta(\zeta) = n + n_d$ or the Kalman rank condition on the linearized model: $\operatorname{rank} \mathcal{O} = n + n_d$
Is the rank condition verified?
no
yes
Test the i-th candidate augmented model on a properly designed disturbance sequence
Is the i-th candidate augmented model observable?
no
Discard the i-th candidate augmented model
yes
Retain the i-th candidate augmented model for the second phase of the disturbance model design procedure (vd.subsection 3.4.2)
Set $i = i + 1$
no
is $i \geqslant \frac{n_\theta!}{n_d!(n_\theta - n_d)!}$?
yes
End of the procedure

Figure 14: Observable augmented models design procedure for systems operating in a neighborhood of unstable steady-states.

$$\text{subject to: } Q_{min} \leqslant Q \leqslant Q_{max}$$

where Q_{min} and Q_{max} are nonnegative diagonal matrices whose diagonal elements represent respectively the lower and the upper bounds of the diagonal elements of Q. These matrices may results from the process knowledge, from information available in literature or from tests previously conducted on the filter.

In filtering literature (see e.g. Powell [51] or Ting et al. [69]) it is common to develop performance criteria as functions of the estimation error. However, when these objective functions are adopted to tune the EKF of the NMPC framework in presence of disturbances and noises, several optimization algorithms do not converge or progress slightly from the initial condition. In particular, gradient-based methods fail because of the difficulty in analytically calculating the gradient, approximating the cost function with a well-behaved function, and the instability of numerical differentiation of noisy functions [4]. Even downhill simplex [51] has reveal itself unsuitable for this class of problems.

In order to overcome this issue, defined:

$$\Delta u_k = u_k - u_{k-1} \tag{3.38}$$

and the *tracking error*:

$$\tilde{y}_k = y_k - \bar{y}_c \tag{3.39}$$

the following solution is proposed:

$$\min_{Q} \sum_{k=0}^{N_{sim}} \ell^{tun}\left(\tilde{y}_k, \Delta u_k\right) \tag{3.40}$$

$$\text{subject to: } Q_{min} \leqslant Q \leqslant Q_{max}$$

where $\ell^{tun} : \mathbb{R}^{(n+n_d)\times(n+n_d)} \times \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}^+$ is a strictly positive definite function.

When the disturbance sequences are applied to the system for which the state estimator has to be tuned, the estimation error returns in a neighborhood of zero in a smaller amount of time respect to the tracking error. Because of this, small improvements in the state estimator performances are more visible on the tracking error than on the estimation error and hence (3.40) is easiest to solve for numerical algorithms than optimization problems which involve estimation error based objective functions.

Furthermore, from (3.18) it is straightforward to see that the smaller the estimation errors, the better the control actions and the smaller the value of ℓ_t in the minimum. For this reason, solve the optimization problem (3.40) indirectly leads to a minimization of a certain function of the estimation error.

In conclusion, note that $\ell^{tun}(\cdot)$ depends solely on measurable terms and hence the filter performances can be easily monitored by operators.

Computational method

The objective function of the nonlinear program (3.40) is subject to noise and is characterized by the presence of multiple local minima. For these reasons, a highly accurate solution is not necessary, and may be impossible to compute also considered the fact that there are no prior information available to initialize the optimization close to the solution. Thus, all that is desired is an improvement in the objective function value rather than a full optimization.

Global optimizers like genetic algorithms, differential evolution, and simulated annealing are very effective methods but they cannot be adopted because of their high computational cost. Therefore, a *brute force* approach is used.

A brute-force approach is an algorithm that evaluates the objective function at each point of a multidimensional grid of points and returns as solution the gridpoint at which the lowest value of the objective function occurs.

Clearly, testing more points increases the computational cost of the method, however, at the same time it raises the odds of the algorithm of getting smaller objective function values. Note that this rule of thumb is not guaranteed because of the stochastic nature of the algorithm. Indeed, it could easily be the case that in one short run the algorithm fortuitously finds a point very close to the minimum achievable while in another much longer run the algorithm is a less fortunate and does not get as close.

Although simplistic, the brute-force approach has the advantage of being able to work with any objective function, even one that has a complex and irregular shape, multiple local optima, and discontinuities. Furthermore, by trying points on the grid and always keeping the optimal one, it is likely to get close to the global optimum and certain not to get stuck in a local optimum [20]. However, when the algorithm gets a point close to the global optimum, it does not improve on it except by possibly testing an even closer point.

Therefore, in order to give a significant improvement to the optimum found by the brute-force search and achieve a better result the algorithm is followed by a certain number of iterations of the *Nelder-Mead method* (see subsection A.3.3).

Since such optimizer is designed to solve unconstrained optimization problems, the nonlinear program (3.40) must be reformulated as follows:

$$\min_{\sqrt{Q}} \sum_{k=0}^{N_{sim}} \ell^{tun}\left(\tilde{y}_k, \Delta u_k\right) \tag{3.41}$$

The Nelder-Mead algorithm does not require any derivative information and typically performs one or two function evaluations per iteration. For these reasons it reveals itself particularly suitable for the resolution of problems like (3.41) characterized by non-smooth and noisy objective functions and very time-consuming function evaluations.

4 CASE STUDY: NON-ISOTHERMAL CSTR

This chapter illustrates the presented technique by determining the optimal disturbance model to be implemented in the offset-free NMPC framework adopted, for the control of an highly nonlinear plant. The control performances obtained are then compared to the ones of other frameworks available in literature. The detailed outline of the chapter is as follows. In section 4.1 a description of the nonlinear system used for the case study is given highlighting the peculiarities of its dynamics and defining which are the control tasks to be achieved. In section 4.2 and section 4.3 concern the state estimator, target calculator and controller designs according to the rules given in the previous chapter. Section 4.4 is devoted to the application of the proposed method to the motivating example; it leads to the synthesis of an NMPC offset-free framework whose control performances are then evaluated and discussed in section 4.5.

4.1 PLANT MODEL AND CONTROL TASK

A non-isothermal continuous stirred tank reactor as shown in Figure 15 is considered. This reactor was described in Pannocchia and Rawlings [47] and is adopted unchanged in this work. An irreversible exothermic first-order reaction A → B occurs in the liquid phase and the temperature is regulated with external cooling.

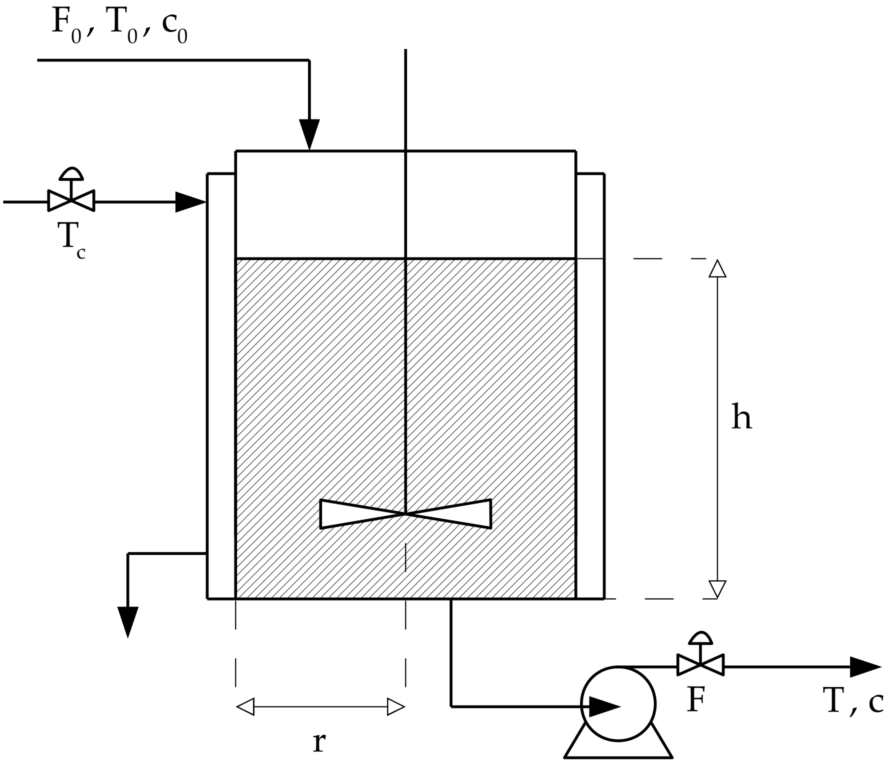


Figure 15: Schematic of the well-stirred reactor.

Mass and energy balances lead to the following nonlinear state-space model:

$$\frac{dc}{dt} = \frac{(F_0 c_0 - Fc)}{\pi r^2 h} - k_0 \exp\left(-\frac{E}{RT}\right) c \tag{4.1a}$$

$$\frac{dT}{dt} = \frac{F_0 (T_0 - T)}{\pi r^2 h} + \frac{(-\Delta H)}{\rho C_p} k_0 \exp\left(-\frac{E}{RT}\right) c + \frac{2U_0}{r\rho C_p}(T_c - T) \tag{4.1b}$$

$$\frac{dh}{dt} = \frac{F_0 - F}{\pi r^2} \tag{4.1c}$$

The model parameters in nominal conditions are reported in Table 2.
The controlled variables are the reactant concentration, c, and the level of the tank, h; the third state variable is the reactor temperature, T. The manipulated variables are the the outlet flow rate, F, and the coolant liquid temperature, T_c.

$$x = \begin{bmatrix} c \\ T \\ h \end{bmatrix} \quad u = \begin{bmatrix} T_c \\ F \end{bmatrix}$$

Thus, (4.1) can be rewritten as:

$$\frac{dx_1}{dt} = \frac{F_0 c_0}{\pi r^2 x_3} - k_0 \exp\left(-\frac{E}{Rx_2}\right) x_1 - \frac{x_1}{\pi r^2 x_3} u_2 \tag{4.2a}$$

$$\frac{dx_2}{dt} = \frac{F_0 (T_0 - x_2)}{\pi r^2 x_3} + \frac{(-\Delta H)}{\rho C_p} k_0 \exp\left(-\frac{E}{Rx_2}\right) x_1 - \frac{2U_0}{r\rho C_p} x_2 + \frac{2U_0}{r\rho C_p} u_1 \tag{4.2b}$$

$$\frac{dx_3}{dt} = \frac{F_0}{\pi r^2} - \frac{1}{\pi r^2} u_2 \tag{4.2c}$$

from which it is straightforward to see that it has a control affine structure and that the level dynamics is an integrator.
This class of reactors is characterized by a dynamics heavily dependent on the reactor parameters and having multiple equilibrium points (stable and unstable ones). Knowing the existence of multiple steady-states and that some may be unstable, it is important to assess how stable the reactor operation is to variations in the processing parameters. In general, stable operating point have low sensitivity to variations in the processing parameters, while clearly the unstable ones are highly sensitive. For the system under investigation the operating conditions corresponding to each open-loop steady-state are reported in Table 3.
As computer calculation are inherently discrete, a discretized model must be found from (4.1). Performing such discretization as an *explicit* 4^{th} *order Runge-Kutta* integration with a fixed sampling time $\Delta t = 12$ *sec* and a zero-order hold on the input provides an acceptable numerical accuracy and hence numerical stability. This discretization scheme consists in dividing each sampling intervals into a fixed number n^* of steps:

$$\Delta t^* = \frac{\Delta t}{n^*} \tag{4.3}$$

and then in computing the value of the state as:

$$x_{k+1} = x_k + \frac{\Delta t^*}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{4.4}$$

Table 2: Parameters of the well-stirred reactor.

Symbol	Parameter	Nominal value	Units
F_0	Inlet flow rate	0.1	$\frac{m^3}{min}$
T_0	Inlet temperature	350.0	K
c_0	Inlet concentration of the reagent	1.0	$\frac{kmol}{m^3}$
r	Radius	0.219	m
k_0	Pre-exponential factor	$7.2 \cdot 10^{10}$	min^{-1}
$\frac{E}{R}$	Activation energy over universal gas constant	8750	K
U_0	Overall heat transfer coefficient	54.94	$\frac{kJ}{min \cdot m^2 \cdot K}$
ρ	Density	1000	$\frac{kg}{m^3}$
C_p	Specific heat capacity	0.239	$\frac{kJ}{kg \cdot K}$
ΔH	Heat of reaction	$-5.4 \cdot 10^4$	$\frac{kJ}{kmol}$

Table 3: Open-loop steady-state operating point of the well-stirred reactor

Eq. point	$c \left[\frac{kmol}{m^3}\right]$	T [°C]	h [m]	Stability
# 1	0.206	97.0	0.659	stable
# 2	0.500	77.0	0.659	unstable
# 3	0.878	51.5	0.659	stable

where:

$$k_1 = f\left(x_k\right) + g\left(x_k\right) u_k \tag{4.5}$$

$$k_2 = f\left(x_k + k_1 \frac{\Delta t^*}{2}\right) + g\left(x_k + k_1 \frac{\Delta t^*}{2}\right) u_k \tag{4.6}$$

$$k_3 = f\left(x_k + k_2 \frac{\Delta t^*}{2}\right) + g\left(x_k + k_2 \frac{\Delta t^*}{2}\right) u_k \tag{4.7}$$

$$k_4 = f\left(x_k + k_3 \Delta t^*\right) + g\left(x_k + k_3 \Delta t^*\right) u_k \tag{4.8}$$

Using an $n^* = 10$ in Equation 4.3, a discrete state-space model is obtained and, assuming that only the reactant concentration and the level of the tank are measured, the output variable is:

$$y = \begin{bmatrix} c \\ h \end{bmatrix}$$

The corresponding nonlinear model has a very complex form and can be expressed in general terms as follows:

$$x_{k+1} = f\left(x_k, u_k\right) \tag{4.9a}$$

$$y_k = C x_k \tag{4.9b}$$

where:

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note that the system dynamics has no longer a control affine structure after the discretization. The input is required to satisfy the following constraints at all times:

$$\begin{bmatrix} 295.0\ \mathrm{K} \\ 0.0\ \frac{m^3}{min} \end{bmatrix} \leqslant u \leqslant \begin{bmatrix} 305.0\ \mathrm{K} \\ 0.25\ \frac{m^3}{min} \end{bmatrix}$$

while the state:

$$\begin{bmatrix} 0.0\ \frac{kmol}{m^3} \\ 320.0\ \mathrm{K} \\ 0.50\ m \end{bmatrix} \leqslant x \leqslant \begin{bmatrix} 1.0\ \frac{kmol}{m^3} \\ 375.0\ \mathrm{K} \\ 0.75\ m \end{bmatrix}$$

The control objective is to regulate the outlet concentration and the liquid level to their set-point values without offset in presence of unmeasured, nonzero disturbances acting on the system and plant-model mismatch.

4.2 AUGMENTED MODEL AND STATE ESTIMATOR DESIGN

In order to achieve offset-free control in the controlled variables the nominal model (4.9) is augmented with the disturbance representation proposed in section 3.2 to give:

$$x_{k+1} = F\left(x_k, u_k, \theta_0 + L\, d_k\right) \tag{4.10a}$$

$$d_{k+1} = d_k \tag{4.10b}$$

$$y_k = C x_k \tag{4.10c}$$

Typically, in CSTR reactors disturbances occur in the feed flow rate, temperature and composition because they are often determined by upstream processing units. Furthermore, determining kinetic and thermodynamic parameters related to a particular chemical reaction and the heat transfer coefficient is often a difficult experimental challenge which may lead to modelling errors.
For these reasons θ_0 is taken as:

$$\theta_0 = \begin{bmatrix} F_0 \\ T_0 \\ c_0 \\ \Delta H \\ \frac{E}{R} \\ k_0 \\ U_0 \end{bmatrix}$$

From (4.10) is straightforward to derive the following evolution of the augmented state ζ_k:

$$\zeta_{k+1} = F_a\left(\zeta_k, u_k, \theta_0 + L_a\zeta_k\right) \tag{4.11a}$$

$$y_k = C_a\zeta_k \tag{4.11b}$$

where:

$$C_a = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Recalled from condition (3.13) that disturbance and output vectors should have the same dimension, note that L_a must select two disturbances at the most.
The extended Kalman filter employed for (4.11) to estimate the augmented state ζ_k given the plant measurement y_k makes use of the mentioned representation of the augmented system dynamics:

$$\zeta_{k+1} = F_a\left(\zeta_k, u_k, \theta_0 + L_a\zeta_k\right) + w_k \tag{4.12a}$$

$$y_k = C_a\zeta_k + v_k \tag{4.12b}$$

The measurements noise covariance matrix is assumed to be determined from the sensor characteristics provided by the manufacturers and is taken as:

$$R = \begin{bmatrix} 10^{-5} & 0 \\ 0 & 10^{-5} \end{bmatrix}$$

while the process noise covariances are derived case by case from a manual tuning of the filter.
According to (3.9) the predicted estimates are defined as:

$$\hat{\zeta}_{k|k-1} = F_a\left(\hat{\zeta}_{k-1|k-1}, u_{k-1}, \theta_0 + L_a\hat{\zeta}_{k-1|k-1}\right) \tag{4.13a}$$

$$\hat{y}_{k|k-1} = C_a\hat{\zeta}_{k|k-1} \tag{4.13b}$$

and the filtering equation is:

$$\hat{\zeta}_{k|k} = \hat{\zeta}_{k|k-1} + K_k\left(y_k - \hat{y}_{k|k-1}\right) \tag{4.14}$$

or equivalently:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k^x\left(y_k - \hat{y}_{k|k-1}\right) \tag{4.15a}$$

$$\hat{d}_{k|k} = \hat{d}_{k|k-1} + K_k^d\left(y_k - \hat{y}_{k|k-1}\right) \tag{4.15b}$$

4.3 TARGET CALCULATOR AND CONTROLLER DESIGN

Given the current disturbance estimate, $\hat{d}_{k|k}$, the state and input targets are computed by solving the following nonlinear program with quadratic cost function:

$$\min_{x_k,u_k} \|y_k - \bar{y}_c\|^2_{Q_{ss}} \tag{4.16}$$

$$\begin{aligned} \text{subject to: } & x_k = F\left(x_k, u_k, \theta_0 + L\,\hat{d}_{k|k}\right) \\ & y_k = Cx_k \\ & r_y\left(y_k\right) = \bar{y}_c \\ & \begin{bmatrix} 0.0\ \frac{kmol}{m^3} \\ 320.0\ K \\ 0.50\ m \end{bmatrix} \leqslant x_k \leqslant \begin{bmatrix} 1.0\ \frac{kmol}{m^3} \\ 375.0\ K \\ 0.75\ m \end{bmatrix} \\ & \begin{bmatrix} 295.0\ K \\ 0.0\ \frac{m^3}{min} \end{bmatrix} \leqslant u_k \leqslant \begin{bmatrix} 305.0\ K \\ 0.25\ \frac{m^3}{min} \end{bmatrix} \end{aligned}$$

where Q_{ss} is a diagonal matrix containing the weights on the squared deviations of the output targets from the set-point values; the larger the coefficient, the more the optimizer tends to minimize the associated deviation. Weighting the deviation relative to another is simple if all outputs would have the same engineering units otherwise may be more complicated. In order to overcome this issue the weights appearing in quadratic cost functions are generally defined as the inverse square of the so called *equal concern error (ECE)*:

$$q_{ii} = \left(\frac{1}{ECE_i}\right)^2 \tag{4.17}$$

Relatively to the target calculation the ECE can be viewed as the deviation of each output target from the desired set-point (in engineering units) that causes a level of concern that is equal to all the other deviations. Note that this factor both normalize the engineering unit values and prioritized them at the same time.

As for a CSTR reactor deviations of the reactant concentration are clearly much more undesired than deviations of the tank level, the following weighting matrix is adopted:

$$Q_{ss} = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.18}$$

At each time instant the optimal control sequence is derived from the solution of the following finite horizon optimal control problem:

$$\min_{X,U} \sum_{i=0}^{N} \|y_k - \bar{y}_c\|^2_{Q_{reg}} + \|\Delta u_k\|^2_{R_{reg}} + V_f\left(\tilde{\Gamma}_N\right) \tag{4.19}$$

$$\begin{aligned}
\text{subject to: } & x_0 = \hat{x}_{k|k} \\
& x_{i+1} = F\left(x_i, u_i, \theta_0 + L\,\hat{d}_{k|k}\right) \\
& y_i = Cx_i \\
& r_y\left(y_k\right) = \bar{y}_c \\
& \begin{bmatrix} 0.0\ \frac{\text{kmol}}{\text{m}^3} \\ 320.0\ \text{K} \\ 0.50\ \text{m} \end{bmatrix} \leqslant x_i \leqslant \begin{bmatrix} 1.0\ \frac{\text{kmol}}{\text{m}^3} \\ 375.0\ \text{K} \\ 0.75\ \text{m} \end{bmatrix} \\
& \begin{bmatrix} 295.0\ \text{K} \\ 0.0\ \frac{\text{m}^3}{\text{min}} \end{bmatrix} \leqslant u_i \leqslant \begin{bmatrix} 305.0\ \text{K} \\ 0.25\ \frac{\text{m}^3}{\text{min}} \end{bmatrix} \\
& \tilde{\Gamma}_N \in \mathbb{C}_f
\end{aligned}$$

where the terminal set and the terminal cost function are designed as follows:

$$\mathbb{C}_f = \{0\} \subset \mathbb{X} \cup \mathbb{U} \tag{4.20}$$

$$V_f\left(\tilde{\Gamma}_N\right) = \|\tilde{\Gamma}_N\|^2_{\Pi^*} \tag{4.21}$$

in accordance with the stabilizing condition outlined in section 3.3.
An horizon of $N = 50$ is used and the weighting matrices are taken as:

$$Q_{reg} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad R_{reg} = \begin{bmatrix} 10^{-3} & 0 \\ 0 & 10^{-3} \end{bmatrix} \tag{4.22}$$

Substituting Equation 4.20 and Equation 4.21 into (4.19) provides the definitive form of the optimal control problem that must be solved in the regulator:

$$\min_{X,U} \sum_{i=0}^{N-1} \|\tilde{y}_i\|^2_{Q_{reg}} + \|\Delta u_i\|^2_{R_{reg}} + \|\tilde{\Gamma}_N\|^2_{\Pi^*} \tag{4.23}$$

$$\begin{aligned}
\text{subject to: } & x_0 = \hat{x}_{k|k} \\
& x_{i+1} = F\left(x_i, u_i, \theta_0 + L\,\hat{d}_{k|k}\right) \\
& y_i = Cx_i \\
& \begin{bmatrix} 0.0\ \frac{\text{kmol}}{\text{m}^3} \\ 320.0\ \text{K} \\ 0.50\ \text{m} \end{bmatrix} \leqslant x_i \leqslant \begin{bmatrix} 1.0\ \frac{\text{kmol}}{\text{m}^3} \\ 375.0\ \text{K} \\ 0.75\ \text{m} \end{bmatrix} \\
& \begin{bmatrix} 295.0\ \text{K} \\ 0.0\ \frac{\text{m}^3}{\text{min}} \end{bmatrix} \leqslant u_i \leqslant \begin{bmatrix} 305.0\ \text{K} \\ 0.25\ \frac{\text{m}^3}{\text{min}} \end{bmatrix} \\
& \tilde{\Gamma}_N \in \{0\}
\end{aligned}$$

Both target calculation and optimal control problem are solved with IPOPT (Wächter and Biegler [71]), a software library for large scale nonlinear optimization (see subsection A.3.2 for further insights) being run from the Python interface to the open-source CasADi framework (Andersson [2]).

4.4 DESIGN OF THE DISTURBANCE MODEL

According to the disturbance and parameter vectors dimensions defined previously the number of candidate augmented model results:

$$\begin{aligned} N_{can} &= \frac{n_\theta!}{n_d!\,(n_\theta - n_d)!} \\ &= \frac{7!}{2!\,(7-2)!} \\ &= 21 \end{aligned} \tag{4.24}$$

In order to determine which of them is the most suitable to be implemented in the NMPC framework the disturbance design strategy proposed in section 3.4 is applied.

4.4.1 Observability limitation

This phase of the procedure is aimed at ensuring the observability of the augmented system and varies according to the nature of the operating point of interest.

Stable operating point #3

In this case the open-loop steady-state operating conditions are the following:

$$h^s = 0.659\ m, \qquad c^s = 0.877\ \frac{kmol}{m^3}, \qquad T^s = 324.5\ K,$$
$$F^s = 0.1\ \frac{m^3}{min}, \qquad T_c^s = 300\ K.$$

from which:

$$x^s = \begin{bmatrix} 0.877 \\ 324.5 \\ 0.659 \end{bmatrix} \quad u^s = \begin{bmatrix} 300 \\ 0.1 \end{bmatrix} \quad \bar{y}_c = \begin{bmatrix} 0.877 \\ 0.659 \end{bmatrix}$$

The operating region of interest for the reactor is chosen to be ± 9.5 % around (x^s, u^s), therefore, each candidate augmented system is perturbed within the stability boundaries of the operating point in the directions identified by the undermentioned set of matrices:

$$T^n = \{I, -I\} \tag{4.25}$$

with the following different perturbation sizes for each perturbation direction:

$$M = \{0.005, 0.01, \ldots, 0.095\} \tag{4.26}$$

Thus, the observability covariance matrix W_O is computed for each candidate augmented model by solving the system (4.10) with perturbed initial conditions from time 0 to 10 min and sampling data every 12 *sec* from the simulation.

The computations result in three hundred ninety-nine matrices (nineteen perturbation sizes for each candidate augmented model) with five rows and column, which are not shown because of their number.

The degrees of observability obtained for each candidate augmented model are reported in Figure 16.

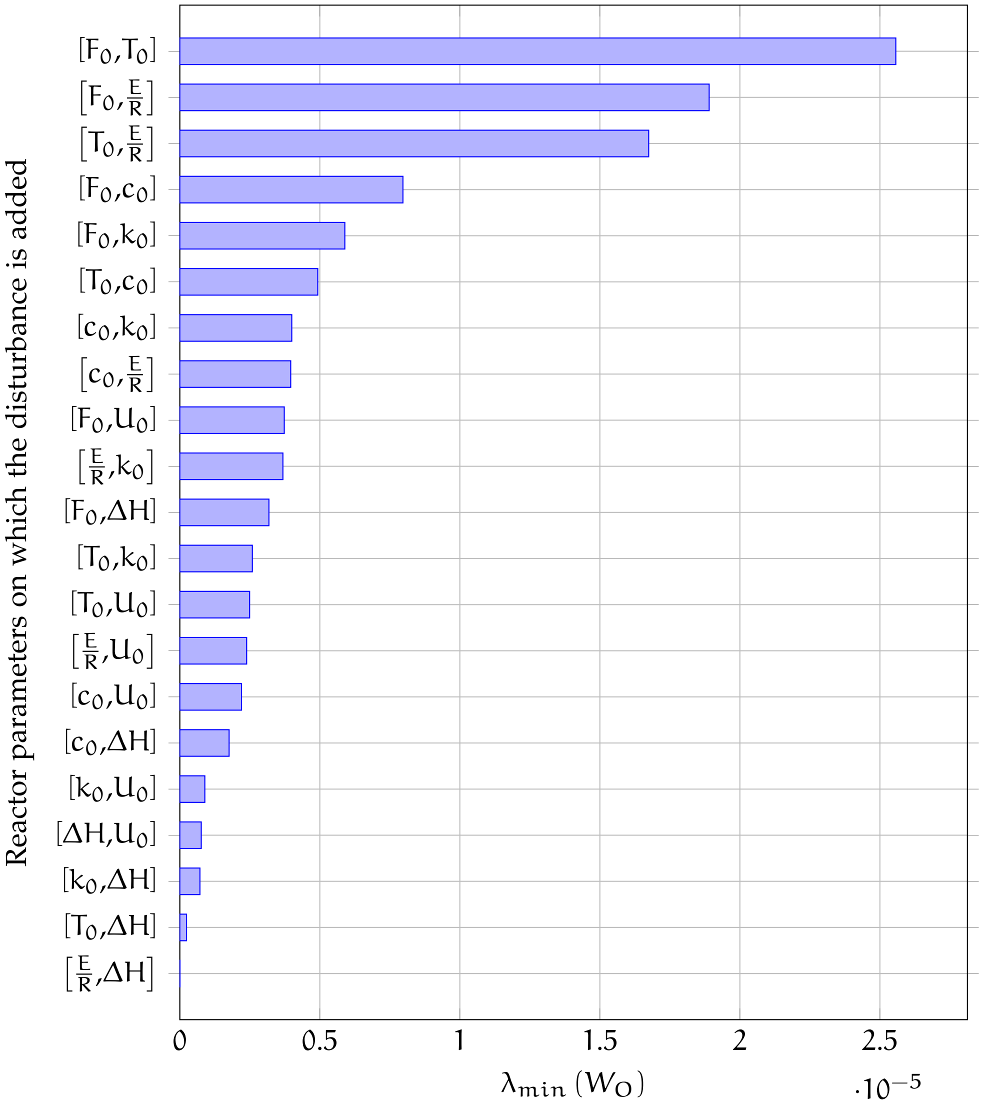


Figure 16: Degree of observability of each candidate augmented model.

Figure 16 shows that $\left[\frac{E}{R}, \Delta H\right]$, to which corresponds following continuous-time augmented model:

$$\frac{dx_1}{dt} = \frac{F_0 c_0}{\pi r^2 x_3} - k_0 \exp\left(-\frac{E}{Rx_2} + d_1\right) x_1 - \frac{x_1}{\pi r^2 x_3} u_2 \tag{4.27a}$$

$$\frac{dx_2}{dt} = \frac{F_0 (T_0 - x_2)}{\pi r^2 x_3} + \frac{(-\Delta H + d_2)}{\rho C_p} k_0 \exp\left(-\frac{E}{Rx_2} + d_1\right) x_1 - \frac{2U_0}{r\rho C_p} x_2 + \frac{2U_0}{r\rho C_p} u_1 \tag{4.27b}$$

$$\frac{dx_3}{dt} = \frac{F_0}{\pi r^2} - \frac{1}{\pi r^2} u_2 \tag{4.27c}$$

$$\frac{dd_1}{dt} = 0 \tag{4.27d}$$

$$\frac{dd_2}{dt} = 0 \tag{4.27e}$$

is unobservable in the region of interest as it is characterized by a practically zero value of $\lambda_{min}(W_0)$. As it is not possible to establish an a priori lower bound for the value of $\lambda_{min}(W_0)$, the dynamic behaviors of the other candidate augmented models are simulated for a time interval $t_{N_{sim}} = 1291$ min, in descending order of $\lambda_{min}(W_0)$, on the disturbance sequence depicted in Figure 17.

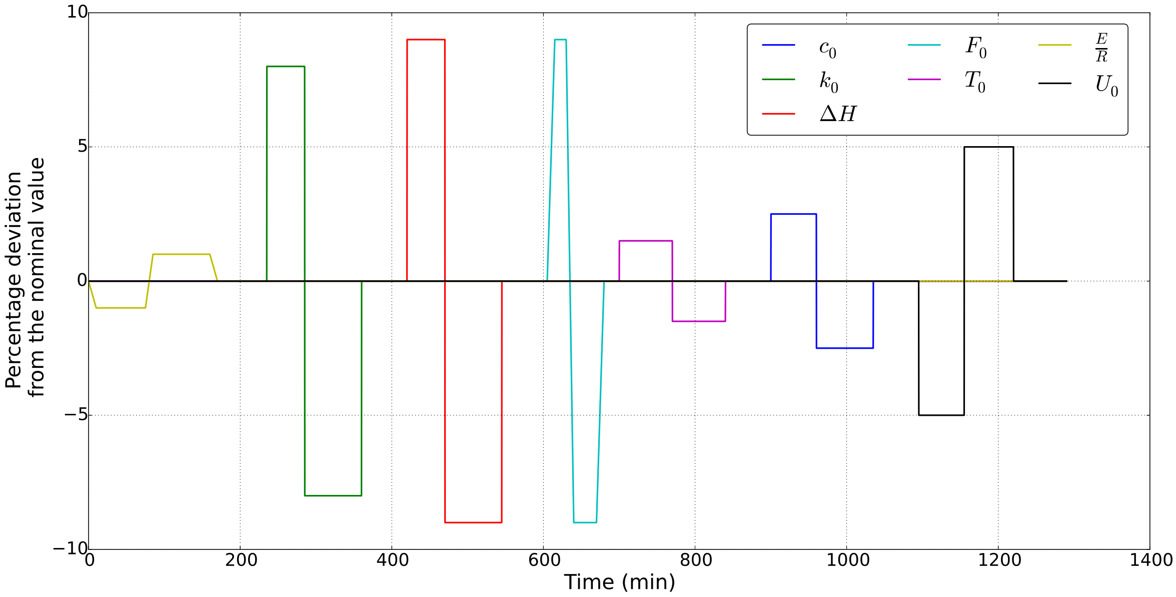


Figure 17: Time varying disturbance sequence in the case of stable operating point.

The results of the simulation made on the augmented model corresponding to $\left[T_0, \frac{E}{R}\right]$ concerning positive deviations of the inlet flow rate from its nominal value in absence of noise have been isolated from the whole sequence for a better comprehension and are reported in Figure 18.

These reveal the presence of offset in the controlled variables which is a clear signal that such augmented model is unobservable in the region of interest for disturbances acting on the inlet flow rate. In particular at the steady-state results:

$$\begin{aligned} \det\left(K^d_{N_{sim}}\right) &= \det\begin{bmatrix} -226.648 & -1.19094 \\ 34.1218 & 0.179297 \end{bmatrix} \\ &= 2 \cdot 10^{-4} \\ &\approx 0 \end{aligned}$$

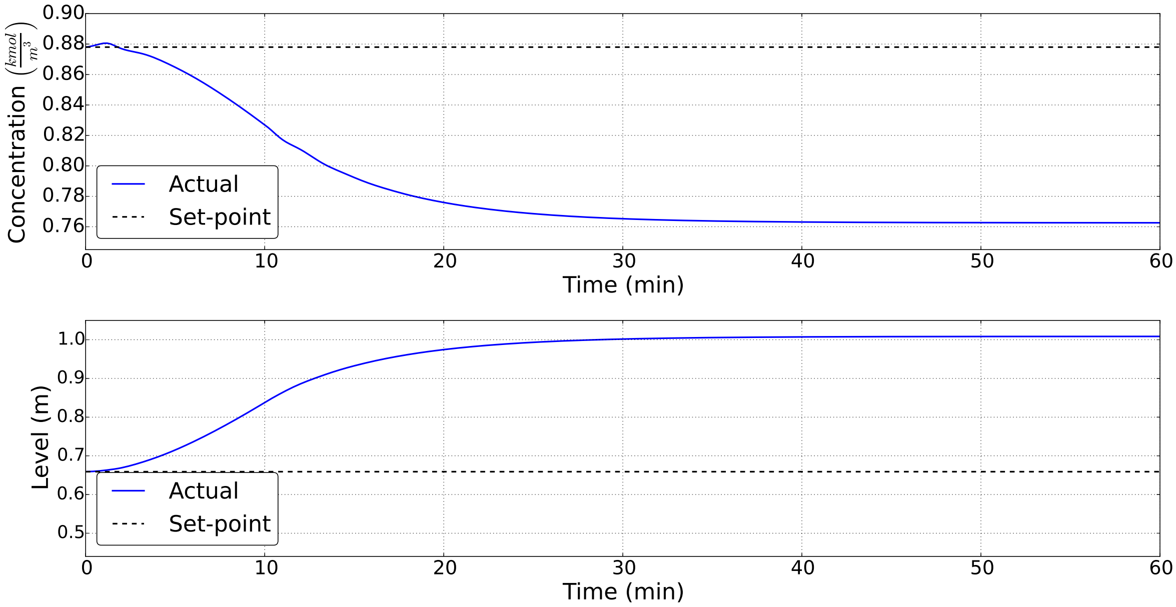


Figure 18: Closed-loop performances in presence of a disturbance on the inlet flow rate for an NMPC framework with implemented the augmented model corresponding to $\left[T_0, \frac{E}{R}\right]$.

which violates condition (2.114b) and hence prevents the disturbance estimate to converge to the true value.

Since such augmented model is not observable, all the others characterized by a smaller degree of observability are discarded while the ones linked to $[F_0, T_0]$ and $\left[F_0, \frac{E}{R}\right]$ are retained for the second phase of the procedure.

Unstable operating point #2

This steady-state is identified by the following operating condition:

$$h^s = 0.659 \text{ m}, \qquad c^s = 0.500 \, \frac{\text{kmol}}{\text{m}^3}, \qquad T^s = 350 \text{ K},$$
$$F^s = 0.1 \, \frac{\text{m}^3}{\text{min}}, \qquad T_c^s = 300 \text{ K}.$$

from which:

$$x^s = \begin{bmatrix} 0.500 \\ 350 \\ 0.659 \end{bmatrix} \qquad u^s = \begin{bmatrix} 300 \\ 0.1 \end{bmatrix} \qquad \bar{y}_c = \begin{bmatrix} 0.500 \\ 0.659 \end{bmatrix}$$

As the discrete-time system under investigation has not a control affine structure, only the Kalman rank test (2) on the linearized model can be used to check the local weak observability of each candidate augmented model. The results obtained by employing the Kalman rank test on the linearized models are shown in Table 4.

Then, the dynamic behavior of each candidate augmented model verifying the Kalman rank condition is simulated for a time interval $t_{N_{sim}} = 1291$ min in presence of the disturbance sequence reported in Figure 19.

Note that this sequence is characterized by minor deviations if compared to the one adopted for the stable steady-state since, as stated above, unstable operating points are more sensitive to variations in the processing parameters.

Even in this case the simulations show that only the candidate augmented model corresponding to $[F_0, T_0]$ and $\left[F_0, \frac{E}{R}\right]$ are observable in the region of interest and hence are the

Table 4: Results of the observability tests performed on each candidate augmented model.

Reactor parameters on which the disturbance is added	Kalman rank condition
$[F_0,T_0]$	Verified
$[F_0,\frac{E}{R}]$	Verified
$[T_0,\frac{E}{R}]$	Verified
$[F_0,c_0]$	Verified
$[F_0,k_0]$	Verified
$[T_0,c_0]$	Unverified
$[c_0,k_0]$	Unverified
$[c_0,\frac{E}{R}]$	Unverified
$[F_0,U_0]$	Verified
$[\frac{E}{R},k_0]$	Unverified
$[F_0,\Delta H]$	Verified
$[T_0,k_0]$	Unverified
$[T_0,U_0]$	Unverified
$[\frac{E}{R},U_0]$	Unverified
$[c_0,U_0]$	Verified
$[c_0,\Delta H]$	Unverified
$[k_0,U_0]$	Unverified
$[\Delta H,U_0]$	Unverified
$[k_0,\Delta H]$	Unverified
$[T_0,\Delta H]$	Unverified
$[\frac{E}{R},\Delta H]$	Unverified

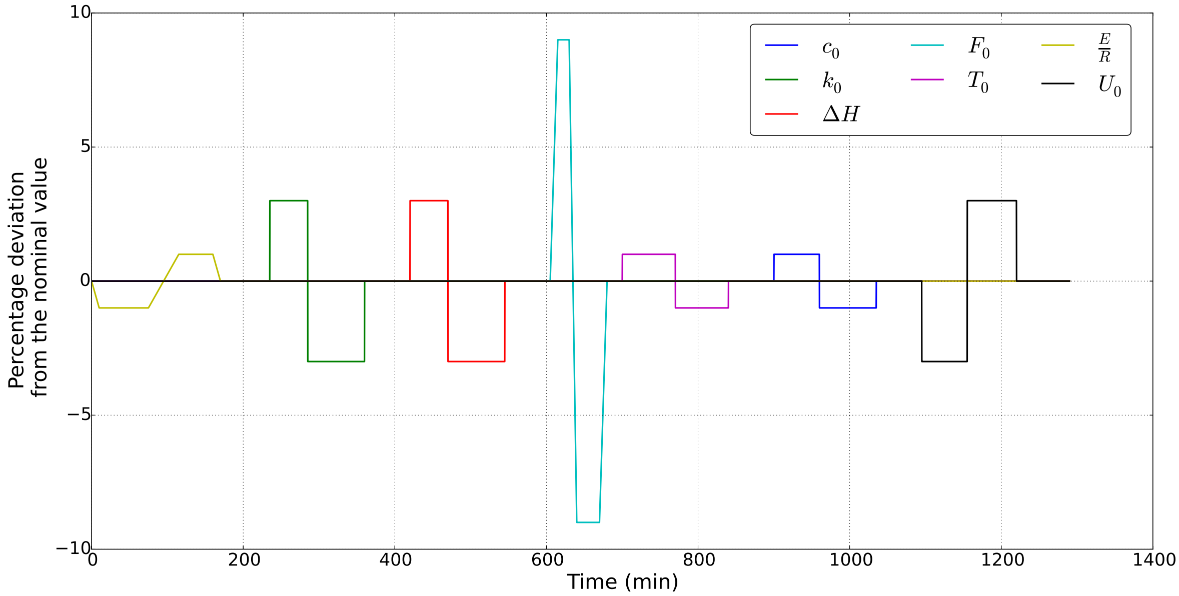


Figure 19: Time varying disturbance sequence in the case of stable operating points.

only retained for the second phase of the disturbance design procedure.

As in both cases of stable and unstable operating point the same candidate augmented models (i.e. the ones corresponding to $[F_0, T_0]$ and $\left[F_0, \frac{E}{R}\right]$) turn out to be observable in the regions of interest, the most performing disturbance model design is almost the same in both situations. Therefore, in the next subsection it will be treated only the case of the stable steady-state # 1.

4.4.2 Selection of the optimal disturbance model

In order to determine which of the two observable augmented model ensures the better closed loop performances the proposed method states that the state estimator must be tuned using an optimization based approach. Furthermore, it asserts that this optimization problem must be solved through an exhaustive search over a multidimensional grid of points followed by a certain number of iterations of the Nelder-Mead algorithm. As the problem domain has infinite dimension, it is not possible to explore it all and hence the grid should be built so as to include a region in which a sufficiently good (and possibly global) minimum is located.
In order to achieve this task there is the need to find an initial point from which to create the grid for the brute force search. In the previous phase of the procedure the extended Kalman filters employed to estimate the augmented state of each candidate augmented model have been tuned by trial and error adjusting the values of Q until acceptable simulated responses were achieved. Varying these Q matrices by trial and error in such a way as to improve the filter performances according to the undermentioned objective function:

$$J^{tun}\left(\tilde{y}_k, \Delta u_k\right) = \sum_{k=0}^{N_{sim}} \|\tilde{y}_k\|^2_{Q_{reg}} + \|\Delta u_k\|^2_{R_{reg}} \tag{4.28}$$

yields to the following initial points:

$$Q_{man}^{[F_0, \frac{E}{R}]} = \begin{bmatrix} 10^{-2} & 0 & 0 & 0 & 0 \\ 0 & 10^{-2} & 0 & 0 & 0 \\ 0 & 0 & 10^{-2} & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 1000 \end{bmatrix} \quad Q_{man}^{[F_0, T_0]} = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 1000 \end{bmatrix}$$

to which corresponds the following values of the objective function:

$$J_{tun,man}^{[F_0, \frac{E}{R}]} = 11.801 \qquad J_{tun,man}^{[F_0, T_0]} = 15.549$$

Determined $Q_{man}^{[F_0,\frac{E}{R}]}$ and $Q_{man}^{[F_0,T_0]}$, by analyzing the dynamic behavior of each associated augmented model during the simulations the following bounds on the optimization variables are derived:

$$Q_{min}^{[F_0,\frac{E}{R}]} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 500 & 0 \\ 0 & 0 & 0 & 0 & 500 \end{bmatrix} \quad Q_{max}^{[F_0,\frac{E}{R}]} = \begin{bmatrix} 10^{-2} & 0 & 0 & 0 & 0 \\ 0 & 10^{-2} & 0 & 0 & 0 \\ 0 & 0 & 10^{-2} & 0 & 0 \\ 0 & 0 & 0 & 3000 & 0 \\ 0 & 0 & 0 & 0 & 3000 \end{bmatrix}$$

$$Q_{min}^{[F_0,T_0]} = \begin{bmatrix} 10^{-2} & 0 & 0 & 0 & 0 \\ 0 & 10^{-2} & 0 & 0 & 0 \\ 0 & 0 & 10^{-2} & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 100 \end{bmatrix} \quad Q_{max}^{[F_0,T_0]} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 3000 & 0 \\ 0 & 0 & 0 & 0 & 3000 \end{bmatrix}$$

The results obtained from the brute force searches conducted in the regions subtended from such bounds are reported in Figure 20.

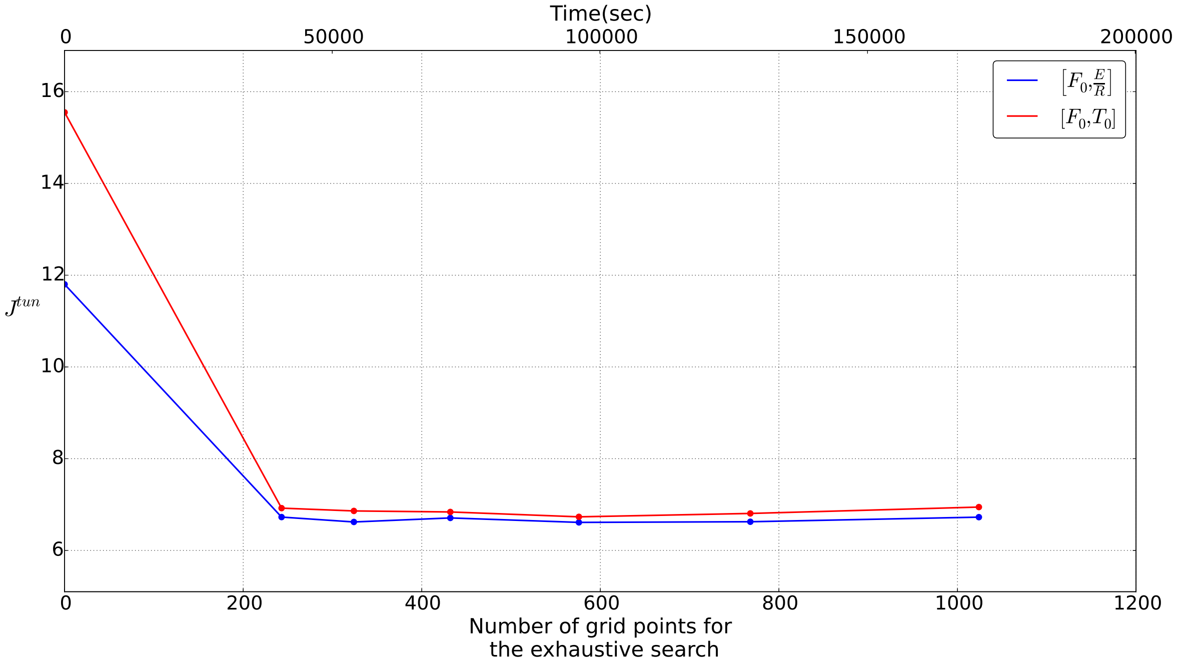


Figure 20: Objective function value against number of grid points for the two observable augmented models.

From the graph it is possible to note how the objective function (4.28) requires approximatively 2.78 min to be evaluated making the resolution of the tuning optimization problem very onerous from a computational point of view. In general, fixed the NMPC framework parameters, the function evaluation time increases with increasing values of $t_{N_{sim}}$ and hence with the number of disturbances investigated in the test sequence.

In second instance, note that in both cases the brute force search in approximatively 300 iterations leads to a $43 \div 55.5\%$ improvement of the filter performances with respect to the manual tuning. This is enough to motivate the use of an optimization based approach for the tuning of the filter and more in general for the design of the optimal augmented model.

Afterwards, the square root of each optimum found with the exhaustive search is taken as initialization for the Nelder-Mead optimizer. In order to control the computation time a maximum number of function evaluations equal to 600 has been used as no-convergence in time test (see subsection A.3.3) for the algorithm. Recalling that the objective function (4.28) needs 2.78 min to be evaluated, the criterion adopted forces the algorithm to compute a solution of the optimization problem within approximatively 28 hr at maximum.
Note that for the Nelder-Mead method such test is not equivalent to impose a limit on the maximum number of iterations, as at each iteration the optimizer may perform from one to three function evaluations depending on the simplex transformation.
The results obtained are reported in Figure 21.

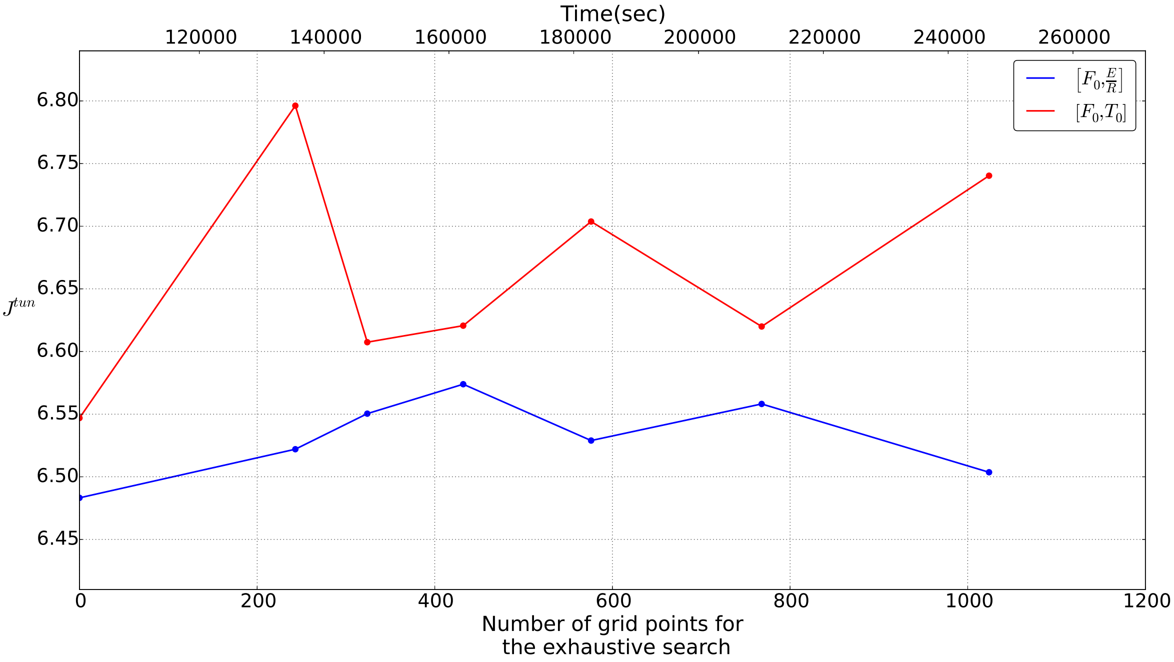


Figure 21: Objective function value against number of grid points for the two observable augmented models.

From the graph it can be noted that the minimum value of the objective function is reached by the candidate disturbance model in which the integrating terms are added to $\left[F_0, \frac{E}{R}\right]$ which is, therefore, the optimal ones to be implemented in the NMPC framework described in the previous sections of this chapter. In the aftermath, it can be stated that similar results could have been achieved even with the exhaustive search only, however, in general the usage of the Nelder-Mead optimizer provides more guarantees on the optimal solution found.
Comparing Figure 21 and Figure 20 it is straightforward to see that the experimental points are still at the same distance from each other and simply shifted in time of 28 hr. This means that for each initialization value the Nelder-Mead optimizer performs the maximum number of function evaluation allowed and hence that is possible in principle to obtain further improvements in the objective function value by increasing the maximum number of function evaluations. Alternatively, one may also think to find better local minima by enlarging the region of the objective function domain and/or by increasing the number of grid points for the exhaustive search. However, in both cases for this problem the computational cost of the optimization may become too onerous even for an off-line procedure like this.

4.5 PERFORMANCES AND COMPARISONS

In order to prove the effectiveness of the proposed method, the disturbance rejection performances of the offset-free NMPC framework derived in the previous sections are compared with the ones of the main proposals available in literature.
These latter implement one of the following alternatives:

1. Input disturbance model (see subsection 2.4.1).
2. Output disturbance model (see subsection 2.4.1).
3. Linear disturbance models (LDMs) of various types.

and use an observer manually tuned on the base of the observed responses.
As offset-free NMPC framework implementing a linear disturbance model a revisited version of the one proposed by Tatjewski [67] for the linear case is considered.
In this formulation, at each time k the current state is estimated employing an extended Kalman filter designed for the nominal model (4.9) by means of:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left(y_k - \hat{y}_k\right) \tag{4.29a}$$

$$\hat{y}_k = C\hat{x}_{k|k-1} \tag{4.29b}$$

Given the filtered estimate, a state disturbance is consequently defined and computed as:

$$d_k^x = \hat{x}_{k|k} - \hat{x}_{k|k-1} \tag{4.30}$$

$$= K_k \left(y_k - \hat{y}_k\right) \tag{4.31}$$

In addition, an output correction term, which is necessary to ensure offset-free tracking, is defined as:

$$d_k^y = y_k - \hat{y}_k \tag{4.32}$$

Thus, the prediction model used, at time k, is:

$$\hat{x}_{k+1|k} = f\left(\hat{x}_{k|k}, u_k\right) + d_k^x \tag{4.33a}$$

$$\hat{y}_k = C\hat{x}_{k|k-1} + d_k^y \tag{4.33b}$$

According to Pannocchia [42], such method is equivalent to a particularization of the general approach theorized in Pannocchia and Rawlings [47] in which the filtered estimates of states and disturbances are given by:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k^x \left(y_k - \hat{y}_k\right) \tag{4.34a}$$

$$\hat{d}_{k|k} = \hat{d}_{k|k-1} + K_k^d \left(y_k - \hat{y}_k\right) \tag{4.34b}$$

while the predictions, at time k, by:

$$\hat{x}_{k+1|k} = f\left(\hat{x}_{k|k}, u_k\right) + B_d\hat{d}_{k|k} \tag{4.35a}$$

$$\hat{d}_{k+1|k} = \hat{d}_{k|k} \tag{4.35b}$$

$$\hat{y}_k = C\hat{x}_{k|k-1} + C_d\hat{d}_{k|k} \tag{4.35c}$$

where:

$$B_d = K_k, \qquad C_d = I - CK_k, \qquad K_k^x = K_k, \qquad K_k^d = I$$

and for all k there holds that:

$$\hat{d}_{k|k} = d_k^y + Cd_k^x \tag{4.36}$$

Note that in special case in which $K_k = 0$ such LDM particularized in the ODM.
Comparisons will regard four "class" of disturbance and plant\model mismatch which often arise in CSTR reactors:

1. Disturbances on the feed properties.
2. Plant\model mismatch in the kinetic parameters of the reaction.
3. Plant\model mismatch in the overall heat transfer coefficient.
4. Plant\model mismatch in the thermodynamic parameters of the reaction.

In addition, in order to obtain results the most significant as possible they will be not considered those disturbances which enter the process through the inlet flow-rate F_0 or through the activation energy of the reaction E. Indeed, in these cases the proposed framework would achieve better performances with respect to the others simply because the disturbance model implemented resembles the actual disturbance acting on the process.
For a more straightforward interpretation of the closed-loop responses resulting from the simulations the following performance indices are used:

- closed-loop cost function (4.23):

$$J^{ocp} = \sum_{i=0}^{N-1} \|\tilde{y}_i\|^2_{Q_{reg}} + \|\Delta u_i\|^2_{R_{reg}} + \|\tilde{\Gamma}_N\|^2_{\Pi^*} \tag{4.37}$$

- *integral of the absolute magnitude of the error (IAE);*

$$IAE = \sum_{k=0}^{N_{sim}} |\bar{y}_c - y_k| \Delta t \tag{4.38}$$

- *integral of the square of the error (ISE);*

$$ISE = \sum_{k=0}^{N_{sim}} (\bar{y}_c - y_k)^2 \, \Delta t \tag{4.39}$$

- *integral of time multiplied by the absolute value of error (ITAE);*

$$ITAE = \sum_{k=0}^{N_{sim}} |\bar{y}_c - y_k| k \Delta t \tag{4.40}$$

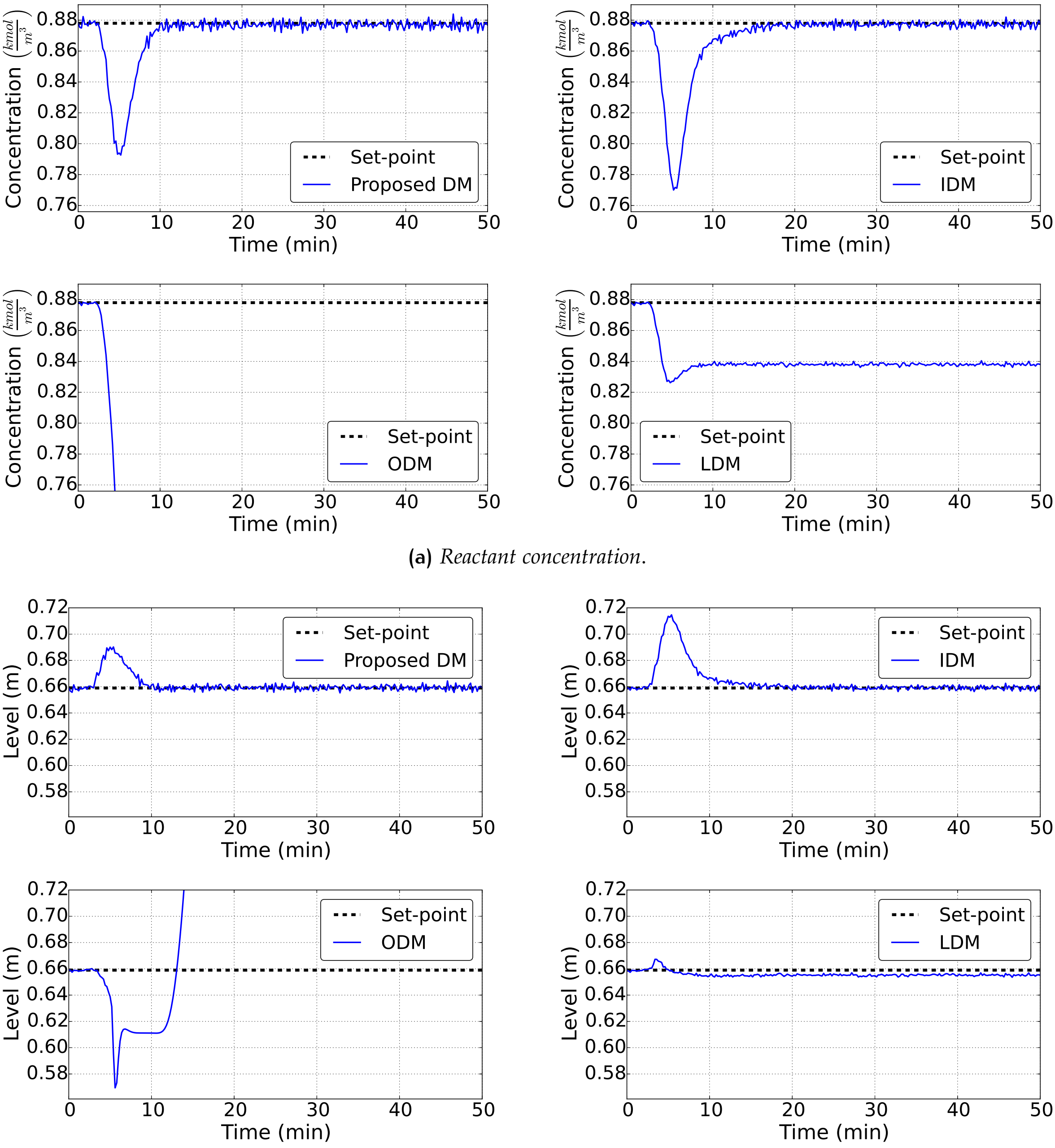
Despite revealing similarities, each error index refers to a different aspect in the performance evaluation: the IAE tends to penalize small tracking errors, the ISE large tracking errors, while the $ITAE$ the persistent ones. Before to proceed with the performance comparison, note that as the observability covariance matrix cannot be computed for such augmented systems whose disturbance model affects the output dynamics, only the degree of observability of the augmented system correlated to the IDM can be determined.
This results to be:

$$\begin{aligned} \lambda^{IDM}_{min} (W_0) &= 2.130 \cdot 10^{-5} \\ &> \lambda^{[T_0, \frac{E}{R}]}_{min} (W_0) \end{aligned}$$

Thus, only the nominal model (4.9) augmented with the IDM is guaranteed to be observable in the control region of interest for the reactor.
All the algorithms compared use the same horizon N, tuning matrices Q_{ss}, R_{ss}, Q_{reg}, R_{reg} and objective functions for the target calculator (4.16) and the regulator (4.23).

4.5.1 Disturbance on the feed temperature

In that instance the feed temperature is supposed to increase linearly from time 2 min to 3 min acting as an unmeasured disturbance. Closed-loop results for all the NMPC frameworks mentioned above are reported in Figure 22 while the related performance indices in Table 5.



(a) *Reactant concentration.*

(b) *Level of the tank.*

Figure 22: Closed-loop responses in presence of disturbances on the feed temperature.

From Figure 22 it can be noted that the ODM leads to closed-loop instability for this type of disturbance. This is due to the fact that in the ODM the integrating term does not directly influence the state dynamics but appears as a bias term on the output. This models badly the actual disturbance dynamics providing very poor disturbance estimates and hence closed-loop instability. At the same time the LDM results to be unobservable in the operating region of interest as revealed by the presence of the offset in the controlled

Table 5: NMPC frameworks performance indices related to the rejection of a disturbance in the feed temperature.

Disturbance Model	J^{ocp}	IAE_c	IAE_h	ISE_c	ISE_h	$ITAE_c$	$ITAE_h$
Proposed DM	$1.027 \cdot 10^{-1}$	0.380	0.180	$1.718 \cdot 10^{-2}$	$2.523 \cdot 10^{-3}$	4.351	2.674
IDM	$1.747 \cdot 10^{-1}$	0.467	0.278	$2.686 \cdot 10^{-2}$	$7.764 \cdot 10^{-3}$	4.675	3.123
ODM	inf.	inf.	inf.	inf.	inf.	inf.	inf.
LDM	$3.929 \cdot 10^{-1}$	1.905	0.181	$7.724 \cdot 10^{-2}$	$7.685 \cdot 10^{-4}$	50.187	4.804

variables and by the rank deficiency of its associated observability matrix evaluated at the steady-state approached.

Thus, the only disturbance models which allow to achieve offset-free performances in presence of a disturbance on the feed temperature are the IDM and the proposed DM. However, taking a look at the indices of the NMPC frameworks reported in Table 5 it can be noted that the NMPC framework which implements the IDM is characterized by higher performance indices if compared to the proposed one and hence reveals to be less effective.

In particular, referring to J^{ocp}, it results 35.31 % suboptimal with respect to the proposed DM. As both the proposed DM and IDM did not resemble the real disturbance acting on the system, this is due to the different tuning made on the state estimators.

4.5.2 Plant\model mismatch in the pre-exponential factor of the reaction

In this case the actual plant and the model (4.9) do not agree since the pre-exponential factor of the reaction is grater than its nominal counterpart; this mismatch is realistic due to the difficulty in identifying reaction kinetic parameters from experiments.

Results of the closed-loop simulations are reported in Figure 23.

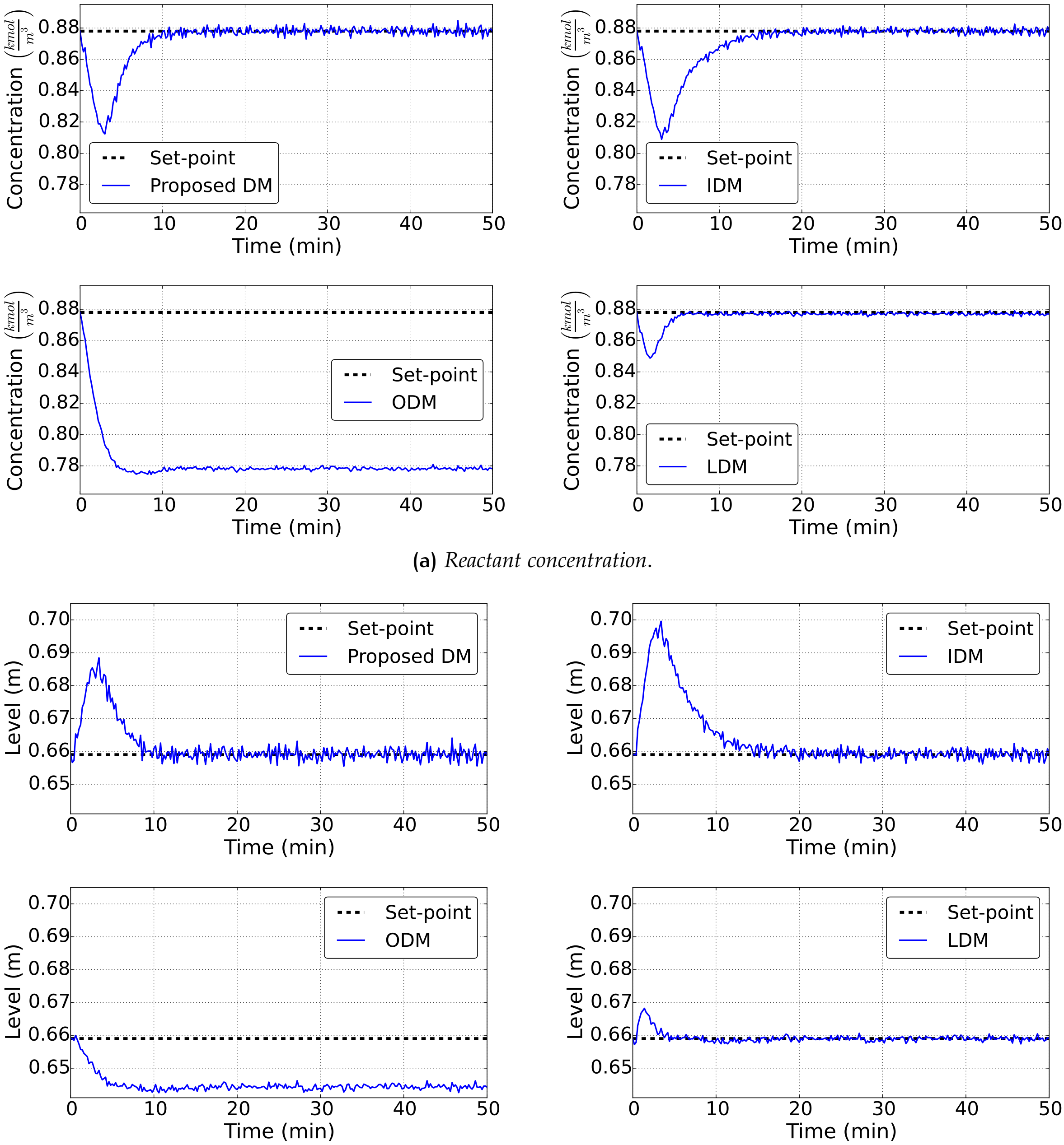
Table 6: NMPC frameworks performance indices related to the compensation of a plant\model mismatch in the pre-exponential factor of the reaction.

Disturbance Model	J^{ocp}	IAE_c	IAE_h	ISE_c	ISE_h	$ITAE_c$	$ITAE_h$
Proposed DM	$7.737 \cdot 10^{-2}$	0.343	0.182	$1.147 \cdot 10^{-2}$	$2.317 \cdot 10^{-3}$	3.361	2.389
IDM	$1.038 \cdot 10^{-1}$	0.4184	0.272	$1.150 \cdot 10^{-2}$	$5.577 \cdot 10^{-3}$	3.665	2.735
ODM	2.44	4.839	0.703	$4.778 \cdot 10^{-1}$	$1.028 \cdot 10^{-2}$	125.129	18.466
LDM	$1.122 \cdot 10^{-2}$	0.136	$6.014 \cdot 10^{-2}$	$1.606 \cdot 10^{-3}$	$1.746 \cdot 10^{-4}$	1.839	1.183

Note that in this situation the ODM leads to stable closed-loop operations. This is due to the fact that, unlike the previous case, the state dynamics depends linearly from the pre-exponential factor and hence even a simplistic linear model like the ODM, despite being unobservable, is able to provide sufficiently good disturbance estimates for the closed-loop system stabilization.

As can be noted more clearly taking a look at Figure 24, also the LDM reveals itself to be unobservable for that kind of mismatch. For this reason it is reasonable to expect that its related error indices, reported in Table 6 becomes larger than those of the frameworks implementing the IDM and the proposed DM for longer simulation times.

In conclusion note that, for similar reasons to those of the previous case, the proposed



(a) *Reactant concentration.*

(b) *Level of the tank.*

Figure 23: Closed-loop responses in presence of plant-model mismatch in the pre-exponential factor of the reaction.

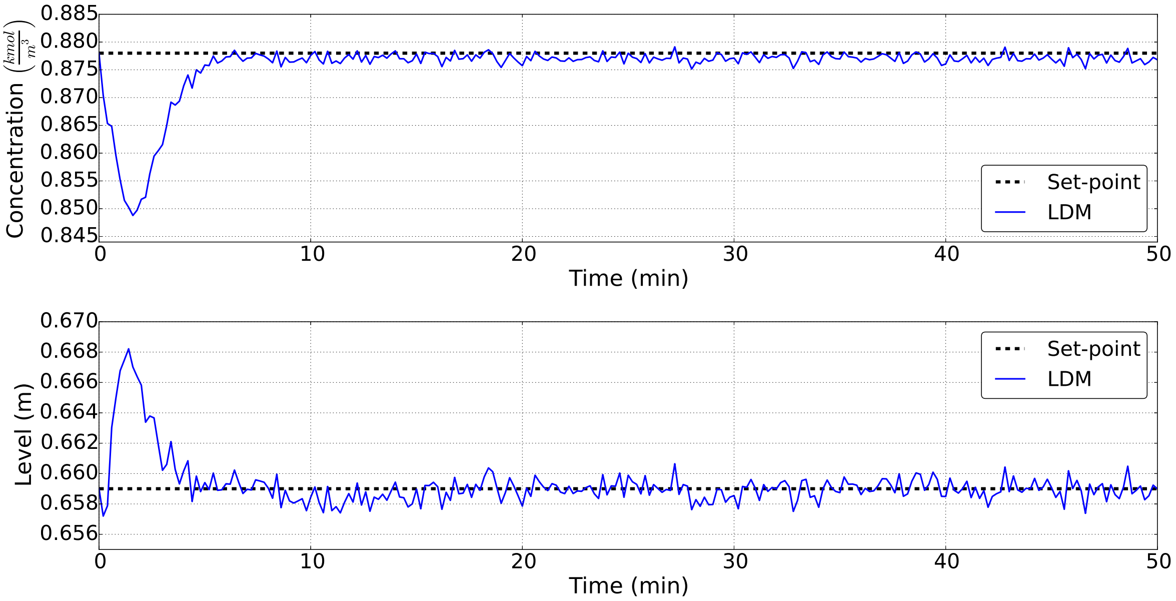


Figure 24: Closed-loop performances in presence of plant\model mismatch in the pre-exponential factor of the reaction for the NMPC framework implementing the LDM

NMPC framework performs better than the other with respect to the totality of the performance indices.

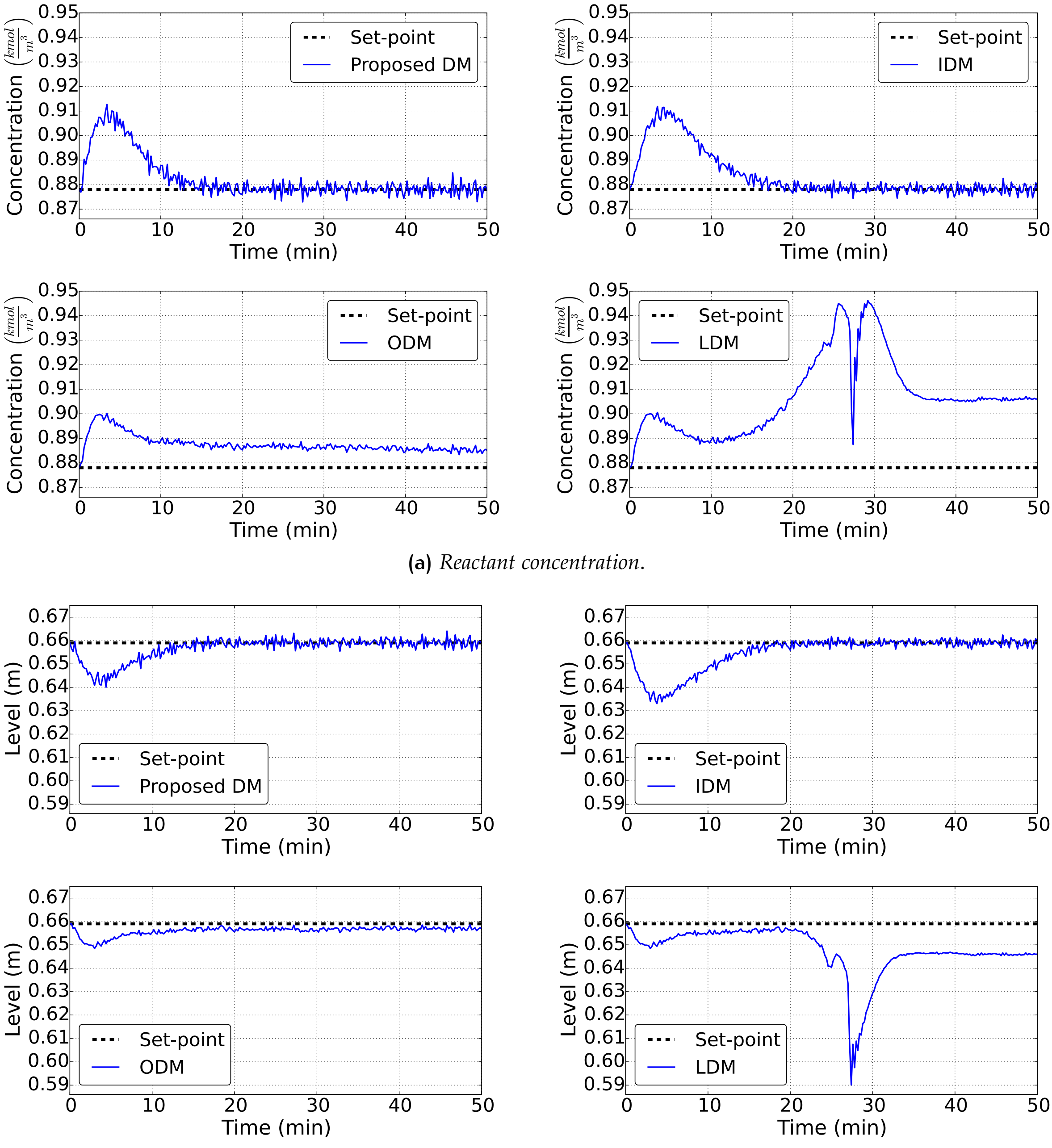
4.5.3 Plant\model mismatch in the overall heat transfer coefficient

Typically, the overall heat transfer coefficients are obtained from correlating equations or diagrams, largely available in literature for different types of flows and systems, which often have an overlooked associated with them. Furthermore, in order to be conservative and prevent possible failures designers tend to underestimate U_0.
For these reasons, the actual U_0 is often greater than its corresponding design value. The results of the simulations for this kind of mismatch are reported in Figure 25.

Table 7: NMPC frameworks performance indices related to the compensation of a plant\model mismatch in the overall heat transfer coefficient.

Disturbance Model	J^{ocp}	IAE_c	IAE_h	ISE_c	ISE_h	$ITAE_c$	$ITAE_h$
Proposed DM	$3.718 \cdot 10^{-2}$	0.285	0.169	$4.864 \cdot 10^{-3}$	$1.446 \cdot 10^{-3}$	3.432	2.464
IDM	$4.932 \cdot 10^{-2}$	0.326	0.248	$6.121 \cdot 10^{-3}$	$3.521 \cdot 10^{-3}$	3.599	2.863
ODM	$3.162 \cdot 10^{-2}$	0.483	0.151	$5.309 \cdot 10^{-3}$	$6.797 \cdot 10^{-4}$	10.385	3.008
LDM	$3.333 \cdot 10^{-1}$	1.448	0.576	$5.371 \cdot 10^{-2}$	$1.212 \cdot 10^{-2}$	40.8672	17.401

Figure 25 shows similar results to those obtained in the previous case i.e. that the NMPC frameworks implementing the ODM and the LDM, despite ensuring stable closed-loop operations, cannot achieve offset-free performances for that kind of mismatch because of unobservability of the augmented model employed at the steady-states approached.
This task is conversely accomplished by the NMPC frameworks which implement the IDM and the proposed DM. This latter is also the one characterized by the lower error indices, probably because of the optimal tuning made on its state estimator unit.



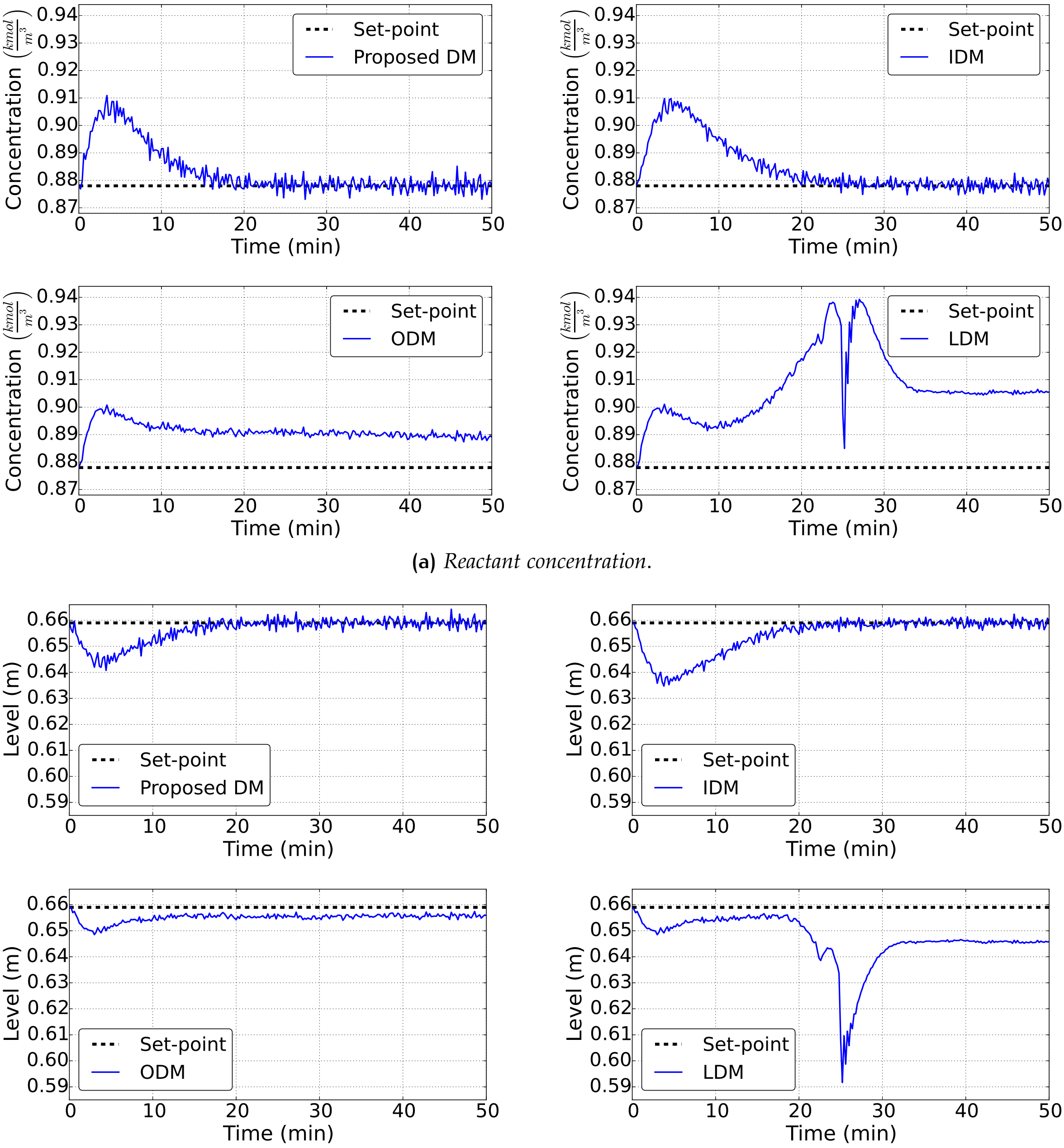
(a) *Reactant concentration.*

(b) *Level of the tank.*

Figure 25: Closed-loop responses in presence of plant-model mismatch in the overall heat transfer coefficient.

4.5.4 Plant\model mismatch in the enthalpy of reaction

In this case the actual plant and the model (4.9) do not agree as the enthalpy of reaction is lower than its nominal value. Note that even this mismatch is realistic due to the difficulty in determining reaction thermodynamics parameters from experiments.
Results of the closed-loop simulations are reported in Figure 26.



(a) *Reactant concentration.*

(b) *Level of the tank.*

Figure 26: Closed-loop responses in presence of plant-model mismatch in the enthalpy of reaction.

From Figure 26 it can be noted that also for this type of mismatch the ODM and the LDM are unobservable and hence reveals themselves unable to remove offset in the controlled variables. On the other hand the IDM is observable and even in this case, according to the error indices reported in Table 8, provides only slightly worse performances than the ones of the proposed DM. For this reason it can be stated that it is a practical and effective

Table 8: NMPC frameworks performance indices related to the compensation of a plant\model mismatch in the enthalpy of reaction.

Disturbance Model	J^{ocp}	IAE_c	IAE_h	ISE_c	ISE_h	$ITAE_c$	$ITAE_h$
Proposed DM	$3.812 \cdot 10^{-2}$	0.309	0.179	$5.123 \cdot 10^{-3}$	$1.506 \cdot 10^{-3}$	3.767	2.602
IDM	$5.158 \cdot 10^{-2}$	0.357	0.269	$6.443 \cdot 10^{-3}$	$3.643 \cdot 10^{-3}$	4.117	3.221
ODM	$5.23 \cdot 10^{-2}$	0.661	0.201	$9.140 \cdot 10^{-3}$	$9.825 \cdot 10^{-4}$	15.382	4.458
LDM	$3.212 \cdot 10^{-1}$	1.463	0.625	$5.055 \cdot 10^{-2}$	$1.288 \cdot 10^{-2}$	39.589	18.225

alternative to this latter in those situations in which high performances are not required but there is only the need of achieving offset-free control.

4.6 RESULTS

In this chapter, an offset-free NMPC framework for the control of a non-isothermal CSTR reactor has been design according to the strategy proposed in chapter 3.
The nominal model have been derived from material and energy balances on the reactor while the disturbance model, once determined the process parameter more frequently affected by disturbances and/or modeling errors, have been design in two phases. The first aimed at ensuring the observability of the augmented model in the region of interest for the control and the second devoted to the optimization of the extended Kalman filter (employed as state estimator) performances in closed-loop over the test disturbance sequence depicted in Figure 19.
The effectiveness of the numerical methods proposed for the solution of such optimization problem has been proved through simulation examples which also quantify the benefits provided by the proposed optimal tuning strategy with respect to the manual one. Furthermore, this optimization needs to be performed off-line, thus with no (reasonable) concerns for its computational complexity.
Next the disturbance rejection properties of the NMPC framework synthesized have been compared in several cases with the ones of those implementing the main disturbance model alternatives available in literature: the IDM, the ODM and the LDM.
Comparative simulations show that, according to the performance indices adopted, the proposed offset-free NMPC algorithm, despite being its design very-time consuming, outperforms the other considered while maintaining a good robustness to noises.
The augmented model corresponding to the IDM does not suffer from observability issues for plant containing integrators and is characterized by a sufficiently high degree of observability to be observable in the region of interest for the control as reveled by the simulation examples. Mainly for this reason it should be considered the best alternative to the proposed disturbance model for this type of application.
The ODM, conversely, is unable to ensure stable closed-loop operations in presence of a disturbance with an highly nonlinear dynamics like the one in the feed temperature T_0 as in this model the disturbance term does not directly influence the state dynamics but appears only as a bias term on the output. This models badly the actual disturbance dynamics and leads to very poor control performances. In addition, as the level dynamics is an integrator the augmented model corresponding to the ODM reveal itself to be unobservable in all the other cases.

Similar results are obtained for the LDM which however, having a linear term affecting the state transition and hence being less simplistic than the ODM, does not lead to unstable closed-loop operations even for a disturbance entering in the feed temperature.

5 CONCLUSION

In this final pages the main accomplishments of this thesis are summarized and the conclusions of this work are drawn. Both strong and weak points of the proposed method are described, providing some hints for future improvements and making some research proposal.
This work has been a first trial to:

1. implement a class of disturbance models in which the integrating term is added to model parameters in an NMPC offset-free framework;
2. verify and quantitatively assess the a priori observability of the augmented nonlinear model to be implemented;
3. optimize the state estimator performances in closed-loop according to a criterion of optimality properly designed for this class of problems;
4. develop a systematic procedure for the design of the optimal nonlinear disturbance model of the class recalled in pt.1 to be implemented in an NMPC offset-free framework.

The results show that:

- The usage of a class of disturbance models in which the integrating term is added to model parameters provides significant advantages over linear ones and input disturbance model for both control and estimation.
- Verifying the a priori observability of the augmented systems resorting to the smallest eigenvalue of their associated observability covariance matrix guarantees more reliable results in a broader set of state trajectories than other methods employed in literature which rely on a linearization of the model (Morari and Maeder [37]) or on differential geometry.
  Nevertheless, these latter methods are useful in performing observability analysis on unstable operating points, as the observability covariance matrix of the augmented model cannot be determined in such situations.
- The optimization of the state estimator performances, according to the suggested optimality criterion, despite being very time-consuming, allows to obtain an NMPC framework able to reject disturbances entering the process faster than another with a manual tuning of the estimator while maintaining a good robustness to measurements noise.
- The proposed strategy is effective in its aim of designing the optimal nonlinear disturbance model to be implemented in an offset-free NMPC framework, however, because of the difficulties and time involved in its application, the benefits which it provides have to be justified.

OPEN ISSUES AND FURTHER IMPROVEMENTS

While this thesis has addressed some open questions about the implementation of nonlinear disturbance models in offset-free NMPC frameworks, many other questions concerning nonlinear model predictive control remain open; an overview of them can be found e.g. in the work of Findeisen and Allgöwer [14].
This section is devoted to suggest some improvements to the proposed strategy that, for several reasons, have not been included in this work. These modifications concern both adding new capabilities and refining of the existing features:

Development of novel approaches for the evaluation of the a priori observability of nonlinear systems in operating regions around unstable steady-states.

As stated many times before, the observability covariance matrix cannot be determined for systems operating on unstable steady-states. Thus, in such situations the procedure foresees to make use of the Kalman rank test on the linearized model or of the observability rank condition. However, these latter methods are valid only locally and give exclusively a yes or no answer preventing from sorting the candidate augmented model on the base of their degree of observability. This facts imply that in the case of unstable operating points there is often a greater number of candidate augmented model to test without, on the other hand, having guarantees on their observability in the region of interest. For these reasons it could be useful to develop a novel approach for the evaluation of the a priori observability of nonlinear systems in operating regions around unstable steady-states (see e.g. Vaidya [70]). This, once implemented, would allow to reduce the time required by the first phase of the procedure and to obtain more reliable results.

Development of more efficient optimal tuning problem formulations

In subsection 3.4.2 a novel approach for the optimal tuning of the state estimator unit in closed-loop have been derived. This overcomes the issues related to the optimization algorithms convergence encountered when using criteria based on the prediction error and opens to new possibilities. In this sense, it may be worth studying the effects of different optimal tuning problem objective functions on the controller performances and develop more efficient formulations.

Usage of more efficient computational strategies

Another field where ulterior improvements are possible is the one of the solvers. All simulation conducted in this work have been performed by using Mumps [1] as linear solver, however, IPOPT can use also the HSL routines [22] MA27, MA57, HSL_MA77, HSL_MA86, and HSL_MA97 when provided as shared library. Such routines are expected to drastically reduce the time required for the solution of the optimization problems (3.14) and (3.18) and hence for the evaluation of the objective function (3.40). Further enhancements in this sense could be achieved by replacing the Nelder-Mead optimizer with faster algorithms robust to noise. For instance, Elster and Neumaier [13] develop a robust algorithm for the resolution of optimization problems characterized by noisy objective functions which reveals itself to be more reliable and on average twice as fast in terms of total number of function evaluations with respect to the Nelder-Mead approach.
The modifications mentioned above would allow to enlarge the region of the domain and/or increase the number of grid points for the exhaustive search and/or perform a

greater number of iterations of the local optimizer and, consequently, achieve better results.

Implement faster and more reliable recursive filters

Typically, when the state transition and observation models are highly nonlinear the extended Kalman filter, although easily implementable, may show slow convergence properties. This is mainly due to the fact that the state distribution is approximated by a Gaussian random variable, which is then propagated analytically through a first order Taylor series linearization of the nonlinear process model. The usage of recursive filters able to capture the true posterior mean and covariance more accurately than the EKF could allow to solve this problem and speed up the procedure.

Among possible alternatives, most notable is the *unscented Kalman filter (UKF)* proposed by Julier and Uhlmann [25]. Employing this filter the state distribution is still approximated by a Gaussian random variable but it is represented using a minimal set of properly chosen sample points. This makes possible to describe the true mean and covariances with an accuracy of the 3rd order (Taylor series expansion) for any nonlinearity while maintaining a computational complexity of the same order as that of the EKF.

A SUPPLEMENTARY MATERIAL

For the sake of completeness, several Definitions, Theorems and mathematical tools which are of central importance for the thesis are reviewed in this appendix. The goal is to focus on intuitive understanding of their meaning rather than on generality and thoroughness of arguments. However, the reader interested in a more complete and rigorous theoretical treatment of these topics will find many references to the relevant literature.

A.1 PROBABILITY THEORY

Here are summarized some notions from probability theory. This summary only concerns those concepts that are part of the mathematical background required for a better comprehension of many topics of this work in particular the ones concerning state estimation. The material presented is taken by Simon [62] and Klenke [27].

A.1.1 Probability

The *probability* of an event A is defined as:

$$P(A) = \frac{\text{Number of times A occurs}}{\text{Number of occurrances}} \tag{A.1}$$

The probability of a single event is called an *a priori* probability, because it refers to an event apart from any previously known information.

Denoted with B an event possibly related with A, one may be interested in an *a posteriori* probability, that is, the probability of A to occur given the fact that some information about B are already known. Two common a posteriori probabilities are the *conditional probability* of A given B, $P(A|B)$, specifically the probability that A occurs given the fact that B occurred, and the *joint probability* of A and B, $P(A, B)$, that is the probability that A and B both occur. These quantities are related by the following expression:

$$P(A|B) = \frac{P(A, B)}{P(B)} \tag{A.2}$$

If A and B are *independent events*, i.e. the occurrence of B has no effect on the probability of the occurrence of A, the previous relation becomes:

$$\begin{aligned} P(A|B) &= \frac{P(A, B)}{P(B)} \\ &= \frac{P(A)\cancel{P(B)}}{\cancel{P(B)}} \\ &= P(A) \end{aligned} \tag{A.3}$$

A.1.2 Random variables and probability density

Defined a random variable (RV), ξ, as a mapping from a set of events to real numbers, the probability that ξ takes on a value less than or equal to x is given by its *probability distribution function*, $F_{\xi}(x)$, formalized as follows:

$$F_{\xi}(x) = P(\xi \leqslant x) \tag{A.4}$$

Another function of interest is the *probability density function*, $p_{\xi}(x)$, which is related to $F_{\xi}(x)$ by the expression:

$$p_{\xi}(x) = \frac{dF_{\xi}(x)}{dx} \tag{A.5}$$

and has the undermentioned interpretation in terms of probability:

$$\int_{x_1}^{x_2} p_{\xi}(x)\, dx = P(x_1 \leqslant \xi \leqslant x_2) \tag{A.6}$$

Given these quantities, it is possible to define the following characteristic statistical properties of any RV:

- *Moment*

$$E[\xi^n] = \int_{-\infty}^{\infty} x^n p_{\xi}(x)\, dx \tag{A.7}$$

- *Mean or first moment about zero*

$$\begin{aligned} \bar{\xi} &= E[\xi] \\ &= \int_{-\infty}^{\infty} x p_{\xi}(x)\, dx \end{aligned} \tag{A.8}$$

- *Variance or second moment about the mean*

$$\begin{aligned} var(\xi) &= \int_{-\infty}^{\infty} \left(x - \bar{\xi}\right)^2 p_{\xi}(x)\, dx \\ &= E\left[\left(\xi - \bar{\xi}\right)^2\right] \\ &= E\left[\xi^2\right] - E^2[\xi] \end{aligned} \tag{A.9}$$

- *Standard deviation*

$$\sigma(\xi) = (var(\xi))^{\frac{1}{2}} \tag{A.10}$$

- *Root mean square value*

$$\psi(\xi) = \sqrt{\bar{\xi}^2 + \sigma^2(\xi)} \tag{A.11}$$

- *Mean square value*

$$\begin{aligned} \psi^2(\xi) &= \int_{-\infty}^{\infty} x^2 p_{\xi}(x)\, dx \\ &= \bar{\xi}^2 + \sigma^2(\xi) \end{aligned} \tag{A.12}$$

A.1.3 Conditional probability

As stated above, the probability that ξ takes on a value less than or equal to a given scalar x is provided by $F_{\xi}(x)$. In analogy, the probability that both two RVs ξ and η take on values less than or equal to the scalars x and y respectively is given by the *joint probability distribution function*, defined as:

$$F_{\xi,\eta}(x,y) = P(\xi \leqslant x, \eta \leqslant y) \tag{A.13}$$

Thus, the related *joint probability density function* can be derived as:

$$p_{\xi,\eta}(x,y) = \frac{\partial^2 F_{\xi,\eta}(x,y)}{\partial x \partial y} \tag{A.14}$$

and its meaning in terms of probability is given by the undermentioned expression:

$$\int_{y_1}^{y_2}\int_{x_1}^{x_2} p_{\xi,\eta}(x,y)\, dx\, dy = P(x_1 \leqslant \xi \leqslant x_2, y_1 \leqslant \eta \leqslant y_2) \tag{A.15}$$

By choosing $\xi \leqslant x$ and $y \leqslant \eta \leqslant y + \delta y$ as events A and B respectively, from Equation A.2 follows:

$$P(\xi \leqslant x | y \leqslant \eta \leqslant y + \delta y) = \frac{P(\xi \leqslant x, y \leqslant \eta \leqslant y + \delta y)}{P(y \leqslant \eta \leqslant y + \delta y)} \tag{A.16}$$

and hence:

$$P(\xi \leqslant x | y \leqslant \eta \leqslant y + \delta y) = \frac{F_{\xi,\eta}(x, y + \delta y) - F_{\xi,\eta}(x,y)}{F_{\eta}(y + \delta y) - F_{\eta}(y)} \tag{A.17}$$

Taking the limits of both side of Equation A.17 as δy tends to zero and using the L'Hopital's rule provide:

$$\begin{aligned} F_{\xi|\eta}(x|y) &= \lim_{\delta y \to 0} \left(\frac{\partial F_{\xi,\eta}(x, y + \delta y) - F_{\xi,\eta}(x,y)}{\partial \delta y} \frac{1}{\frac{\partial F_{\eta}(y+\delta y) - F_{\eta}(y)}{\partial \delta y}} \right) \\ &= \frac{\partial F_{\xi,\eta}(x,y)}{\partial y} \frac{1}{p_{\eta}(y)} \end{aligned} \tag{A.18}$$

where $F_{\xi|\eta}(x|y)$ is the *conditional probability distribution function* of ξ given a specific realization y of η. Differentiating the above expression with respect to x yields:

$$p_{\xi|\eta}(x|y) = \frac{p_{\xi,\eta}(x,y)}{p_{\eta}(y)} \tag{A.19}$$

where $p_{\xi|\eta}(x|y)$ is the *conditional probability density function* of ξ given a specific realization y of η.
Recalling that two events are statistically independent if they satisfy:

$$P(A,B) = P(A)\,P(B) \tag{A.20}$$

from the definitions of joint distribution and density function it is straightforward to see that this implies:

$$F_{\xi,\eta}(x,y) = F_{\xi}(x)\,F_{\eta}(y) \tag{A.21}$$

$$p_{\xi,\eta}(x,y) = p_{\xi}(x)\,p_{\eta}(y) \tag{A.22}$$

Thus, if RVs ξ and η are statistically independent, then Equation A.19 takes the form:

$$\begin{aligned} p_{\xi|\eta}(x|y) &= \frac{p_{\xi,\eta}(x,y)}{p_{\eta}(y)} \\ &= \frac{p_{\xi}(x)\,\cancel{p_{\eta}(y)}}{\cancel{p_{\eta}(y)}} \\ &= p_{\xi}(x) \end{aligned} \tag{A.23}$$

A.1.4 Multivariate statistics

In applications where there is a need to deal with multiple RVs one may think to group them in a vector and let it takes on values in $\mathbb{R}^n$ with $n > 1$. Given two random column vectors ξ and η it is possible to define the *covariance matrix* as:

$$\operatorname{cov}(\xi,\eta) = E\left[\left(\xi-\bar{\xi}\right)(\eta-\bar{\eta})^T\right] \tag{A.24}$$

Such matrix takes the name of *autocovariance matrix* in the special case of $\eta = \xi$.
Since $\operatorname{cov}(\xi,\xi)_{ij} = \operatorname{cov}(\xi,\xi)_{ji}$ an autocovariance matrix is always symmetric. Moreover, for any n-element column vector z holds that:

$$\begin{aligned} z^T\operatorname{cov}(\xi,\xi)\,z &= z^T E\left[\left(\xi-\bar{\xi}\right)\left(\xi-\bar{\xi}\right)^T\right] z \\ &= E\left[z^T\left(\xi-\bar{\xi}\right)\left(\xi-\bar{\xi}\right)^T z\right] \\ &= E\left[\left(z^T\left(\xi-\bar{\xi}\right)\right)^2\right] \\ &\geqslant 0 \end{aligned} \tag{A.25}$$

So an autocovariance matrix is also positive semidefinite.
Autocovariance matrix and mean value are sufficient to fully characterize the probability density function of a *Gaussian* (or normal) n-valued RV. For this reason the following notation:

$$\begin{gathered} \xi \sim \mathcal{N}(m,P) \\ p_{\xi}(x) = n(x,m,P) \end{gathered}$$

is commonly used to denote a normally distributed random vector ξ with mean m and autocovariance P, in which:

$$n(x,m,P) = \frac{1}{\sqrt{(2\pi)^n\det(P)}}\exp\left[-\frac{1}{2}(x-m)^T P^{-1}(x-m)\right] \tag{A.26}$$

Several interesting results are available for this class of vectors. Most notable, taken form Rawlings and Mayne [55], are reported below.

JOINT INDEPENDENT NORMALS. Let ξ and η be random vectors and suppose that they have realizations x, y respectively. If:

i) $\xi \sim \mathcal{N}(m_{\xi},P_{\xi})$.

ii) $\eta \sim \mathcal{N}(m_{\eta},P_{\eta})$.

iii) ξ is statistically independent of η.

then their joint density is given by:

$$\begin{bmatrix}\xi\\ \eta\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}m_{\xi}\\ m_{\eta}\end{bmatrix},\begin{bmatrix}P_{\xi} & 0\\ 0 & P_{\eta}\end{bmatrix}\right) \tag{A.27}$$

LINEAR TRANSFORMATION OF A NORMAL Let ξ be a normally distributed random vectors with mean m and autocovariance P. If η is a linear transformation of ξ of the following type:

$$\eta = A\xi$$

where A is a properly dimensioned matrix, then results:

$$\eta \sim \mathcal{N}\left(Am, APA^T\right) \tag{A.28}$$

CONDITIONAL OF A JOINT NORMAL Let ξ and η be random vectors and suppose that they have realizations x, y respectively. If ξ and η are jointly normally distributed as:

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_\xi \\ m_\eta \end{bmatrix}, \begin{bmatrix} P_\xi & P_{\xi\eta} \\ P_{\eta\xi} & P_\eta \end{bmatrix}\right) \tag{A.29}$$

then:

$$p_{\xi|\eta}\left(x|y\right) = n\left(x, m, P\right) \tag{A.30}$$

with:

$$m = m_\xi + P_{\xi\eta}P_\eta^{-1}\left(y - m_\eta\right) \tag{A.31}$$

$$P = P_\xi - P_{\xi\eta}P_\eta^{-1}P_{\eta\xi} \tag{A.32}$$

m is known as the *conditional mean* of ξ given η and can be derived regardless of the shape of the distribution as:

$$m = \int_{-\infty}^{\infty} x p_{\xi|\eta}\left(x|y\right)\, dx \tag{A.33}$$

Extending the previous results to the case in which they are conditioned on additional random variables yields what follows.

JOINT INDEPENDENT NORMALS. Let ξ, η and ϑ be random vectors and suppose that they have realizations x, y and z respectively. If:

i) $p_{\xi|\vartheta}\left(x|z\right) = n\left(x, m_x, P_x\right)$.

ii) $\eta \sim \mathcal{N}\left(m_\eta, P_\eta\right)$.

iii) η is statistically independent of ξ and ϑ.

then:

$$p_{\xi,\eta|\vartheta}\left(\begin{bmatrix} x \\ y \end{bmatrix} \middle| z\right) = n\left(\begin{bmatrix} x \\ y \end{bmatrix}, \begin{bmatrix} m_\xi \\ m_\eta \end{bmatrix}, \begin{bmatrix} P_\xi & 0 \\ 0 & P_\eta \end{bmatrix}\right) \tag{A.34}$$

LINEAR TRANSFORMATION OF A NORMAL. Let ξ and ϑ be random vectors and suppose that $p_{\xi|\vartheta}\left(x|z\right)$ is a Gaussian with mean m and covariance P. If η is a linear transformation of ξ of the following type:

$$\eta = A\xi$$

where A is a properly dimensioned matrix, then results:

$$p_{\eta|\vartheta}\left(y|z\right) = n\left(y, Am, APA^T\right) \tag{A.35}$$

CONDITIONAL OF A JOINT NORMAL Let ξ and η be random vectors and suppose that they have realizations x, y respectively. If ξ and η are jointly normally distributed as:

$$p_{\xi,\eta|\vartheta}\left(\begin{bmatrix} x \\ y \end{bmatrix}\middle| z\right) = n\left(\begin{bmatrix} x \\ y \end{bmatrix}, \begin{bmatrix} m_\xi \\ m_\eta \end{bmatrix}, \begin{bmatrix} P_\xi & P_{\xi\eta} \\ P_{\eta\xi} & P_\eta \end{bmatrix}\right) \tag{A.36}$$

then:

$$p_{\xi|\eta,\vartheta}\left(x|y,z\right) = n\left(x, m, P\right) \tag{A.37}$$

in which:

$$m = m_\xi + P_{\xi\eta} P_\eta^{-1}\left(y - m_\eta\right) \tag{A.38}$$

$$P = P_\xi - P_{\xi\eta} P_\eta^{-1} P_{\eta\xi} \tag{A.39}$$

A.1.5 Random processes and white noise

When the random vector ξ defined above changes with respect to an independent variable it takes the name of random process RP.
Let $\xi\left(t\right)$ be an RP and suppose that it has a realization $x\left(t\right)$. If the random vector $\xi\left(t_1\right)$ is statistically independent from the random vector $\xi\left(t_2\right)$ for all times $t_1 \neq t_2$ then $\xi\left(t\right)$ is called white noise.
By assuming that $\xi\left(t\right)$ is an ergodic RP and hence that its statistical properties are invariant with respect to translation in time, they can be computed over a long enough time interval of length T as:

$$\bar{\xi} = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} x\left(x\right)\, dt \tag{A.40}$$

$$var\left(\xi\right) = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} \left(x(t) - \bar{\xi}\right)^2\, dt \tag{A.41}$$

$$\psi^2\left(\xi\right) = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} x^2(t)\, dt \tag{A.42}$$

and so on for all the other previously defined statistics.
Given an expression for $\psi\left(\xi\right)$, it is possible to define the *power spectral density function* of $\xi\left(t\right)$ as:

$$\psi^2\left(\xi\right) = \int_0^\infty S_\xi(\omega)\, d\omega \tag{A.43}$$

Note that $S_\xi(\omega)$ represents the rate of change of the mean square value with frequency.
It can be demonstrated (see Simon [62] for all details) that the requirement of statistical independence stated at the beginning of this section implies that a white noise has a constant power spectral density function for all frequencies. Thus, since the power of an ergodic RP is defined as:

$$P_\xi = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_\xi(\omega)\, d\omega \tag{A.44}$$

it merges that continuous-time white noise is not something that occurs in real world because it would have infinite power (and mean square value). Nevertheless, it is a useful theoretical approximation in mathematical analyses of signals and systems.

A.2 DIFFERENTIAL GEOMETRY AND LIE DERIVATIVES

As linear algebra is essential in the study of linear control systems, differential geometry of manifolds is fundamental for the study of nonlinear ones [35].
This section contains an introduction to differential geometry, Lie derivatives and related concepts as analysis tool for nonlinear process control systems and serves mainly as a background for the nonlinear observability material that is discussed in section 2.3.
The reader interested in differential geometry of manifolds and Lie algebras is referred to introductory texts and review articles like Doyle and Henson [12], Kwatny and Blankenship [29] and Baldea and Daoutidis [5] for further insights.

A.2.1 Manifolds

A manifold, usually denoted with $\mathcal{M}$, is an open subset of $\mathbb{R}^n$, possibly equal to $\mathbb{R}^n$, which has special properties [36] that are useful for the results that follow. Most notably, a manifold is locally Euclidean.

Definition 19 (Locally Euclidean manifold)**.** A manifold $\mathcal{M}$ is called *locally Euclidean of dimension n* if every point $p \in \mathcal{M}$ has a neighborhood $\mathcal{U}$ such that there is a map $\phi : \mathcal{U} \to \mathbb{R}^n$ with the following properties:

i) ϕ is bijection.

ii) ϕ is continuous.

iii) ϕ^{-1} is continuous.

Loosely speaking, "locally Euclidean of dimension n" just means that, for every point p which lies on an n dimensional manifold $\mathcal{M}$ the points in the neighborhood $\mathcal{U}$ of p can be specified by using a set of n coordinates $(\phi_1(p), \ldots, \phi_n(p))$ in $\mathbb{R}^n$ called *set of local coordinates* of p in the *coordinate chart* $(\mathcal{U}, \phi)$.
In the remainder of this section the mapping ϕ and its inverse ϕ^{-1} are assumed to be C^∞ functions.

A.2.2 Vector fields

A mapping f which associates a point $x = (x_1, \ldots, x_n)$ on an open subset of $\mathbb{R}^n$ (like $\mathcal{M}$) with the vector:

$$f(x_1, \ldots, x_n) = \begin{bmatrix} f_1(x_1, \ldots, x_n) \\ f_2(x_1, \ldots, x_n) \\ \vdots \\ f_m(x_1, \ldots, x_n) \end{bmatrix} \tag{A.45}$$

in $\mathbb{R}^m$ takes the name of *vector mapping*. The notion of vector mapping is strictly related with the one of *vector field* introduced by the following definition.

Definition 20 (Vector field)**.** Let $T_p^{\mathcal{M}}$ be the space of tangent vectors to $\mathcal{M}$ at the point p. A *vector field* f on $\mathcal{M}$ is a vector mapping which assigns a tangent vector at p, $f(p) \in T_p^{\mathcal{M}}$, to any point $p \in \mathcal{M}$.

The vector field f is called *smooth* if its components are of class C^∞.
In the remainder of this section, only smooth vector fields will be considered.

Integral curve of a vector field

By definition, any vector field f can be written as a differential equation of the form:

$$\dot{x} = f(x) \tag{A.46}$$

The following definition is given.

Definition 21 (Integral curve)**.** An *integral curve* of a vector field f on $\mathcal{M}$ is a parametrized curve $p = \gamma(t)$, $t \in I \subset \mathbb{R}$ whose tangent vector at any point coincides with f at that point.

In local coordinates the image of an integral curve $x(t) = \phi(\gamma(t))$ satisfies the above differential equation:

$$\frac{dx(t)}{dt} = f(x) \tag{A.47}$$

Definition 22 (Maximal integral curve)**.** Let I denote an open interval of $\mathbb{R}$ containing the origin. Suppose $\gamma(t)$ is an integral curve of the vector field f such that $\gamma(0) = p$. The integral curve $\gamma(t)$ is said *maximal integral curve* if for any other integral curve $\hat{\gamma}(t)$ with $\hat{\gamma}(0) = p$, then $\hat{I} \subset I$ and $\hat{\gamma}(t) = \gamma(t)$ for $t \in \hat{I}$.

From theorems on existence and uniqueness of solutions of ordinary differential equations it follows that if f is of class C^k and $k \geqslant 1$, then there exist a unique maximal integral curve $\gamma(t)$ passing through the point $p \in \mathcal{M}$.

Flow of a vector field

Let f and $\Psi(t,p)$ be respectively a smooth vector field and the parametrized maximal integral curve through $p \in \mathbb{M}$ so that $\Psi : I \text{ x } \mathbb{M} \to \mathbb{M}$ and $\Psi(0,p) = p$. $\Psi(t,p)$ is called the *flow of the vector field f*.

The flow of a smooth vector field has some interesting properties:

i) $\Phi(0) = id$.

ii) $\Phi(t_2, \Phi(t_1,p)) = \Phi(t_1 + t_2, p)$.

iii) Satisfies the differential equation on $\mathcal{M}$:

$$\frac{d\Phi(t,p)}{dt} = f(\Phi(t,p))$$

As a consequence of (i) and (ii), it is clear that $\Phi(t,p)^{-1} = \Phi(-t,p)$.

If I coincide with $\mathbb{R}$ then the vector field f is called *complete* and the flow is well defined for all $t \in \mathbb{R}$ and $p \in \mathbb{M}$.

A.2.3 Lie derivatives

The *Lie derivative* of a scalar function $h(x) : \mathbb{R}^n \to \mathbb{R}$ along a vector field $f(x)$ measures how the function changes along the solutions of the differential equation associated with the vector field (i.e. the flow of f). Roughly speaking, it can be interpreted as the directional derivative of h along f.

In local coordinates the operator Lie derivative is given in the following manner:

$$L_f[h(x)] = \begin{bmatrix} \frac{\partial h(x)}{\partial x_1} \cdots \frac{\partial h(x)}{\partial x_n} \end{bmatrix} \begin{bmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{bmatrix} \tag{A.48}$$

Note that $L_f\,[h(x)]$ is itself a scalar function of x and that $L_f\,[h(x)] : \mathbb{R}^n \to \mathbb{R}$. Consequently, we can calculate its directional derivative along the vector field f, as:

$$L_f\,[L_f\,[h(x)]] = L_f^2\,[h(x)] \tag{A.49}$$

or along the vector field g, as:

$$L_g\,[L_f\,[h(x)]] = L_g L_f\,[h(x)] \tag{A.50}$$

with the latter representing the *mixed* Lie derivative of $L_f\,[h(x)]$ with respect to the vector field g.
Similarly, higher order Lie derivatives can be defined recursively as:

$$L_f^0\,[h(x)] = h(x) \tag{A.51}$$

$$L_f^i\,[h(x)] = L_f\left[L_f^{i-1}\,[h(x)]\right] = \frac{\partial L_f^{i-1}\,[h(x)]}{\partial x} f \qquad i = 1, 2, \dots \tag{A.52}$$

It is also possible to define $L_B h(x)$ as an $(m \text{ x } m)$-dimensional matrix of Lie derivatives of a vector function $h(x) : \mathbb{R}^n \to \mathbb{R}^m$ along the columns B^i, $i = 1, \dots, m$ of the matrix function $B(x) : \mathbb{R}^n \to \mathbb{R}^n \text{ x } \mathbb{R}^m$. $L_B\,[h(x)]$ is computed by multiplying the Jacobian of $h(x)$ and $B(x)$:

$$L_B\,[h(x)] = \begin{bmatrix} L_{B^1}\,[h_1(x)] & \dots & L_{B^m}\,[h_1(x)] \\ \vdots & \ddots & \vdots \\ L_{B^1}\,[h_m(x)] & \dots & L_{B^m}\,[h_m(x)] \end{bmatrix} = \begin{bmatrix} \frac{\partial h_1(x)}{\partial x_1} & \dots & \frac{\partial h_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m(x)}{\partial x_1} & \dots & \frac{\partial h_m(x)}{\partial x_n} \end{bmatrix} B(x) \tag{A.53}$$

A.3 FUNDAMENTALS OF NUMERICAL OPTIMIZATION

This section aims at providing some very basic concepts of non-convex optimization and a description of the main numerical algorithms employed in various parts of this work. There are several books which pursue the details of numerical optimization in greater depth than is presented here. Most notable are the texts by Nocedal and Wright [39] and Biegler [8].

A.3.1 Mathematical formulation

A mathematical programming problem can be formalized in general terms as follows:

$$\begin{aligned} &\min_{\chi \in \Omega} \varphi\,(\chi) && \textit{(objective function)} \\ \text{subject to: } &c\,(\chi) = 0 && \textit{(equality constraints)} \\ &\nu\,(\chi) \leqslant 0 && \textit{(inequality constraints)} \end{aligned}$$

where $\Omega = \{\chi \in \mathbb{R}^n \mid c\,(\chi) = 0,\ \nu\,(\chi) \leqslant 0\}$ is the *feasible set*.
When constraints are not present the optimization problem is defined *unconstrained* and $\Omega = \mathbb{R}^n$ otherwise it is defined *constrained* and $\Omega \subset \mathbb{R}^n$.
Let $\varphi : \mathbb{R}^n \to \mathbb{R}$, $c : \mathbb{R}^n \to \mathbb{R}^{n_c}$ and $\nu : \mathbb{R}^n \to \mathbb{R}^{n_\nu}$ to be twice continuously differentiable and consider a *feasible point* $\chi^* \in \Omega$. The following definitions are given.

Definition 23 (Global minimizer)**.** A point χ^* is a *global minimizer* if:

$$\varphi\,(\chi^*) \leqslant \varphi\,(\chi) \quad \forall\, \chi \in \Omega$$

Definition 24 (Local minimizer)**.** A point χ^* is a *local minimizer* if there is a neighborhood $\mathcal{M} \subseteq \Omega$ of χ^* such that:

$$\varphi\left(\chi^*\right) \leqslant \varphi\left(\chi\right) \quad \forall\, \chi \in \mathcal{M}$$

Since usually the knowledge of the objective function is only local, most algorithms are able to find only local minimizers.
In the unconstrained case, having assumed that the objective functions are twice continuously differentiable allows to formulate the following necessary and sufficient conditions of optimality to find out whether a point χ^* is a local minimum.

Theorem 6 (First-Order Necessary Conditions)**.** *If χ^* is a local minimizer, φ is continuously differentiable in an open neighborhood of χ^*, and constraints are not present, then $\nabla\varphi\left(\chi^*\right) = 0$.*

Theorem 7 (Second-Order Necessary Conditions)**.** *If χ^* is a local minimizer of φ, $\nabla^2\varphi$ exists and is continuous in an open neighborhood of χ^*, and constraints are not present, then $\nabla\varphi\left(\chi^*\right) = 0$ and $\nabla^2\varphi\left(\chi^*\right)$ is positive semidefinite.*

Theorem 8 (Second-Order Sufficient Conditions)**.** *Suppose that $\nabla^2\varphi$ is continuous in an open neighborhood of χ^*, $\nabla\varphi\left(\chi^*\right) = 0$, $\nabla^2\varphi\left(\chi^*\right)$ is positive definite, and constraints are not present. Then χ^* is a local minimizer of φ.*

In order to derive analogous conditions for the characterization of the solutions of constrained optimization problems there is the need to describe the feasible set in the neighborhood of χ^*. Indeed, it may be that not all the i^{th} inequality constraints are *active* (i.e. $\nu_i\left(\chi^*\right) = 0$) in such region. Let $\mathcal{A}\left(\chi^*\right)$ be the index set of all active inequality constraints in χ^* defined as follows:

$$\mathcal{A}\left(\chi^*\right) = \left\{i \in \{1, \dots, n_\nu\} \mid \nu_i\left(\chi^*\right) = 0\right\}$$

The characterization of such set is strictly related with the concept of *Linear Independence Constraint Qualification (LICQ)* introduced in the undermentioned definition.

Definition 25 (Linear Independence Constraint Qualification)**.** Given the point $\chi^* \in \Omega$ the LICQ is said to hold at χ^* if all vectors $\nabla c_i\left(\chi^*\right)$ for $i \in \{1, \dots, n_c\}$ and $\nabla \nu_i\left(\chi^*\right)$ for $i \in \mathcal{A}\left(\chi^*\right)$ are linearly independent.

It is now possible to define the equivalents of the first-order necessary conditions of optimality given above in the case of constrained optimization problems. Such conditions are often known as the *Karush-Kuhn-Tucker conditions*, or *KKT conditions* for short.

Definition 26 (Karush-Kuhn-Tucker optimality conditions)**.** If χ^* is a local minimizer and the LICQ holds at χ^*, then there are two vectors $\lambda^* \in \mathbb{R}^{n_c}$ and $\mu^* \in \mathbb{R}^{n_\nu}$ such that the following conditions are satisfied at $(\chi^*, \lambda^*, \mu^*)$:

$$\nabla_\chi \mathcal{L}\left(\chi^*, \lambda^*, \mu^*\right) = 0 \tag{A.54a}$$

$$c\left(\chi^*\right) = 0 \tag{A.54b}$$

$$\nu\left(\chi^*\right) \leqslant 0 \tag{A.54c}$$

$$\mu^* \geqslant 0 \tag{A.54d}$$

$$\mu_i^{*T} \nu_i\left(\chi^*\right) = 0 \qquad \text{for } i \in \{1, \dots, n_\nu\} \tag{A.54e}$$

where $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^{n_c} \times \mathbb{R}^{n_\nu} \to \mathbb{R}$ is the *Lagrangian function*, defined as:

$$\mathcal{L}\left(\chi, \lambda, \mu\right) = \varphi(\chi) + \lambda^T c\left(\chi\right) + \mu^T \nu\left(\chi\right)$$

while λ and μ are the *Lagrange multipliers vectors*.
A triple $(\chi^*, \lambda^*, \mu^*)$ satisfying KKT conditions and LICQ is called *KKT point*.

The conditions (A.54e) are called *complementarity conditions*.
A special case of complementarity is important in the remainder of this section and it is introduced in the following definition.

Definition 27 (Strict complementarity)**.** Given a KKT point $(\chi^*, \lambda^*, \mu^*)$ the strict complementarity conditions is said to hold at χ^* if and only if $\mu^* > 0$ for each active inequality constraint in χ^*.

Note that in case of strict complementarity of a KKT point $(\chi^*, \lambda^*, \mu^*)$, the objective function can be seen as subject to equality constraints only which collected define the following vector:

$$c^\bullet = \begin{bmatrix} c(\chi) \\ \nu_i(\chi) \text{ for } i \in \mathcal{A}(\chi^*) \end{bmatrix}$$

The second-order optimality conditions for constrained optimization problems can now be defined.

Definition 28 (Second-Order Necessary Conditions)**.** Consider a KKT point $(\chi^*, \lambda^*, \mu^*)$ and let strict complementarity hold in χ^*. If χ^* is a local minimizer, then:

$$s^T \nabla^2_\chi \mathcal{L}(\chi^*, \lambda^*, \mu^*) s \geqslant 0 \quad \text{for } s \in \text{null}\left(\left. \frac{\partial c^\bullet}{\partial \chi} \right|_{\chi=\chi^*} \right)$$

Definition 29 (Second-Order Sufficient Conditions)**.** Consider a KKT point $(\chi^*, \lambda^*, \mu^*)$, a basis s of the null-space of $\left. \frac{\partial c^\bullet}{\partial \chi} \right|_{\chi=\chi^*} \in \mathbb{R}^{n_{c^\bullet} \times n_\chi}$ and let strict complementarity hold in χ^*. If:

$$s^T \nabla^2_\chi \mathcal{L}(\chi^*, \lambda^*, \mu^*) s > 0$$

then χ^* is a local minimizer in its neighborhood.

In this section all the elements necessary to distinguish a local minimizer have been introduced; in the remainder of this chapter the optimization algorithms most frequently used in this work for the resolution of both constrained and unconstrained optimization problems will be presented.

A.3.2 Interior point algorithm

Consider the following equivalent form of the general constrained optimization problem described in the previous section:

$$\begin{aligned} &\min_{\chi, s^\bullet} \varphi(\chi) \\ \text{subject to: } & c_E(\chi) = 0 \\ & c_I(\chi) - s^\bullet = 0 \\ & s^\bullet \geqslant 0 \end{aligned} \tag{A.55}$$

where the vector $c_I(\chi) \in \mathbb{R}^{n_I}$ is formed from scalar function and similarly $c_E(\chi) \in \mathbb{R}^{n_E}$. Note that the introduction of a vector $s^\bullet$ of slack variables allows to transform the inequalities $c_I \geqslant 0$ into equalities.

The KKT conditions for the nonlinear program (A.55) can be written as:

$$\nabla \varphi(\chi) - J_E^T(\chi)\lambda - J_I^T(\chi)\mu = 0 \tag{A.56a}$$

$$S\mu - \tau e = 0 \tag{A.56b}$$

$$c_E(\chi) = 0 \tag{A.56c}$$

$$c_I(\chi) - s^\bullet = 0 \tag{A.56d}$$

in which $\tau = 0$, together with:

$$s^\bullet \geqslant 0, \quad \mu \geqslant 0 \tag{A.57}$$

Here J_E and J_I are the Jacobian matrices of the functions c_E and c_I, respectively, λ and μ are their Lagrange multipliers, S is a diagonal matrix whose diagonal entries are given by the elements of the vector $s^\bullet$ and $e = (1, \dots, 1)^T$.

Consider now the following optimization problem known with the name of *barrier problembarrier problem*:

$$\min_{\chi, s^\bullet} \varphi(\chi) - \tau \sum_{i=1}^{n_I} \log(s_i^\bullet) \tag{A.58}$$

$$\text{subject to: } c_E(\chi) = 0$$
$$c_I(\chi) - s^\bullet = 0$$

where τ is a positive parameter and $\log(\cdot)$ denotes the natural logarithm function.

By comparing objective functions (A.55) and (A.58) and considering that the KKT conditions of the barrier problem have the same form of those of (A.56) but are characterized by a positive value of τ, it follows that the solution of the barrier problem does not coincide with that of (A.55) until $\tau = 0$.

IPOPT and more in general all the interior-point methods try to find an approximate solution of the barrier problem for a sequence of positive *barrier parameters* $\{\tau_k\}$ that converges to zero. Indeed, in this way the final solution found by the algorithm, in the limit, is equivalent to the solution of the optimization problem (A.55).

Such solution must satisfy the KKT conditions (A.56) with $\tau > 0$ for some vectors $s^\bullet$, λ and μ. These conditions form a nonlinear algebraic system in the unknowns χ, $s^\bullet$, λ and μ that is solved in interior-point algorithms via Newton-like methods.

Applying Newton's method to the nonlinear system (A.56) yields:

$$\begin{bmatrix} \nabla^2_{\chi\chi}\mathcal{L} & 0 & -J_E^T(\chi) & -J_I^T(\chi) \\ 0 & Z & 0 & S \\ J_E(\chi) & 0 & 0 & 0 \\ J_I(\chi) & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} p_\chi \\ p_{s^\bullet} \\ p_\lambda \\ p_\mu \end{bmatrix} = - \begin{bmatrix} \nabla \varphi(\chi) - J_E^T(\chi)\lambda - J_I^T(\chi)\mu \\ S\mu - \tau e \\ c_E(\chi) \\ c_I(\chi) - s^\bullet \end{bmatrix} \tag{A.59}$$

where Z is a diagonal matrix whose diagonal entries are given by the elements of the vector μ and $\mathcal{L}$ denotes the Lagrangian for (A.55):

$$\mathcal{L}(\chi, s^\bullet, \lambda, \mu) = \varphi(\chi) - \lambda^T c_E(\chi) - \mu^T(c_E(\chi) - s^\bullet) \tag{A.60}$$

Then after the step $p = (p_\chi, p_{s^\bullet}, p_\lambda, p_\mu)$ have been determined, the new iterates are computed as:

$$\chi^+ = \chi + \alpha_{s^\bullet} p_\chi \tag{A.61}$$

$$s^{\bullet,+} = s^\bullet + \alpha_{s^\bullet} p_{s^\bullet} \tag{A.62}$$

$$\lambda^+ = \lambda + \alpha_\mu p_\lambda \tag{A.63}$$

$$\mu^+ = \mu + \alpha_\mu p_\mu \tag{A.64}$$

$$\tag{A.65}$$

in which $\alpha_{s^\bullet} \in (0, 1]$ and $\alpha_\lambda \in (0, 1]$.

Denoted by $E(\chi, s^\bullet, \lambda, \mu; \tau)$ the maximum absolute error of the KKT equations, defined for some vector norm $\|\cdot\|$ as:

$$\begin{aligned} E(\chi, s^\bullet, \lambda, \mu; \tau) = \max\{&\|\nabla\varphi(\chi) - J_E^T(\chi)\lambda - J_I^T(\chi)\mu\|, \|S\mu - \tau e\|, \|c_E(\chi)\|, \\ &\|c_I(\chi) - s^\bullet\|\} \end{aligned} \tag{A.66}$$

the basic algorithm implemented in IPOPT is summarized in Algorithm 3 (in which j is the index of the outer loop, k is the index of the inner loop, and $\varepsilon > 0$ is an user-defined convergence tolerance).

Algorithm 3 IPOPT Algorithm

```
1: Initialize the method by supplying the values of $\tau_0, \alpha_r, \lambda_0, \mu_0$ and form the initial nonlinear system (A.59) accordingly. Set: $j = 0$, $k = 0$.
2: Given the current iterates $\chi_k$, $s_k^\bullet$, $\lambda_k$ and $\mu_k$ determine the Newton step $p$ and then compute the new iterates as:
       $\chi^+ = \chi + \alpha_{s^\bullet} p_\chi$
       $s^{\bullet,+} = s^\bullet + \alpha_{s^\bullet} p_{s^\bullet}$
       $\lambda^+ = \lambda + \alpha_\mu p_\lambda$
       $\mu^+ = \mu + \alpha_\mu p_\mu$
3: If $E_0(\chi_{k+1}, s_{k+1}^\bullet, \lambda_{k+1}, \mu_{k+1}; \tau_{k+1}) \leqslant \varepsilon$ stop. Otherwise, go to Step 4.
4: If $E_j(\chi_{k+1}, s_{k+1}^\bullet, \lambda_{k+1}, \mu_{k+1}; \tau_{k+1}) \leqslant \kappa\tau_j$ (for some $\kappa > 0$) go to Step 5. Otherwise, update $k \leftarrow k + 1$ and go to Step 2.
5: Set $\tau_{j+1} = \frac{\tau_j}{\iota}$ (for some $\iota > 0$), update $j \leftarrow j + 1$, $k \leftarrow k + 1$ and go to Step 2.
```

A.3.3 Nelder-Mead algorithm

The Nelder-Mead algorithm is one of the most known derivative-free multidimensional unconstrained optimizer.

In order to explain how it works the following definition must be introduced.

Definition 30 (Simplex)**.** Let $\chi_1, \dots, \chi_{n+1} \in \mathbb{R}^n$ be $n + 1$ affinely independent points. The simplex determined by them is the set of points

$$S = \left\{ \Upsilon \in \mathbb{R}^n | \ \Upsilon = \sum_{i=1}^{n+1} a_i \chi_i, \ a_i \geqslant 0, \ \sum_{i=1}^{n+1} a_i = 1 \right\} \tag{A.67}$$

The method begins by generating a set of $n+1$ points around the point of initialization $\chi_{init} \in \mathbb{R}^n$ and considers them as vertices of an initial working simplex S_0. The initial working simplex has to be *nondegenerate*, i.e. the $n+1$ points generated must not lie in the same hyperplane.
Then, at each iteration the method performs a sequence of transformation of the current working simplex S_k aimed at decreasing the function values at its vertices. In particular, the algorithm keep track of the $n+1$ vertices of S_k and sort them on the base of their corresponding objective function value:

$$\varphi(\chi_1) \leqslant \cdots \leqslant \varphi(\chi_{n+1}) \tag{A.68}$$

Then, only the worst vertex χ_{n+1} is replaced with a new point of the form:

$$\chi(\alpha) = (1+\alpha)\chi_c - \alpha\chi_{n+1} \tag{A.69}$$

where χ_c is the *centroid* of the n remaining points, defined as:

$$\chi_c = \frac{1}{n}\sum_{j=1}^{n}\chi_j \tag{A.70}$$

Thus, the idea is to make a *reflection* of the worst vertex χ_{n+1} with respect to χ_c, where $\alpha \geqslant 0$ is a properly chosen *reflection coefficient*. Typically, α is selected from a sequence of the following type:

$$-1 \leqslant \alpha_{ic} < 0 < \alpha_{oc} < \alpha_r < \alpha_e \tag{A.71}$$

A common choice is:

$$\{\alpha_r, \alpha_e, \alpha_{oc}, \alpha_{ic}\} = \left\{1, 2, \frac{1}{2}, -\frac{1}{2}\right\} \tag{A.72}$$

The effects of different choices of α on the shape of the simplex having χ_1, χ_2 and χ_3 as vertices are shown in Figure 27.

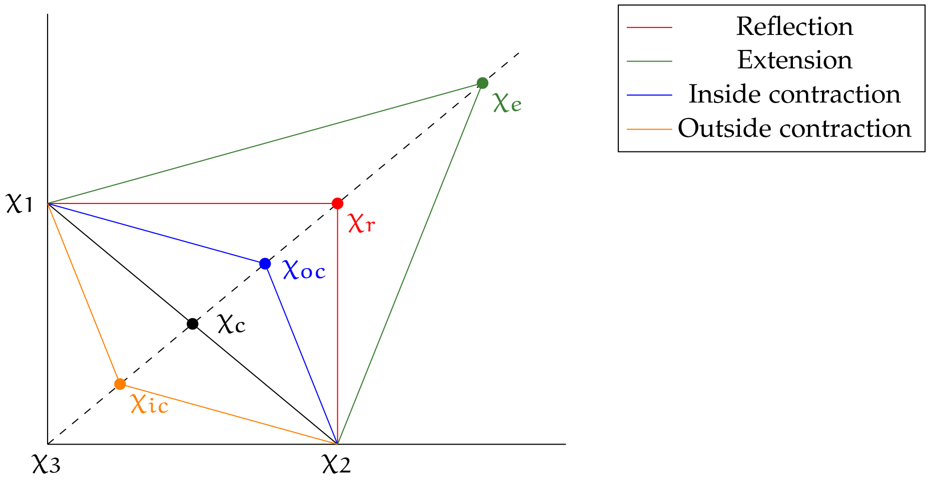


Figure 27: Comparison of different simplex transformations: $\varphi(\chi_3) \geqslant \varphi(\chi_2) \geqslant \varphi(\chi_1)$.

A practical implementation of the Nelder-Mead method must include a test that ensures termination in a finite amount of time. For simplicity, assume that the algorithm computes the boolean value $\mathtt{break}$ which becomes true when it is time to stop the iterations. Quite generally, $\mathtt{break}$ is composed of three different parts: $\mathtt{break}_\chi$, $\mathtt{break}_f$ and $\mathtt{break}_{fail}$.
$\mathtt{break}_\chi$ is the domain convergence or termination test and becomes true when some or all

vertices of the working simplex S are close enough in some sense.
Conversely, $\mathtt{break}_f$ is the function value convergence test which becomes true when some or all function values $\varphi\left(\chi_j\right)$ are close enough in some sense.
Finally, $\mathtt{break}_{fail}$ is the no-convergence in time test and becomes true if the number of iteration or function evaluations exceeds some prescribed maximum allowed value.
The algorithm terminates as soon as at least one of these tests becomes true.
The general algorithm is summarized in the Algorithm 4.

Algorithm 4 Nelder-Mead Algorithm

```
1: Initialize the method by supplying the values of $\chi_{init}, \alpha_r, \alpha_e, \alpha_{oc}$ and $\alpha_{ic}$ and form the
   initial working simplex $S_{init}$ accordingly. Set: $k = 0$.
2: Sort the vertices of the working simplex $S_k$ in such a way as to have:
       $\varphi(\chi_1) \leqslant \cdots \leqslant \varphi(\chi_{n+1})$
3: If break is true stop, otherwise compute:
       $\chi_c = \frac{1}{n}\sum_{j=1}^{n}\chi_j$
       $\chi(\alpha_r) = (1+\alpha_r)\chi_c - \alpha_r\chi_{n+1}$
       $\varphi_r = \varphi(\chi(\alpha_r))$
4: Perform the transformation of the working simplex $S_k$.
   REFLECTION: If $\varphi(\chi_1) \leqslant \varphi_r \leqslant \varphi(\chi_n)$ replace $\chi_{n+1}$ with $\chi(\alpha_r)$, update $k \leftarrow k+1$ and
       go to Step 2.
   EXPANSION: If $\varphi_r < \varphi(\chi_1)$ compute:
       $\varphi_e = \varphi(\chi(\alpha_e))$
       and if $\varphi_e < \varphi_r$ replace $\chi_{n+1}$ with $\chi(\alpha_e)$; otherwise replace $\chi_{n+1}$ with $\chi(\alpha_r)$,
       update $k \leftarrow k+1$ and go to Step 2.
   OUTSIDE CONTRACTION: If $\varphi(\chi_n) \leqslant \varphi_r \leqslant \varphi(\chi_{n+1})$ compute:
       $\varphi_{oc} = \varphi(\chi(\alpha_{oc}))$
       and if $\varphi_{oc} < \varphi_r$ replace $\chi_{n+1}$ with $\chi(\alpha_{oc})$, update $k \leftarrow k+1$ and go to Step 2;
       otherwise go to Step 5.
   INSIDE CONTRACTION: If $\varphi_r \leqslant \varphi(\chi_{n+1})$ compute:
       $\varphi_{ic} = \varphi(\chi(\alpha_{ic}))$
       and if $\varphi_{ic} < \varphi_{n+1}$ replace $\chi_{n+1}$ with $\chi(\alpha_{ic})$, update $k \leftarrow k+1$ and go to Step 2;
       otherwise go to Step 5.
5: Set:
       $\chi_i = \chi_1 - \frac{(\chi_i - 1)}{2} \quad \forall\, 2 \leqslant i \leqslant n+1$
   compute $\varphi(\chi_i)$, update $k \leftarrow k+1$ and go to Step 2.
```

From the scheme results clear that the idea behind the algorithm is to try to expand the simplex if good values of the objective function are found and to contract it otherwise.
As very little is known about the convergence properties of the method (with mainly negative results), the Nelder-Mead algorithm is commonly considered an heuristic technique.

However, despite of this it has reveal itself very efficient in practice, particularly for the solution of problems with less than ten degrees of freedom [50].
For this reason it is commonly implemented in various standard optimization libraries.

B PYTHON CODES

B.1 EVALUATION OF THE DEGREE OF OBSERVABILITY

```
"
SUMMARY:
This function computes the observability gramian\covariance matrix Wo for
stable dynamical systems by making use of data collected along system
trajectories. It applies initial condition perturbations in each state to
compute the empirical observability gramian or the observability covariance
matrix.
Moreover, it computes the magnitude of the smallest eigenvalue for Wo useful
to represent observability of a nonliner system over an operating region.

SYNTAX:
assignment = W(f, h, t_par, Cm, xss, uss, flag)

ARGUMENTS:
f       - SXFunction having state, x, and control, u, symbolic variables
          as inputs and state derivative symbolic variable as outputs;
            signature: f = SXFunction([x,u],[dx/dt])
where:
* x = SX.sym(x,state size)
* u = SX.sym(u,control size)
* dx/dt = \
           vertcat([1st eq., \
                    2nd eq., \
                    ......., \
                    nth eq.])

h       - SXFunction having state, x, symbolic variables
          as inputs and measurement symbolic variable as outputs;
            signature: h = SXFunction([x,u],[h(x)])
where:
* x = SX.sym(x,state size)
* u = SX.sym(u,control size)
* h(x) = \
         vertcat([1st eq., \
                  2nd eq., \
                  ......., \
                  nth eq.])

t_par   - time discretization;
            signature: t_par = np.array([start,stop,step])
```

```
Cm      - excitation sizes for each excitation direction;
            signature: Cm = [Cm_1,...,Cm_n]
xss     - steady-state operating point;
            signature: xss = [x1_ss,...,xn_ss]
uss     - steady-state value of the input;
            signature: uss = [u1_ss,...,un_ss]
flag    - gramian\covariance matrix type;
            signature:
            ∗ ’obs_cov’   : observability covariance matrix
            ∗ ’obs_gram’  : empirical observability gramian
"

from casadi import∗
from casadi.tools import∗
import numpy


def W(f, h, t_par, Cm, xss, uss, flag):

    # redefine some variables for clarity of notation

    # run-time parameters
    t_start = t_par[0] # simulation start time
    t_end = t_par[1]   # simulation end time:
    # be aware that the system will have reached equilibrium
    #for some time tf < t_end
    delta_t = t_par[2]  # sample length
    t_npti = int(np.floor((t_end-t_start)/delta_t)+1) # # of discr. pts
    time = np.linspace(t_start, t_end, t_npti) # time vector

    # get vectors size
    p = f.inputExpr(1).shape[0] # # of inputs
    n = f.inputExpr(0).shape[0] # # of states
    k = h.outputExpr(0).shape[0] # # of outputs

    # get initial steady state value of the output
    h.setInput(xss)
    h.evaluate()
    yss = h.getOutput().toArray().squeeze()

    # scaling transformation matrices calculation
    Tx = diag(xss)
    Tu = diag(uss)

    # scale the states with respect to their steady state values
    x_scal = mul(Tx,f.inputExpr(0))
    u_scal = mul(Tu,f.inputExpr(1))
    f_scal = mul(inv(Tx),f([x_scal,u_scal])[0])

    # computational settings (part 1)
    r = 2 # # of matrices for excitation directions (DON'T CHANGE)
    cm = Cm # excitation sizes for each direction
    s = len(cm) # # of different excitation sizes for each direction

    # set the whole series of sundials options available for the user
    opts = {}
    opts_b = {}
    opts["fsens_err_con"] = True
    opts_b["fsens_err_con"] = True
    opts["quad_err_con"] = True
    opts_b["quad_err_con"] = True
    opts["abstol"] = 1e-6
    opts_b["abstol"] = 1e-6
    opts["reltol"] = 1e-6
    opts_b["reltol"] = 1e-6
    opts["t0"] = t_start
    opts_b["t0"] = t_start
    opts["tf"] = t_end
    opts_b["tf"] = t_start+delta_t

    # initialization (part 1)
    w_o = DMatrix.zeros(n,n)
    x_init = DMatrix.ones(n)
    u_d = DMatrix.ones(p)

    # computational settings (part 2)
    T = DMatrix.zeros(n,2*n)  # matrices for excitation
    T[0:n,0:n] = DMatrix.eye(n)
    T[0:n,n:2*n] = -DMatrix.eye(n)
    e = DMatrix.eye(n) # matrix filled with standard unit vectors

    for l in range(r):

        for m in range(s):

        # initialization (part 2)
        chsi = DMatrix.zeros(n, n)
        z = DMatrix.zeros(n, t_npti*k)

            for i in range(n):

            # compute perturbed initial condition
            x_pert = x_init + cm[m]*mul(T[0:n,l*n:n*(l+1)],e[:,i])

            # set input on its scaled steady state value
            u = u_d

            # Manipulate the function to adhere to the integrator
            #input/output signature f(time;states;parameters)
            f_ode = SXFunction("ODE_right_hand_side",\
            daeIn(x=f.inputExpr(0),\
            p=f.inputExpr(1)),\
            daeOut(ode=f_scal))

            # define the integrator
            integrator = Integrator("integrator","cvodes", f_ode, opts)
            integrator.setInput(x_pert, "x0")
            integrator.setInput(u_d, "p")
            integrator.evaluate()
            integrator.reset()

            #define a convenient function to acces x(t)
            def out(t):
                integrator.integrate(t)
                return integrator.getOutput().toArray()
            x = np.array([out(t) for t in time]).squeeze()

            # compute the output of the system corresponding to the
            #perturbed  initial condition
            y = DMatrix.zeros(k,x.shape[0])

            for index in range (x.shape[0]):
                x_star = x[index,:]
                h.setInput(x_star)
                h.evaluate()
                y[:,index] = (h.getOutput().toArray()).squeeze()

            y = y.T

            for iii in range(k):

                # compute difference between the output of the system
                #corresponding to the perturbed initial condition and
                #the steady state that the output will reach after per-
                #turbation
                if flag == 'obs_gram':

                    # compute the steady state that the output will reach
                    #after perturbation
                    yss_fin = yss

                    if k == 1:

                        dev = (y[:,iii] - DMatrix.ones(len(time),1))*\
                               yss_fin

                    else:
```

```

              dev = (y[:,iii] - DMatrix.ones(len(time),1))*\
              yss_fin[iii]

          elif flag == 'obs_cov':

            # compute the steady state that the output will reach
            #after perturbation
            yss_fin = (y[len(time)-1,:]).T

            if k == 1:

              dev = (y[:,iii] - DMatrix.ones(len(time),1)*\
                 yss_fin)*yss

            else:

              dev = (y[:,iii] - DMatrix.ones(len(time),1)*\
              yss_fin[iii])*yss[iii]


          z[i,t_npti*iii:t_npti*(iii+1)] = dev.T

        #chsi matrix calculation
        chsi = mul(z, z.T)

        # observability covariance matrix\empirical grammian calcula-
        #tion
        chsi_TT = mul(chsi,(T[0:n,l*n:n*(l+1)]).T)
        w_o = w_o + (1/(r*s*(cm[m]**2)))*delta_t*mul(T[0:n,l*n:n*(l+1)],\
        chsi_TT)

    # symmetrization
    Wo = 0.5*(w_o+w_o.T)

    # compute the magnitude of the smallest and biggest eigenvalues for Wo
    lambda_min = np.amin(np.array(np.linalg.eigvalsh(Wo)))

    return Wo, lambda_min
```

Listing 1: Evaluation of the degree of observability

B.2 VERIFICATION OF THE KALMAN RANK CONDITION

```
"
SUMMARY:
This function computes the observability matrix and performs the Kalman
rank test for a given linear or nonlinear system.

SYNTAX:
assignment = KRC(f, h, xss, uss, t_steps)

ARGUMENTS:
f       - SXFunction having state, x, and control, u, symbolic variables
          as inputs and state derivative symbolic variable as outputs;
            signature: f = SXFunction([x,u],[dx/dt])
where:
* x = SX.sym(x,state size)
* u = SX.sym(u,control size)
* f = \
      vertcat([1st eq., \
               2nd eq., \
               ......., \
               nth eq.])

h       - Expression having state, x, symbolic variables
          as inputs and measurement symbolic variable as outputs;
            signature: h = \
                           vertcat([1st eq., \
                                    2nd eq., \
                                    ......., \
                                    nth eq.])

xss     - steady-state operating point;
            signature: xss = [x1_ss,...,xn_ss]
uss     - steady-state value of the input;
            signature: uss = [u1_ss,...,un_ss]
t_steps - discretization parameters;
            signature: [t_steps_1, t_steps_2]
where:
* t_steps_1 = # of time interval in each sampling interval.
* t_steps_2 = sampling interval for the integrator.
"


from casadi import *
from casadi.tools import *
import numpy


def KRC(f, h, xss, uss, t_steps):

    # get vectors size
    ny = h.shape[0] # # of outputs
    nx = xss.shape[0] # # of states

    # create the simple RK4 integrator for propagating the system model
    # dynamics
    Faug_model = simpleRK(f, t_steps[0])
    Faug_model.init()
    [X_next] = Faug_model([f.inputExpr(0), f.inputExpr(1), t_steps[1]])


    # get the augmented system Jacobian functions
    Adummy = jacobian(X_next,f.inputExpr(0))
    A = SXFunction([f.inputExpr(0),f.inputExpr(1)], [Adummy])
    A.init()
    Cdummy = jacobian(h,f.inputExpr(0))
    C = SXFunction([f.inputExpr(0)], [Cdummy])
    C.init()

    # build the observability matrix
    [A_lin] = A([xss,uss])
    [C_lin] = C([xss])
    O_rc = DMatrix.zeros(ny*nx, nx)

    for i in range(nx):
        O_rc[ny*i:(i+1)*ny,:] = mul(C_lin,np.linalg.matrix_power(A_lin,i))

    # perform the Kalman rank test
    if np.linalg.matrix_rank(O_rc) == nx:

        flag = ’observable’

    else:

        flag = ’unobservable’

    return O_rc, flag
```

Listing 2: Verification of the Kalman rank condition

B.3 OPTIMIZATION OF THE STATE ESTIMATOR PERFORMANCES

```
"
SUMMARY:
This function optimizes the state estimator performances with respect to
a given optimality criterion acting on the diagonal elements of the
process noise covariance matrix.
The optimization consists in an exhaustive search followed by a certain
number of Nelder-Mead algorithm iterations.

SYNTAX:
assignment = opt_se(fobj_bf, fobj_nm, maxfev, grid)

ARGUMENTS:
fobj_bf - Objective function to be minimized with the exhaustive search;
            signature: fobj_bf = f(x, *args)
fobj_nm - Objective function to be minimized with the Nelder-Mead algorithm;
            signature: fobj_nm = f(x, *args)
where:
* x   : argument in the form of a 1-D array.
* args: tuple of any additional fixed parameters needed to completely
        specify the function.

maxfev  - Maximum number of function evaluations to be performed with
          the Nelder-Mead algorithm;
            signature: maxfev = #
grid    - evaluation grid;
            signature: grid = slice(lower bound, upper bound)
"

from casadi import*
from casadi.tools import*
import scipy.optimize
import numpy

def opt_se(fobj_bf, fobj_nm, maxfev, grid)

    # call the global optimizer
    q_bf = scopt.brute(fobj_bf, grid, full_output=True, finish=None)

    # improve q_bf by performing a certain number of Nelder-Maed iterations
    q_opt = scipy.optimize.minimize(fobj_nm, np.sqrt(q_bf),\
                                    method="Nelder-Mead",\
                                    options={"maxfev":maxfev})

    return q_opt
```

Listing 3: Optimization of the state estimator performances

BIBLIOGRAPHY

[1] P. R. Amestoy, I. S. Duff, and J. Y. L'Excellent. "Multifrontal parallel distributed symmetric and unsymmetric solvers". In: *Computer methods in applied mechanics and engineering* 184.2 (2000), pp. 501–520 (Cited on page 79).

[2] J. Andersson. "A General-Purpose Software Framework for Dynamic Optimization". PhD thesis. Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium: Arenberg Doctoral School, KU Leuven, Oct. 2013 (Cited on page 59).

[3] M. Anguelova. "Nonlinear observability and identifiability: General theory and a case study of a kinetic model for S. Cerevisiae". In: (2004).

[4] D. M. Asmar and G. J. Eslinger. *Nonlinear programming approach to filter tuning*. 2012 (Cited on page 51).

[5] M. Baldea and P. Daoutidis. *Dynamics and Nonlinear Control of Integrated Process Systems*. Cambridge Series in Chemical Engineering. Cambridge University Press, 2012. ISBN: 9780521191708 (Cited on page 87).

[6] M. Bauer and I. K. Craig. "Economic assessment of advanced process control–a survey and framework". In: *Journal of Process Control* 18.1 (2008), pp. 2–18 (Cited on page 3).

[7] A. Bicchi. *Lecture notes for the course "Robotics II"*. 2015. URL: http://www.centropiaggio.unipi.it/sites/default/files/rob2.pdf.

[8] L.T. Biegler. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2010 (Cited on page 89).

[9] J. Choi. *Lecture notes for the course "Linear Systems and Control"*. 2010. URL: http://www.egr.msu.edu/classes/me851/jchoi/lecture/Lect_14.pdf.

[10] C. R. Cutler and B. L. Ramaker. "Dynamic matrix control ? A computer control algorithm". In: *Joint automatic control conference*. 17. 1980, p. 72 (Cited on page 40).

[11] E. J. Davison and H. W. Smith. "Pole assignment in linear time-invariant multivariable systems with constant disturbances". In: *Automatica* 7.4 (1971), pp. 489–498 (Cited on page 32).

[12] Francis J. Doyle III and Michael A. Henson. "Nonlinear Process Control". In: ed. by M. A. Henson and D. E. Seborg. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997. Chap. Nonlinear Systems Theory, pp. 111–147. ISBN: 0-13-625179-X (Cited on page 87).

[13] C. Elster and A. Neumaier. "A trust-region method for the optimization of noisy functions". In: *Computing* 58 (1997), pp. 31–46 (Cited on page 79).

[14] R. Findeisen and F. Allgöwer. "An introduction to nonlinear model predictive control". In: *21st Benelux Meeting on Systems and Control*. Vol. 11. 2002, pp. 119–141 (Cited on page 79).

[15] B. A. Francis and W. M. Wonham. “The internal model principle of control theory”. In: *Automatica* 12.5 (1976), pp. 457–465 (Cited on page 32).

[16] M. Gabiccini et al. “A Computational Framework for Environment-Aware Robotic Manipulation Planning”. In: ().

[17] J. Hahn and T. F. Edgar. “An improved method for nonlinear model reduction using balancing of empirical gramians”. In: *Computers & chemical engineering* 26.10 (2002), pp. 1379–1397 (Cited on page 28).

[18] J. Hahn and T. F. Edgar. “Balancing approach to minimal realization and model reduction of stable nonlinear systems”. In: *Industrial & engineering chemistry research* 41.9 (2002), pp. 2204–2212.

[19] T. F. Hahn J. Edgar and W. Marquardt. “Controllability and observability covariance matrices for the analysis and order reduction of stable nonlinear systems”. In: *Journal of Process Control* 13.2 (2003), pp. 115–127.

[20] D. W. Harder and R. Khoury. *Numerical Analysis for Engineering*. 2010. URL: https://ece.uwaterloo.ca/~dwharder/NumericalAnalysis/ (visited on 03/22/2016) (Cited on page 52).

[21] R. Hermann and A. J. Krener. “Nonlinear controllability and observability”. In: *Automatic Control, IEEE Transactions on* 22.5 (Oct. 1977), pp. 728–740.

[22] *HSL. A collection of Fortran codes for large scale scientific computation*. 2016. URL: http://www.hsl.rl.ac.uk/ (Cited on page 79).

[23] R. Huang, L. T. Biegler, and S. C. Patwardhan. “Fast offset-free nonlinear model predictive control based on moving horizon estimation”. In: *Industrial & Engineering Chemistry Research* 49.17 (2010), pp. 7882–7890 (Cited on page 41).

[24] H. Huijberts and H. Nijmeijer. “Controllability and Observability of Nonlinear Systems”. English. In: *Control Systems, Robotics, and Automation*. Vol. 12. Encyclopedia of Life Support Systems (EOLSS). Paris, France: Eolss Publishers. URL: http://www.eolss.net (Cited on page 19).

[25] S. J. Julier and J. K. Uhlmann. *New extension of the Kalman filter to nonlinear systems*. International Society for Optics and Photonics, 1997 (Cited on page 80).

[26] R. E. Kalman. “A new approach to linear filtering and prediction problems”. In: *Journal of basic Engineering* 82.1 (1960), pp. 35–45 (Cited on page 12).

[27] A. Klenke. *Probability Theory: A Comprehensive Course*. Springer, 2013. ISBN: 9781447153627 (Cited on page 81).

[28] H. Kwakernaak and R. Sivan. *Linear optimal control systems*. Vol. 1. Wiley-interscience New York, 1972 (Cited on page 32).

[29] H.G. Kwatny and G. Blankenship. *Nonlinear Control and Analytical Mechanics: A Computational Approach*. Control Engineering. Birkhäuser Boston, 2000. ISBN: 9780817641474 (Cited on page 87).

[30] S. Lall, J. E. Marsden, and S. Glavaški. “A subspace approach to balanced truncation for model reduction of nonlinear control systems”. In: *International journal of robust and nonlinear control* 12.6 (2002), pp. 519–535 (Cited on page 29).

[31] U. Maeder, F. Borrelli, and M. Morari. “Linear offset-free model predictive control”. In: *Automatica* 45.10 (2009), pp. 2214–2222 (Cited on page 41).

[32] M. Mangold. *Lecture notes for the course "State Estimation"*. 2015. URL: http://ifatwww.et.uni-magdeburg.de/~mangold/mod_mess/downloads/Nonlinear_Observability.pdf.

[33] T. E. Marlin. *Process control: designing processes and control systems for dynamic performance*. McGraw-Hill, 1995.

[34] E. S. Meadows and J. B. Rawlings. "Model predictive control". In: *Nonlinear Process Control*. Ed. by M. A. Henson and D. E. Seborg. Prentice Hall, 1997. Chap. 5, pp. 233–310 (Cited on page 41).

[35] S. Mehta and J. Chiasson. "Nonlinear control of a series DC motor: theory and experiment". In: *Industrial Electronics, IEEE Transactions on* 45.1 (1998), pp. 134–141 (Cited on page 87).

[36] J.W. Milnor. *Topology from the Differentiable Viewpoint*. Princeton Landmarks in Mathematics. Princeton University Press, 1997. ISBN: 9780691048338 (Cited on page 87).

[37] M. Morari and U. Maeder. "Nonlinear offset-free model predictive control". In: *Automatica* 48.9 (2012), pp. 2059–2067 (Cited on pages 33, 41, 78).

[38] K. R. Muske and T. F. Edgar. "Nonlinear Process Control". In: ed. by M. A. Henson and D. E. Seborg. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997. Chap. Nonlinear State Estimation, pp. 311–370. ISBN: 0-13-625179-X.

[39] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006. ISBN: 9780387400655 (Cited on page 89).

[40] B. J. Odelson, M. R. Rajamani, and J. B. Rawlings. "A new autocovariance least-squares method for estimating noise covariances". In: *Automatica* 42.2 (2006), pp. 303–308 (Cited on page 19).

[41] B. A. Ogunnaike and W. H. Ray. *Process dynamics, modeling, and control*. Vol. 1. Oxford University Press New York, 1994.

[42] G. Pannocchia. "Offset-free tracking MPC: A tutorial review and comparison of different formulations". In: *Control Conference (ECC), 2015 European*. IEEE. 2015, pp. 527–532 (Cited on pages 5, 41, 68).

[43] G. Pannocchia. "Strategies for Robust Multivariable Model-Based Control". PhD thesis. University of Pisa, 2002.

[44] G. Pannocchia and A. Bemporad. "Combined design of disturbance model and observer for offset-free model predictive control". In: *Automatic Control, IEEE Transactions on* 52.6 (2007), pp. 1048–1053 (Cited on page 41).

[45] G. Pannocchia, A. De Luca, and M. Bottai. "Prediction Error Based Performance Monitoring, Degradation Diagnosis and Remedies in Offset-Free MPC: Theory and Applications". In: *Asian Journal of Control* 16.4 (2014), pp. 995–1005 (Cited on page 19).

[46] G. Pannocchia, M. Gabiccini, and A. Artoni. "Offset-free MPC explained: novelties, subtleties, and applications". In: *IFAC-PapersOnLine* 48.23 (2015), pp. 342–351 (Cited on pages 40, 41).

[47] G. Pannocchia and J. B. Rawlings. "Disturbance models for offset-free model-predictive control". In: *AIChE journal* 49.2 (2003), pp. 426–437 (Cited on pages 33, 40, 41, 53, 68).

[48] G. Pannocchia and J. B. Rawlings. "Robustness of MPC and disturbance models for multivariable ill-conditioned processes". In: *Department of Chemical Engineering, University of Wisconsin-Madison, Madison, WI* 53706 (2001) (Cited on page 33).

[49] G. Pannocchia and J. B. Rawlings. "The velocity algorithm LQR: a survey". In: *TWMCC, Department of Chemical Engineering, University of Wisconsin-Madison, Tech. Rep* 1 (2001) (Cited on page 41).

[50] V. Piccialli. *Lecture notes for the course "Continuous Optimization"*. 2006. URL: http://www.dis.uniroma1.it/~piccialli/derfree.pdf (Cited on page 96).

[51] T. D. Powell. "Automated tuning of an extended Kalman filter using the downhill simplex algorithm". In: *Journal of Guidance, Control, and Dynamics* 25.5 (2002), pp. 901–908 (Cited on pages 19, 51).

[52] S. J. Qin and T. A. Badgwell. "A survey of industrial model predictive control technology". In: *Control engineering practice* 11.7 (2003), pp. 733–764 (Cited on page 3).

[53] S. J. Qin and T. A. Badgwell. "An Overview of Nonlinear Model Predictive Control Applications". English. In: *Nonlinear Model Predictive Control*. Ed. by F. Allgöwer and A. Zheng. Vol. 26. Progress in Systems and Control Theory. Birkhäuser Basel, 2000, pp. 369–392. ISBN: 978-3-0348-9554-5 (Cited on page 3).

[54] M. R. Rajamani, J. B. Rawlings, and S. J. Qin. "Achieving state estimation equivalence for misassigned disturbances in offset-free model predictive control". In: *AIChE Journal* 55.2 (2009), pp. 396–407 (Cited on pages 5, 41).

[55] J. B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009. ISBN: 9780975937709 (Cited on pages 19, 38, 84).

[56] W. H. Ray. *Advanced process control*. McGraw-Hill chemical engineering series. McGraw-Hill, 1981. ISBN: 9780070512504 (Cited on pages 8, 27).

[57] I. Reid. "Estimation II". In: ().

[58] M. I. Ribeiro. "Kalman and extended kalman filters: Concept, derivation and properties". In: (2004).

[59] J. Richalet et al. "Model predictive heuristic control: Applications to industrial processes". In: *Automatica* 14.5 (1978), pp. 413–428 (Cited on page 40).

[60] M. Saha, B. Goswami, and R. Ghosh. "Two novel costs for determining the tuning parameters of the Kalman Filter". In: *arXiv preprint arXiv:1110.3895* (2011) (Cited on page 19).

[61] R. Scattolini. *Stability of model predictive control*. 2009. URL: http://users.dimi.uniud.it/~franco.blanchini/scuolasidra09/Ch6_Stability_MPC.pdf (visited on 04/15/2016) (Cited on page 37).

[62] D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley, 2006. ISBN: 9780470045336 (Cited on pages 14, 81, 86).

[63] S. Singer and J. Nelder. "Nelder-Mead algorithm". In: *Scholarpedia* 4.7 (2009), p. 2928.

[64] A. K. Singh and J. Hahn. "Determining optimal sensor locations for state and parameter estimation for stable nonlinear systems". In: *Industrial & engineering chemistry research* 44.15 (2005), pp. 5645–5659 (Cited on pages 29, 30).

[65] A. K. Singh and J. Hahn. "On the use of empirical gramians for controllability and observability analysis". In: *American Control Conference, 2005. Proceedings of the 2005*. IEEE. 2005, pp. 140–146.

[66] M. Srinivasarao, S. C. Patwardhan, and R. D. Gudi. "From data to nonlinear predictive control. 2. Improving regulatory performance using identified observers". In: *Industrial & engineering chemistry research* 45.10 (2006), pp. 3593–3603 (Cited on page 41).

[67] P. Tatjewski. “Disturbance modeling and state estimation for offset-free predictive control with state-space process models”. In: *International Journal of Applied Mathematics and Computer Science* 24.2 (2014), pp. 313–323 (Cited on pages 41, 68).

[68] M. J. Tenny. “Computational Strategies for Nonlinear Model Predictive Control”. PhD thesis. University of Wisconsin-Madison, June 2002 (Cited on page 7).

[69] T. O. Ting et al. “Tuning of Kalman Filter Parameters via Genetic Algorithm for State-of-Charge Estimation in Battery Management System”. In: *The Scientific World Journal* 2014 (2014) (Cited on pages 19, 51).

[70] U. Vaidya. “Observability gramian for nonlinear systems”. In: *Decision and Control, 2007 46th IEEE Conference on*. IEEE. 2007, pp. 3357–3362 (Cited on page 79).

[71] A. Wächter and L. T. Biegler. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. English. In: *Mathematical Programming* 106.1 (2006), pp. 25–57. ISSN: 0025-5610. DOI: 10.1007/s10107-004-0559-y. URL: http://dx.doi.org/10.1007/s10107-004-0559-y (Cited on page 59).

ACRONYMS

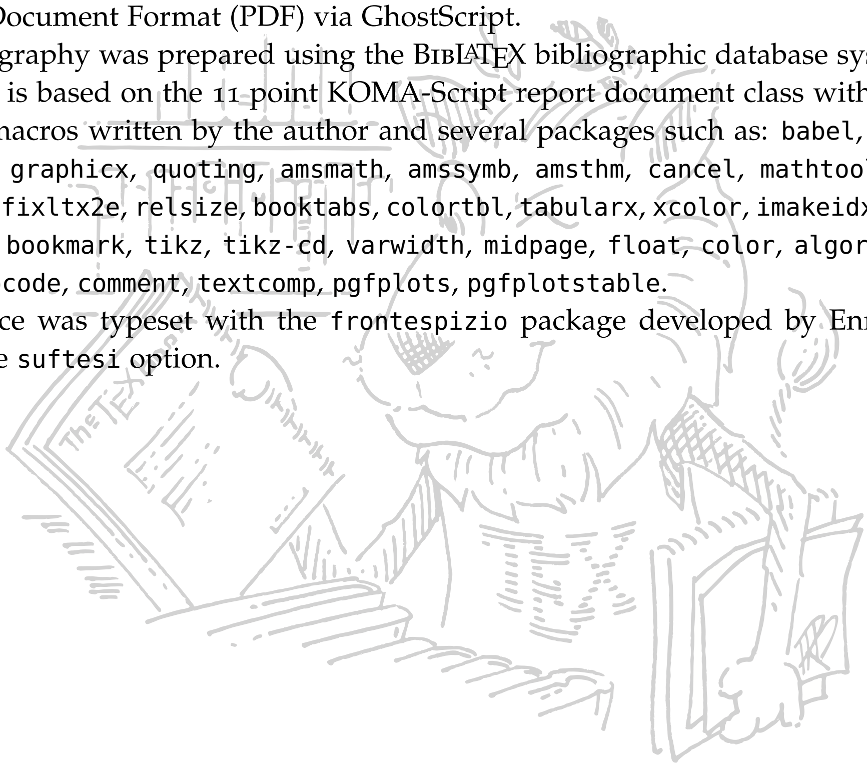
CN	Condition Number.
DAE	Differential Algebraic Equation.
DARE	Discrete-time Algebraic Riccati Equation.
DMC	Dynamic Matrix Control.
ECE	Equal Concern Error.
EKF	Extended Kalman Filter.
IAE	Integral of the Absolute magnitude of the Error.
IDM	Input Disturbance Model.
ISE	Integral of the Square of the Error.
ITAE	Integral of Time multiplied by the Absolute value of Error.
KF	Kalman Filter.
LICQ	Linear Independence Constraint Qualification.
LMPC	Linear Model Predictive Control.
LP	Linear Program.
LQR	Linear Quadratic Regulation.
LTI	Linear Time Invariant.
MAP	Maximum A Posteriori.
MHE	Moving Horizon Estimation.
MIMO	Multiple Input Multiple Output.
MMSE	Minimum Mean Square Error.
MPC	Model Predictive Control.
NLP	Nonlinear Program.
NMPC	Nonlinear Model Predictive Control.
OCP	Optimal Control Problem.
ODE	Ordinary Differential Equation.
ODM	Output Disturbance Model.

PDE	Partial Differential Equation.
QP	Quadratic Program.
RP	Random Process.
RV	Random Variable.
SISO	Single Input Single Output.
UKF	Unsceted Kalman Filter.

INDEX

COLOPHON

This thesis was typeset with the LaTeX 2ε Documentation System together with ArsClassica, a reworked version by Lorenzo Pantieri of the ClassicThesis style developed by André Miede inspired by Robert Bringhurst's seminal book on typography *"The Elements of Typographic Style"*, both on a notebook running Microsoft Windows 7 based on the TeX-Live total distribution.
Text editing was done inTeXstudio 2.9.4.
The illustrations and graphs were saved as Encapsuled PostScript (EPS) and converted into Portable Document Format (PDF) via GhostScript.
The bibliography was prepared using the BibLaTeX bibliographic database system.
This work is based on the 11 point KOMA-Script report document class with the addition of many macros written by the author and several packages such as: `babel`, `indentfirst`, `enumitem`, `graphicx`, `quoting`, `amsmath`, `amssymb`, `amsthm`, `cancel`, `mathtools`, `varioref`, `mparhack`, `fixltx2e`, `relsize`, `booktabs`, `colortbl`, `tabularx`, `xcolor`, `imakeidx`, `glossaries`, `geometry`, `bookmark`, `tikz`, `tikz-cd`, `varwidth`, `midpage`, `float`, `color`, `algorithm`, `subfig`, `algpseudocode`, `comment`, `textcomp`, `pgfplots`, `pgfplotstable`.
Frontispiece was typeset with the `frontespizio` package developed by Enrico Gregorio adding the `suftesi` option.



DECLARATION OF AUTHORSHIP

I, Mirco Lupi, declare that this thesis entitled:

"Disturbance Models for Offset-free Nonlinear Predictive Control"

and the work presented in it is my own except where otherwise indicated.

I confirm that:

- I have read and understood the information and guidance on academic good practice and plagiarism.
- Where any part of this dissertation has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated.
- I have clearly indicated the presence of all material I have quoted from other sources, including any diagrams, charts, tables or graphs.
- I have acknowledged appropriately any assistance I have received in addition to that provided by my advisor.
- I have not copied from the work of any other candidate.
- I have not used the services of any agency providing specimen, model or ghostwritten work in the preparation of this thesis.

Pisa, 10/05/2016

Mirco Lupi