POLITENICO DI TORINO

Master's Degree in Mechatronic Engineering

Master's Degree Thesis

A polynomial optimization approach to gray-box system identification

Supervisors

Prof. Diego REGRUTO

Dr. Sophie FOSSON

Prof. Vito CERONE

Candidate

Simone PIRRERA

May 2021

Abstract

System identification consists in the study of techniques allowing to estimate parameters of models for dynamical systems starting from input and output data experimentally collected. This problem is strongly motivated in the context of control system design and by a number of other practical situations such as the determination of useful system parameters in system's testing or analysis.
Among all the system identification algorithms, it is possible to recognize two big classes:

- Black-box identification: allows to estimate parameters of a model whose structure is selected according to some general a-priori information which does not require any knowledge of the physics of the systems.
- Gray-box identification: allows to estimate parameters of a so-called gray-box model, which is a dynamical model built by means of the first principles of physics in which some parameters may be unknown.

At the current state of art most of the system identification techniques are developed under the assumption that the model to be estimated belongs to the LTI systems class. Moreover most of the system identification algorithms deals with the problem of identifying black-box models, mostly described as a discrete-time transfer function in the $\mathcal{Z}$-domain.
On the contrary when dealing with gray-box identification the most natural choice is to consider continuous-time models having the form of state-space models. This is because when modeling a system thanks to the first principles of physics the involved equations are differential ones.
This thesis arises from the analysis of a recent article: 'Combining linear algebra and numerical optimization for gray-box affine state-space model identification', O. Prot and G. Mercère, IEEE Trans. Autom. Control, 2020. In this paper the authors, propose a linear algebra approach for grey-box system identification. Differently from previous literature, this approach has the advantage of not relying on non-convex optimization, thus it does not get trapped at local minima. Nevertheless, it has the drawback of not envisaging the presence of noise on the data.
In this thesis, we propose a new mathematical formulation of the gray-box identification problem that takes into account the possible presence of noise and relaxes the technical conditions considered in the above mentioned paper. In particular, this formalization of the problem is carried out in a set-membership framework, i.e. under the assumption that all variables involved in the problem belongs to some bounded set. In fact, the only assumption on the noise is that it is bounded and solving the problem we look for upper and lower bounds on each parameter we want to estimate, obtaining therefore an estimate giving information about the quality of the result.
The problem of finding the bounds of interest is formulated as a set of polynomial optimization problems (POPs), which are solved for global optimal by means of the tool of convex SDP relaxation, implemented in software as sparsePOP, SeDuMi and Mosek. Such software has been used in order to perform a validation of the proposed solutions with numerical examples.
Finally further relaxations of conditions under which we developed this new solution are discussed, in particular gray-box models with polynomial dependency from the parameters and MIMO systems are handled.

Acknowledgements

First, I would like to to express my gratitude to my supervisor, Prof. Diego Regruto Tomalino, who gave me the chance to work on this subject. He believed in my potentiality and motivated me to do my best. Besides him, I would like to thank my co-supervisors: Dr. Sophie Fosson for her patience and her careful advice and Prof. Vito Cerone for his continuous support.
Last but not least, I would also like to thank my girlfriend Jennifer and my family for encouraging me whenever I needed.

Table of Contents

List of Tables

List of Figures

Acronyms

SI
: System Identification

LTI
: Linear Time Invariant

DT
: Discrete Time

CT
: Continuous Time

SISO
: Single Input Single Output

SM
: Set Membership

FPS
: Feasible Parameters Set

PUI
: Parameters Uncertainty Interval

EIV
: Error in Variable

OE
: Output Error

POP

Polynomial Optimization Problem

SDP

SemiDefinite Program

SNR

Signal to Noise Ratio

Chapter 1

Preliminary Concepts

The objective of this thesis is to present a set of tools which can be used in order to perform the identification of gray-box dynamical models for linear time invariant systems. This is why in this first chapter we give an overview of the system identification problem and recall the distinction between black-box and gray-box models. Moreover some notions about identifiability are recalled.
Then, since in the following chapters the problem is dealt with in a set-membership framework, we also give a description of the set-membership estimation problem with particular focus on the estimation of discrete-time dynamical system parameters for system identification. Lastly, due to the fact that in such context polynomial optimization problems are naturally arising, a brief overview about theory allowing to solve them is presented.

1.1 System identification: an overview

System identification (SI) is the subject that studies how to build mathematical models of dynamical systems out of experimentally collected data [1].
This problem is strongly motivated for a number of reasons, among the main ones we can find

- Feedback control system's design, which in (almost) all cases requires a mathematical model of the plant to be controlled [2], [3].
- Modeling of economic processes useful for trying to predict behaviour of stocks for effective investments [4].
- Need to make simulations of systems: useful for example when we want to determine the response of a system when stimulated by an input which cannot be generated during an experiment [1].

In SI, two important classes of system are considered: black-box and gray-box models.

1.1.1 Black-box models

We define black-box those models whose structure is selected according to some general a-priori information which does not require any knowledge of the physics of the systems. [1].
The most common situation is when the system can be modeled as a linear time-invariant (LTI) system or can be linearized around the observation point. In this case the model structure is that of the discrete-time transfer function of some fixed order n

$$H(z) = \frac{\beta_n z^n + \cdots + \beta_1 z^1 + \beta_0}{z^n + \cdots + \alpha_1 z^1 + \alpha_0} \tag{1.1}$$

where α and β are parameters to be determined thanks to available data.
This problem has been widely studied in the literature, because on one hand this is the problem naturally arising in the context of control systems design [2], but also because its solution turns out to be easy under some assumptions [1]. Many solutions to this problem are available, and in Section 2.5 we present one of them, which is useful in the solution of the gray-box problem.
A less common choice for the a-priori choice of the model is that of the continuous-time transfer function

$$H(s) = \frac{\beta_n s^n + \cdots + \beta_1 s + \beta_0}{s^n + \cdots + \alpha_1 s + \alpha_0} \tag{1.2}$$

Also in this case the problem is motivated by control system's design problem [3], but its solution turns out to be harder with respect to the discrete-time case due to the presence of the time derivative operator s instead of the difference one which is simpler to be managed. A detailed overview of the state of art in this condition is presented in Chapter 3, where we also propose two algorithms for the solution of this problem.

1.1.2 Gray-box models

We define gray-box models those models built up by means of the first principles of physics, but where the physical parameters involved are unknown and needs to be estimated thanks to experimental data.
Dynamic systems can be modeled by means of differential equations [5]. In this case the most common setting is that when the system can be also assumed LTI and therefore gray-box model is described in terms of a state-space model as

$$\mathcal{M}_G = \begin{cases} \dot{x}(t) = A(\theta)x(t) + B(\theta)u(t) \\ y(t) = C(\theta)x(t) \end{cases} \tag{1.3}$$

where θ is some vector of physical parameters to be estimated.
This problem is still considered hard to be solved, even under some simplifying conditions, and is generally tackled through the minimization of some nonlinear and non-convex optimization problem which can give very bad results due to the fact that standard algorithms could trap in local minimums. In Chapter 3 we present a possible solution to this problem which is not affected by this issue.
Similarly, the gray-box identification problem can be posed in discrete-time when the gray-box model is

$$\mathcal{M}_G = \begin{cases} x(k+1) = A(\theta)x(k) + B(\theta)u(k) \\ y(k) = C(\theta)x(k) \end{cases} \tag{1.4}$$

Although this is less motivated, there are practical situations when problems of this kind can arise. The entire Chapter 2 is devoted to the solution of this problem.

1.2 Identifiability

The system identification problem, in the most common case of parametric models, can be mathematically posed as follows:
Problem: given a parametric model $\mathcal{M}(\theta)$ and a set of input-output data of the system $[u(\cdot), y(\cdot)]$, we want to find a parameter vector $\tilde{\theta}$ such that

$$y(\cdot) = \mathcal{M}(\tilde{\theta})u(\cdot) \tag{1.5}$$

Now, the questions we want to answer are

- Does the solution exist? Assuming that the model structure is correctly describing the behaviour of the system, yes.
- Does the solution is unique? This is the subject of identifiability, topic of the following discussion as in [1].

Two models $\mathcal{M}(\theta_1)$ and $\mathcal{M}(\theta_2)$ are said to be indistinguishable if

$$y_{\theta_1}(\cdot) = y_{\theta_2}(\cdot) \quad \forall u(\cdot) \tag{1.6}$$

and the model class $\mathcal{M}(\theta)$ is said to be globally identifiable if

$$\nexists \theta_1 \neq \theta_2 : \mathcal{M}(\theta_1) \text{ and } \mathcal{M}(\theta_2) \text{ are indistinguishable} \tag{1.7}$$

If $\mathcal{M}(\theta)$ is globally identifiable, then it is possible to determine the parameter vector $\tilde{\theta}$ uniquely.
Consider the SISO discrete-time transfer function

$$H(z|\theta) = \frac{\theta_{n+1} + \theta_{n+2}z^{-1} + \cdots + \theta_{2n+1}z^{-n}}{1 + \theta_1 z^{-1} + \cdots + \theta_n z^{-n}} \tag{1.8}$$

In this case for the identity principle for polynomials it holds

$$H(z|\theta_1) = H(z|\theta_2) \Longleftrightarrow \theta_1 = \theta_2 \tag{1.9}$$

and therefore the model is globally identifiable.
Next, consider a generic model [6], as for example the gray-box model in equation (1.4). In this case it is of course possible to find a corresponding transfer function model by

$$\begin{aligned} H(z|\theta) &= C(\theta)[zI_n - A(\theta)]^{-1}B(\theta) \\ &= \frac{\lambda_{n+1} + \lambda_{n+2}z^{-1} + \cdots + \lambda_{2n+1}z^{-n}}{1 + \lambda_1 z^{-1} + \cdots + \lambda_n z^{-n}} = H(z|\lambda) \end{aligned} \tag{1.10}$$

Of course, the coefficients of the transfer function λ must be related to the model's parameter θ by some function $\mathbb{R}^{\dim\theta} \to \mathbb{R}^{2n+1}$

$$\lambda = \lambda(\theta) \tag{1.11}$$

At this point it is possible to state that if the solution of this equation is unique, then $\mathcal{M}(\theta)$ is globally identifiable [6]. As a consequence, if $\dim\theta > 2n+1$ it is not possible to identify the system, since the solution cannot be unique. On the other hand, the condition $\dim\theta \leq 2n+1$ is not sufficient to state that the model is globally identifiable because in general equations (1.11) are non-linear and can have multiple solutions.
When the solution of (1.11) is not unique, then it is possible to consider the notion of local identifiability [1], [7]. A model class $\mathcal{M}(\theta)$ is said locally identifiable in $\hat{\theta}$ if there exists an ϵ-neighbourhood $\mathcal{B}_\epsilon(\hat{\theta})$ of $\hat{\theta}$ such that

$$\nexists\theta_1, \theta_2 \in \mathcal{B}_\epsilon(\hat{\theta}) : \mathcal{M}(\theta_1),\ \mathcal{M}(\theta_2) \text{ are indistinguishable} \tag{1.12}$$

Along this thesis, we will always assume that every gray-box model $\mathcal{M}_G(\theta)$ is identifiable at least locally.

1.2.1 Relation with observability and controllability

Given an LTI model $\mathcal{M}(\tilde{\theta})$ of order n, it is said to be

- Observable: when all the components of the state vector $x(t)$ can be estimated from the measure of the output $y(t)$. This can be checked by testing whether the matrix

$$\mathcal{O}_n = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{1.13}$$

  if full rank.

- Controllable: when the input signal $u(t)$ is able to move the state of the system $x(t)$ from any initial condition to any final condition in a finite time. In practice describes the ability of the input to modify the state. This can be checked by testing whether the matrix

$$\mathcal{C}_n = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \tag{1.14}$$

  if full rank.

It can be shown that, if some components of θ does not appear in the reachable and observable of $\mathcal{M}(\theta)$, then $\mathcal{M}(\theta)$ is not identifiable [8]. This can be interpreted, at least form an intuitive point of view, as the fact that in order to perform a sufficiently informative experiment the input must be able to excite all the dynamics of the system and the output must be able to capture such dynamics.
For this reason from now on, along this thesis it is assumed that the gray-box models $\mathcal{M}(\theta)$ we want to identify are minimal (i.e. both controllable and observable).

1.3 Set-membership identification

According to the set-membership (SM) estimation theory, it is assumed that the available measurements $\tilde{y}$ belong to a bounded set $\mathcal{Y}$. Also, is given some a-priori information on the system $\mathcal{S}$ to be estimated as

$$\mathcal{S} : f(\theta, \tilde{y}) = 0 \tag{1.15}$$

Thanks to available data and such a-priori information, it is then possible to define the feasible parameters set (FPS) $\mathcal{D}_\theta$ as the set of parameters θ which are consistent with both the a-priori information and the available data. In formulae:

$$\mathcal{D}_\theta = \left\{ \theta \in \mathbb{R}^D \ : \ f(\theta, \tilde{y}) = 0 \quad \forall \tilde{y} \in \mathcal{Y} \right\} \tag{1.16}$$

The SM estimation problem consist into correctly characterizing the set $\mathcal{D}_\theta$.
When SM estimation is applied to system identification [9, 10], such elements are given as:

- measurements collected from the system are a set of input and output samples experimentally collected

$$[\tilde{u}, \tilde{y}] \tag{1.17}$$

  and corrupted by some bounded noise. Generically it is possible to write

$$\begin{aligned} \tilde{u} &= \tilde{u}(u, \epsilon) \qquad \|\epsilon\|_1 \leq \Delta_\epsilon \in \mathbb{R} \\ \tilde{y} &= \tilde{y}(y, \eta) \qquad \|\eta\|_1 \leq \Delta_\eta \in \mathbb{R} \end{aligned} \tag{1.18}$$

  where u, y are the true values of the signals, while η, ϵ are the noise sequences.

- a-priori information on the system is given in terms of a regressor model

$$y(k) = f(\theta, r(k), \zeta(k)) \tag{1.19}$$

where $r(k)$ is the regressor, a vector depending on the past values of input and output and possibly by the present value of the input. In the same way $\zeta(k)$ is a vector containing present and past values of the noise samples.

This generic description allows to model many different noise structures and system's models. In this context, among the most common noise structures we can find

- Equation error (EE): when the regressor model has form

$$y(k) = f(\theta, r(k)) + e(k) \tag{1.20}$$

where $e(k)$ is an unique bounded noise sample $|e(k)| \leq \Delta_e$. In this case, under the additional assumption that $f(\cdot)$ is linear the problem can be solved by a set of linear programs (LP) since the FPS is defined by the set of linear inequalities

$$\begin{cases} y(k) - f(\theta, r(k)) \leq \Delta_e \in \mathbb{R} & k = 1, \ldots, N \\ f(\theta, r(k)) - y(k) \leq \Delta_e \in \mathbb{R} & k = 1, \ldots, N \end{cases}$$

and therefore is a polyhedron. Although this mathematical characteristic turns out to be nice form the computational point of view, the EE noise structure assumption turns out to be poorly realistic. For this reason we will not enter into more details about it.

- Errors-in-variables (EIV): in this case the noise is assumed to be additive on the signals as

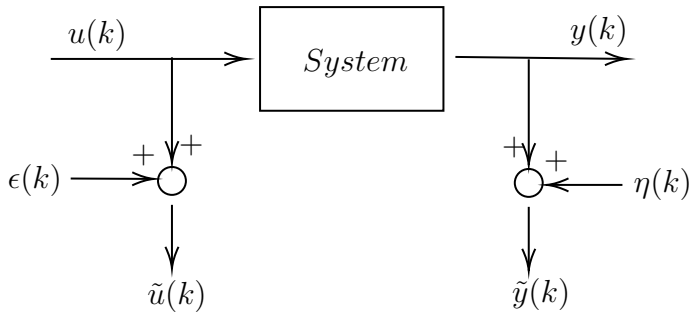

Figure 1.1: Block diagram of DT EIV noise structure

$$\begin{aligned} \tilde{u}(k) &= u(k) + \epsilon(k) \\ \tilde{y}(k) &= y(k) + \eta(k) \end{aligned} \tag{1.21}$$

Moreover each sample of the noise $\eta(k)$ and $\epsilon(k)$ is bounded as

$$\begin{aligned} \underline{\Delta_\eta}(k) \leq \eta(k) \leq \overline{\Delta_\eta}(k) \\ \underline{\Delta_\epsilon}(k) \leq \epsilon(k) \leq \overline{\Delta_\epsilon}(k) \end{aligned} \tag{1.22}$$

See [11].

- Output error (OE): is a particular case of the EIV structure, when

$$\underline{\Delta_\epsilon}(k) = \overline{\Delta_\epsilon}(k) = 0 \qquad \forall k = 1, \ldots, N$$

i.e. the input measurements are assumed to be noise free.

In the following, we will always assume an EIV noise structure as it is much more realistic in practical situations than the EE one. Another assumption we also do is that $f(\cdot)$ is linear in u, y (i.e. the system is LTI) and described by (1.1), so the regressor form of the model is

$$y(k) + \sum_{i=0}^{n-1} \alpha_i y(k-n+i) = \sum_{i=0}^{n} \beta_i u(k-n+i) \tag{1.23}$$

Therefore the FPS is described by

$$\begin{cases} \tilde{y}(k) - \eta(k) + \sum_{i=0}^{n-1} \alpha_i \tilde{y}(k-n+i) + \alpha_i \eta(k-n+i) = \\ \quad = \sum_{i=0}^{n} \beta_i \tilde{u}(k-n+i) + \beta_i \epsilon(k-n+i) \qquad k = n+1, \ldots, N \\ \underline{\Delta_\eta}(k) \leq \eta(k) \leq \overline{\Delta_\eta}(k) \qquad k = 1, \ldots, N \\ \underline{\Delta_\epsilon}(k) \leq \epsilon(k) \leq \overline{\Delta_\epsilon}(k) \qquad k = 1, \ldots, N \end{cases} \tag{1.24}$$

where $\theta = [\alpha, \beta]$ is parameter vector. When we want to find upper and lower bounds on the parameters θ_i in order to characterize the FPS, we need to solve optimization problems

$$\begin{aligned} \underline{\theta_i} &= \min_{\theta \in \mathcal{D}_\theta} \theta_i \qquad i = 1, \ldots, 2n+1 \\ \overline{\theta_i} &= \max_{\theta \in \mathcal{D}_\theta} \theta_i \qquad i = 1, \ldots, 2n+1 \end{aligned} \tag{1.25}$$

Since the description of $\mathcal{D}_\theta$ (1.24) involves polynomial equations of degree two (i.e. products between unknown variables η, ϵ, θ are present), optimization problems (1.25) is subject to polynomial equality constraints. Therefore (1.25) belongs to the class of polynomial optimization problems (POPs).

1.4 Polynomial optimization problems

As just discussed in the previous section, the problem of estimating parameters of a black-box DT transfer function model in the SM framework is formulated a set of POPs. Moreover, as we will accurately motivate along the following chapters, also the problems of estimating parameters of gray-box and continuous time models in the set-membership framework can be formulated with a set of POPs as well.
For those reasons, in this section we present some notions about POPs and their solution which allows to motivate considerations made in the following.
In general, a polynomial optimization problem [12] is the optimization problem

$$\begin{aligned} p^* = \min_x \, & p(x) \\ \text{subject to} \quad & \\ & g_j(x) \geq 0 \quad j = 1, \ldots, m \end{aligned} \tag{1.26}$$

where $p(x), g_1(x), \ldots, g_m(x) \in \mathbb{R}[x]$ are multivariate polynomials in x.
The feasible set of such optimization problem is

$$K = \{x : g_j(x) \geq 0 \quad j = 1, \ldots, m\} \tag{1.27}$$

and being described by a set of nonlinear and non-convex polynomial inequalities, this is in general also a non-convex semialgebraic set.
Non-convex problems are NP-hard and could in general present many local minima [13]. For this reason standard solvers based on gradient descent algorithms are likely to trap in some local minima, gives bad results.
In order to deal efficiently with the complexity of such problem, some authors proposed some approximation of the problem in order to reconduce it to a convex-one which can be solved efficiently for global optimality. In particular the hard polynomial problem is approximated with a semidefinite program (SDP), and this can be done mainly by means of two approaches.

1. SOS relaxations: it is possible to re-write the problem (1.26) as

$$\begin{aligned} p^* = \sup_{x,\rho} \, & \rho \\ \text{subject to} \quad & p(x) - \rho \geq 0 \text{ on } K \end{aligned}$$

   Moreover, since positive polynomials are a subset of sum of squares (SOS) polynomials, the relaxation is introduced by approximating

$$p(x) - \rho \approx s_0 + \sum_{j=1}^{m} s_j g_j \qquad \text{where } s_0, \ldots, s_m \text{ are SOS} \tag{1.28}$$

At this point, imposing that a polynomial $g_j(x)$ is a SOS can be simply done through the constraint

$$g_j(x) \text{ is SOS} \iff \exists X \succeq 0 : \tilde{x}^\top X \tilde{x}$$

where $\tilde{x} = \tilde{x}(x)$ is a monomial basis for the polynomial. Notice that such constraint is a linear matrix inequality as long as the degree of the polynomials are fixed, and this brings to an SDP problem.
So, chosen some integer t such that $2t \geq \max\{\deg(p), \deg(g_1), \ldots, \deg(g_m)\}$, the SOS-relaxed SDP problem is

$$\begin{aligned} p_t^{SOS} = & \sup_{x,\rho} \rho \\ & \text{subject to} \\ & p(x) - \rho = s_0 + \sum_{j=1}^{m} s_j g_j \qquad \text{where } s_0, \ldots, s_m \text{ are SOS} \\ & \deg(s_0), \deg(s_j g_j) \leq 2t \end{aligned} \tag{1.29}$$

2. Moment relaxation: proposed by Lasserre in [12], it is strongly based on theory of moments. Without entering into many formal details, it is possible to state that the problem (1.26) is rewritten as

$$\begin{aligned} p^* &= \inf_{x \in K} p(x) = \inf_{\mu} \int_K p(x)\mu(dx) = \\ &= \inf_{\mu} \sum_{\alpha} p_\alpha \int_K x^\alpha \mu(dx) = \\ &= \inf_{y} p^\top y \text{ s.t. } y_0 = 1, \, y \text{ has a representing measure on } K \end{aligned}$$

where μ is a Borel measure on $\mathbb{R}^n$ and $y = (y_\alpha)_{\alpha \in \mathbb{N}^n}$ is the sequence of moments of the measure μ, defined as $y_\alpha = \int x^\alpha \mu(dx)$.
Now, the condition 'y has a representing measure on K' can be replaced by the weaker conditions

$$M(y) \succeq 0 \qquad M(g_j y) \succeq 0 \, j = 1, \ldots, m$$

where $M(y)$ is the moment matrix of the sequence y (an infinite matrix whose (α, β)-th entry is $y_{\alpha+\beta}$) and $M(gy)$ is the localising moment matrix of y on $K_j = \{x : g_j(x) \geq 0\}$ defined as the moment matrix of the sequence $M(y)g_j$. This problem, cannot be already cast as an SDP due to the infinite dimensional matrices involved. For this reason, we need to consider instead truncated moment matrices $M_t(y)$, obtained in the same way but starting from a sequence

y truncated at order t. In this way, the relaxed problem can be formulated as the SDP

$$\begin{aligned} p^* \leq p_t^{MOM} = \inf_{y \in \mathbb{R}^{\mathbb{N}^n_{2t}}} & p^\top y \\ & \text{subject to} \\ & y_0 = 1, \quad M_t(y) \succeq 0 \\ & M_{t-\deg(g_j)}(g_j y) \succeq 0 \quad j = 1, \ldots, m \end{aligned} \tag{1.30}$$

Let us now see some properties about bounds p_t^{SOS} and p_t^{MOM} which can be obtained with (1.29) and (1.30).

- Duality: it is possible to verify that the problems are dual SDPs. For this reason

$$p_t^{SOS} \leq p_t^{MOM} \leq p^* \qquad \forall t \tag{1.31}$$

  Moreover when K has a nonempty relative interior, strong duality holds and $p_t^{SOS} = p_t^{MOM}$.

- Convergence: when the quadratic module $M(g_1, \ldots, g_m)$ is Archimedean (see Putinar's Positivstellensatz, [12, 14]) then

$$\lim_{t \to +\infty} p_t^{SOS} = p^* \tag{1.32}$$

  and considered (1.31), it is also true that

$$\lim_{t \to +\infty} p_t^{SOS} = \lim_{t \to +\infty} p_t^{MOM} = p^* \tag{1.33}$$

  Moreover, when the POP involves polynomial equality constraints $\{h_1(x) = 0, \ldots, h_{m_0}(x) = 0\}$ such that given the real variety associated to the ideal generated by given constraints is zero-dimensional, i.e. the set

$$V(\mathcal{J}) = \{x \in \mathbb{R} : f(x) = 0 \, \forall f \in \mathcal{J}\} \text{ where } \mathcal{J} = \left\{ \sum_{i=1}^{m_0} h_i u_i : u_i \in \mathbb{R}[x] \right\}$$

  is finite, then there exists some finite t for which $p^* = p_t^{MOM}$. When above condition is true in the complex field (harder condition), then $p^* = p_t^{SOS} = p_t^{MOM}$. See [15, 16].
  Considered the structure of the problems we are going to formulate, this is not an unrealistic condition, in fact the POPs formulated in this thesis will always involve a huge number of polynomial equality constraints.

Since the objective of the POPs we are formulating in the context of set-membership identification is to give a characterization of the feasible set in terms of upper and lower bounds of points belonging to the FPS, it is important to figure out what is the meaning of obtaining lower bounds for the global optima by means of DSP relaxations.

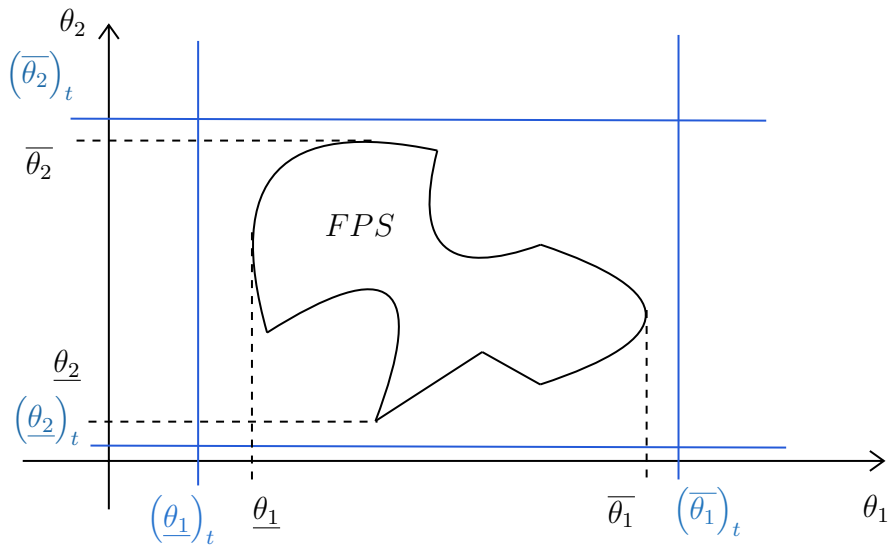

Figure 1.2: FPS relaxation

Consider figure 1.2. Since the parameters $(\underline{\theta_1})_t, (\underline{\theta_2})_t$ are evaluated by means of DSP relaxation of order t of POP (1.25), then given (1.31) it holds

$$(\underline{\theta_i})_t \leq \underline{\theta_i}$$

Equivalently when solving the maximization problems, since the maximization corresponds to the minimization of the negative objective, it also holds

$$-(\overline{\theta_i})_t \leq -\overline{\theta_i} \quad \Longrightarrow \quad (\overline{\theta_i})_t \geq \overline{\theta_i}$$

In other words, when solving the problem with SDP relaxation we are (at worst) giving a conservative description of the solution. Instead, if classical non-convex optimization methods were used, the solution could also trap in some local minima inside the FPS, therefore giving a wrong bound estimate.
Mentioned SDP relaxation methods are implemented in software as sparsePOP [17, 18]. This software requires a description of the polynomials $p(x), g_j(x), h(x)$ and is able to generate the associated SDP relaxed problem. Then it automatically calls some SDP solver such as SeDuMi [19] or Mosek [20] in order to compute the solution. In this work, all numerical results given in the examples are obtained thanks to these software.

Chapter 2

Discrete Time Problem

In this chapter we focus on the problem of estimating gray-box state space model parameters for SISO discrete-time LTI systems. Even if, as discussed in Section 1.1.2, the gray-box problem is usually much more motivated in continuous-time, it is worth investigating this problem for two reasons:

1. problems of this kind can naturally arise in problems involving networked systems as sensor networks;
2. the procedure established in this chapter will turn out to be useful when dealing with the harder CT problem in the following.

Firstly, results from [21] are reported and discussed, then an alternative formulation of the problem based on polynomial optimization is proposed. Starting from this formulation, modifications allowing to handle the presence of noise in the set-membership framework are introduced. Finally results from numerical experiments are reported.

2.1 Discrete time gray-box problem statement

The problem can be formulated as follows:
Problem: given the discrete time dynamical system described by state-space model $\mathcal{M}_G$

$$\mathcal{M}_G : \begin{cases} x(k+1) = A(\theta)x(k) + B(\theta)u(k) \\ y(k) = C(\theta)x(k) \end{cases} \tag{2.1}$$

and a set of samples from input and output sequences

$$\{\tilde{u}, \tilde{y}\} = \{\tilde{u}(1), \ldots, \tilde{u}(N), \tilde{y}(1), \ldots, \tilde{y}(N)\} \tag{2.2}$$

we need to find the parameters vector θ.

2.2 A linear approach

The classical solution to the problem [1], is based on the minimization of the prediction error as

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\theta|\tilde{u},\tilde{y}) = \arg\min_{\theta} \sum_{k=1}^{N} \|\tilde{y}(k) - \hat{y}_{\tilde{u}}(k,\theta)\|_2^2 \tag{2.3}$$

where $\hat{y}_{\tilde{u}}$ denotes the output of the parametric model when input $\tilde{u}$ is applied. The main issue of this solution is that in many cases the above cost function to be minimized is highly non-convex and as a consequence standard solvers are not able in general to compute the global optima.
In [21], the authors propose a linear algebra solution for the problem allowing to avoid this issue. Their approach can be briefly resumed in the following steps:

1. By means of any black-box system identification algorithm, compute a state space black-box model $\mathcal{M}_B$ for the system starting from the given input-output data. Out of this step we get a set of matrices $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$ such that

$$\mathcal{M}_B : \begin{cases} x(k+1) = \mathcal{A}x(k) + \mathcal{B}u(k) \\ y(k) = \mathcal{C}x(k) \end{cases} \tag{2.4}$$

2. Matrices $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$ obtained from the previous step are related to the matrices $A(\theta), B(\theta)$ and $C(\theta)$ by a similarity transformation

$$\mathcal{A}T = TA(\theta) \qquad \mathcal{B} = TB(\theta) \qquad \mathcal{C}T = C(\theta) \tag{2.5}$$

   or equivalently

$$S\mathcal{A} = A(\theta)S \qquad S\mathcal{B} = B(\theta) \qquad \mathcal{C} = C(\theta)S \tag{2.6}$$

   Where $S = T^{-1}$.
   Out of this equations it is possible to find a set of linear equalities thanks to the known terms of $A(\theta), B(\theta)$ and $C(\theta)$.

3. Equalities obtained in the previous step can now be arranged in matrix form as

$$M_T \text{vec}(T) = \beta_T \qquad \text{or} \qquad M_S \text{vec}(S) = \beta_S \tag{2.7}$$

   depending on where (2.5) or (2.6) has been chosen at the previous step, where with $\text{vec}(X)$ is denoted the vector obtained by arranging the elements of matrix X in a column vector, while M_X and β_X are completely known matrices and vectors.

4. if either M_S or M_T are full rank, then it is possible to solve for unknowns T or S, and finally find θ by comparison of the matrices $T^{-1}\mathcal{A}T$ with $A(\theta)$, $T^{-1}\mathcal{B}$ with $B(\theta)$ and $\mathcal{C}T$ with $C(\theta)$. Otherwise, if neither M_S nor M_T are full rank, it is possible to combine this technique with nonlinear optimization in order to find the missing coefficients of the similarity transformation.

Such an approach relies on linear algebra tools only, and therefore turns out to be computationally really efficient in finding the solution. On the other hand, since the matrices M_S and M_T are not guaranteed to be full rank in general, the "pure" linear algebra approach fails and the problem gets harder due to the necessity to introduce nonlinear optimization.
Secondly, this procedure does not take into account any form of noise affecting the data and assumes that the black-box description obtained in the first step of the procedure perfectly describes the system. In fact, numerical experiments show that in the presence of noise the parameters θ are not correctly estimated and the error may become dramatically large when the noise augments. Moreover, out of such procedure there is no way to determine the quality of the result even if the noise can be correctly characterized.

2.3 Polynomial approach

In the procedure summarized above, a critical aspect is that equations (2.5) are in general bilinear even if the matrices $A(\theta), B(\theta)$ and $C(\theta)$ are assumed to be affine in θ. As a consequence, in order to get linear equations out of them, we need to discard all bilinear equations and keep just those among them which are linear.
An alternative idea, is to retain all information from such equations and solve instead the polynomial problem of finding the unknown θ by means of the solution of the system of $n_x^2 + 2n_x$ polynomial (bilinear) equations in $n_x^2 + D$ unknowns:

$$\begin{cases} \mathcal{A}T = TA(\theta) \\ \mathcal{B} = TB(\theta) \\ \mathcal{C}T = C(\theta) \end{cases} \tag{2.8}$$

The problem of solving such polynomial systems of equations has been considered in the literature. One method can be used in order to determine the solution is the Stetter-Möller method (or eigenvalue method). This method is based on the relation between the variety associated to the ideal $\mathcal{I}$ generated by the polynomials with eigenvalues of the quotient space $\mathbb{R}[x]/\mathcal{I}$. For a detailed description of the method, refer to [22].
It is possible to notice that the considered system of equations involves usually more equations than unknowns since $D \ll 2n_x$. As a consequence now we can

make some considerations about the expected number of solutions of the problem under different conditions.

- When the identified black-box matrices $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$ are realization of the same system as the gray-box ones $\{A(\theta), B(\theta), C(\theta)\}$ (e.g. the black-box state space model has been obtained through noiseless data), the mapping T between the two systems is unique and we expect to find just one solution.

- When the identified black-box matrices $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$ and the gray-box ones $\{A(\theta), B(\theta), C(\theta)\}$ are not realization of the same system, i.e. there is no matrix T allowing to map one into the others, we expect an empty set of solutions. This is the most reasonable situation since usually experimentally collected data are effected by noise and therefore it is not possible to find $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$ exactly representing the true system.

Overall, this approach turns out to be useless since even a small noise can cause an incorrect estimation of $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$ and as a consequence the final set of solutions is empty.

2.4 Set-membership problem

Formulating the problem in a set-membership framework, allows to overcome the problems due to the presence of the noise in the latter approach. It is possible to show, that
Result: if the noise affecting the data can be modelled as an EIV (refer to Section 1.3), then it is possible to obtain a black-box model describing the system as

$$\mathcal{M}_B : \begin{cases} x(k+1) = \mathcal{A}x(k) + \mathcal{B}u(k) \\ y(k) = \mathcal{C}x(k) \end{cases} \tag{2.9}$$

where matrices $\mathcal{A} \in \mathbb{R}^{n_x, n_x}, \mathcal{B} \in \mathbb{R}^{n_x}$ and $\mathcal{C}^\top \in \mathbb{R}^{n_x}$ are unknown but bounded. In formulas, this means that each element of $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$ can be described by:

$$\begin{aligned} &\underline{\mathcal{A}}_{ij} \leq \mathcal{A}_{ij} \leq \overline{\mathcal{A}}_{ij} \quad 1 \leq i, j \leq n_x \\ &\underline{\mathcal{B}}_i \leq \mathcal{B}_i \leq \overline{\mathcal{B}}_i \quad 1 \leq i \leq n_x \\ &\underline{\mathcal{C}}_i \leq \mathcal{C}_i \leq \overline{\mathcal{C}}_i \quad 1 \leq i \leq n_x \end{aligned} \tag{2.10}$$

How to obtain such a description of the system is the core of the next section, where a constructive procedure allowing to build such model is presented.
Thanks to such result, it is possible to formulate the problem of finding bounds on

the parameters to be estimated as the optimization problems

$$\begin{aligned}
\underline{\theta}_i = & \min_{\theta,\mathcal{A},\mathcal{B},\mathcal{C},T} \theta_i \\
& \text{subject to} \\
& \mathcal{A}T = TA(\theta) \\
& \mathcal{B} = TB(\theta) \\
& \mathcal{C}T = C(\theta) \\
& \underline{\mathcal{A}}_{ij} \leq \mathcal{A}_{ij} \leq \overline{\mathcal{A}}_{ij} \quad 1 \leq i,j \leq n_x \\
& \underline{\mathcal{B}}_i \leq \mathcal{B}_i \leq \overline{\mathcal{B}}_i \quad 1 \leq i \leq n_x \\
& \underline{\mathcal{C}}_i \leq \mathcal{C}_i \leq \overline{\mathcal{C}}_i \quad 1 \leq i \leq n_x
\end{aligned} \tag{2.11}$$

And similarly for the upper bound

$$\begin{aligned}
\overline{\theta}_i = & \max_{\theta,\mathcal{A},\mathcal{B},\mathcal{C},T} \theta_i = \min_{\theta,\mathcal{A},\mathcal{B},\mathcal{C},T} -\theta_i \\
& \text{subject to} \\
& \mathcal{A}T = TA(\theta) \\
& \mathcal{B} = TB(\theta) \\
& \mathcal{C}T = C(\theta) \\
& \underline{\mathcal{A}}_{ij} \leq \mathcal{A}_{ij} \leq \overline{\mathcal{A}}_{ij} \quad 1 \leq i,j \leq n_x \\
& \underline{\mathcal{B}}_i \leq \mathcal{B}_i \leq \overline{\mathcal{B}}_i \quad 1 \leq i \leq n_x \\
& \underline{\mathcal{C}}_i \leq \mathcal{C}_i \leq \overline{\mathcal{C}}_i \quad 1 \leq i \leq n_x
\end{aligned} \tag{2.12}$$

Let's now analyze the nature of the optimization problems (2.11) and (2.12):

- The optimization variable is composed by all the unknown parameters in the gray-box model θ, but also by all elements of matrices $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and T. So, letting
$$D = \dim \theta \quad \text{and} \quad n_x = \text{order of the system} \tag{2.13}$$
then the number of (scalar) variables involved in the optimization problems is $D + 2n_x^2 + 2n_x$. This is quadratically increasing with the order of the system and in practice is quite small in most of the cases.

- The objective function is a simple linear function involving just one variable of the vector θ

- The feasible set is defined by the following constraints:

  1. $\mathcal{A}T = TA(\theta)$ represents a vector of n_x^2 equality constraints involving bi-linear terms due to $\mathcal{A}T$ and, in general, nonlinear terms due to the

product $TA(\theta)$ when each element of $A(\theta)$ is any nonlinear function of the variable θ.
In order to simplify the following derivations, I assume that $A(\theta)$ is affine in θ and as a consequence the terms of the product $TA(\theta)$ are bi-linear in the optimization variables.

2. $\mathcal{B} = TB(\theta)$ are a vector of n_x equality constraints involving nonlinear terms in θ in general, but assuming that $B(\theta)$ is affine in θ as already did for matrix $A(\theta)$, such constraints turns out again to be bi-linear.
3. $\mathcal{C}T = C(\theta)$ is another vector of n_x potentially nonlinear equality constraints. As long as I assume also $C(\theta)$ to be affine in θ, then such constraints are also bilinear in the optimization variable.
4. $\underline{\mathcal{A}}_{ij} \leq \mathcal{A}_{ij} \leq \overline{\mathcal{A}}_{ij}$, $\underline{\mathcal{B}}_i \leq \mathcal{B}_i \leq \overline{\mathcal{B}}_i$, $\underline{\mathcal{C}}_i \leq \mathcal{C}_i \leq \overline{\mathcal{C}}_i$ are linear inequality constraints.

Therefore, it is possible to state that, under the assumption of $A(\theta), B(\theta), C(\theta)$ being affine in θ the problems (2.11) and (2.12) are POPs due to the polynomial constraints $\mathcal{A}T = TA(\theta)$, $\mathcal{B} = TB(\theta)$ and $T\mathcal{C} = C(\theta)$ and have degree 2.

2.5 Derivation of the uncertain black-box model

In this section we discuss how to obtain an uncertain discrete-time state space model as described in equations (2.9) and (2.10) starting from a set $\{\tilde{u}, \tilde{y}\}$ of input-output data. Such data are assumed to be corrupted by a bounded noise as in (1.21) and (1.22).
The procedure allowing us to get such a black-box model, consists in two steps: at first we need to find bounds on transfer function parameters, then convert the transfer function model into an equivalent state space one.

2.5.1 From data to transfer function

The first step is to obtain bounds on the coefficients (α, β) of the transfer function

$$H(z|\alpha, \beta) = \frac{\beta_1 z^{-1} + \cdots + \beta_n z^{-n}}{1 + \alpha_1 z^{-1} + \cdots + \alpha_n z^{-n}} \tag{2.14}$$

describing the input-output mapping, given the available data and the system order n_x. Notice that it is assumed $\beta_0 = 0$ due to the fact that the considered structure of the gray-box model implicitly requires the considered system to be strictly proper. More then one solution have been proposed in [23], [24] and more recently by means of polynomial optimization techniques [11].

In the SM framework and considering EIV noise structure, the problem consists in the solution of POPs (1.25) which are explicitly written as

$$\begin{aligned}
\underline{\theta_i^B} = \quad & \min_{\alpha,\beta,\eta,\epsilon} \theta_i^B \\
& \text{subject to} \\
& \tilde{y}(k) = \eta(k) + \sum_{j=1}^{n_x} \beta_j \tilde{u}(k-j) - \sum_{j=1}^{n_x} \beta_j \epsilon(k-j) + \\
& \quad - \sum_{j=1}^{n_x} \alpha_j \tilde{y}(k-j) + \sum_{j=1}^{n_x} \alpha_j \eta(k-j) \quad k = n_x+1, \ldots, N \\
& \underline{\Delta_\eta}(k) \leq \eta(k) \leq \overline{\Delta_\eta}(k) \qquad k = 1, \ldots, N \\
& \underline{\Delta_\epsilon}(k) \leq \epsilon(k) \leq \overline{\Delta_\epsilon}(k) \qquad k = 1, \ldots, N
\end{aligned} \tag{2.15}$$

and

$$\begin{aligned}
\overline{\theta_i^B} = \max_{\alpha,\beta,\eta,\epsilon} \theta_i^B = \min_{\alpha,\beta,\eta,\epsilon} & -\theta_i^B \\
& \text{subject to} \\
& \tilde{y}(k) = \eta(k) + \sum_{j=1}^{n_x} \beta_j \tilde{u}(k-j) - \sum_{j=1}^{n_x} \beta_j \epsilon(k-j) + \\
& \quad - \sum_{j=1}^{n_x} \alpha_j \tilde{y}(k-j) + \sum_{j=1}^{n_x} \alpha_j \eta(k-j) \quad k = n_x+1, \ldots, N \\
& \underline{\Delta_\eta}(k) \leq \eta(k) \leq \overline{\Delta_\eta}(k) \qquad k = 1, \ldots, N \\
& \underline{\Delta_\epsilon}(k) \leq \epsilon(k) \leq \overline{\Delta_\epsilon}(k) \qquad k = 1, \ldots, N
\end{aligned} \tag{2.16}$$

Lower (or upper, when maximizing) bounds on the parameters can be found by means of convex relaxation techniques such as the SDP ones presented in 1.4.
The number number of optimization problems to be solved in this step is $2(2n_x) = 4n_x$, while the number of variables are $2n_x + 2N$.

2.5.2 From transfer function to state space

The conversion from transfer function to state space model can be performed in more than one way, and each conversion have different properties. In order to understand which conversion is most suitable for this particular problem, let us now recall some standard techniques allowing to operate such conversion.
Given a n_x-th order transfer function (2.14) there exists an infinite set of minimal state space models (2.9) such that their transfer function computed as

$$H(z) = \mathcal{C}(zI_{n_x} - \mathcal{A})^{-1}\mathcal{B} \tag{2.17}$$

corresponds to the given transfer function. Among them, some canonical forms have been considered in the literature:

- controllable canonical form: involves the coefficients of $H(z)$ in a row of $\mathcal{A}$ and in $\mathcal{C}$ as

$$\begin{aligned}\mathcal{A}^{ctb} &= \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ -\alpha_{n_x} & -\alpha_{n_x-1} & \dots & -\alpha_2 & -\alpha_1 \end{bmatrix} \\ \mathcal{B}^{ctb} &= \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \end{bmatrix}^\top \\ \mathcal{C}^{ctb} &= \begin{bmatrix} \beta_{n_x} & \beta_{n_x-1} & \dots & \beta_2 & \beta_1 \end{bmatrix}\end{aligned} \tag{2.18}$$

- observable canonical form: involves the coefficients of $H(z)$ in a column of $\mathcal{A}$ and in $\mathcal{B}$ as

$$\begin{aligned}\mathcal{A}^{obs} &= \begin{bmatrix} 0 & 0 & \dots & 0 & -\alpha_{n_x} \\ 1 & 0 & \dots & 0 & -\alpha_{n_x-1} \\ 0 & 1 & \dots & 0 & -\alpha_{n_x-2} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & -\alpha_1 \end{bmatrix} \\ \mathcal{B}^{obs} &= \begin{bmatrix} \beta_{n_x} & \beta_{n_x-1} & \dots & \beta_2 & \beta_1 \end{bmatrix}^\top \\ \mathcal{C}^{obs} &= \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \end{bmatrix}\end{aligned} \tag{2.19}$$

- Jordan canonical form: does not involve directly all the coefficients but the poles $p \in \mathbb{C}^{n_x}$ and the residuals $r \in \mathbb{C}^{n_x}$ of the transfer function. Suppose as an example that $H(z)$ allows a partial fraction expansion in the form

$$H(z) = \frac{r_{1,1}}{(z+p_1)^2} + \frac{r_{1,2}}{z+p1} + \frac{r_2}{z+p_2} + \dots + \frac{r_{n_x}}{z+p_{n_x}} \tag{2.20}$$

  Then the corresponding state-space model matrices in Jordan canonical form are

$$\begin{aligned}\mathcal{A}^{Jordan} &= \begin{bmatrix} -p_1 & 1 & 0 & \dots & 0 \\ 0 & -p_1 & 0 & \dots & 0 \\ 0 & 0 & -p_2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & -p_{n_x} \end{bmatrix} \\ \mathcal{B}^{Jordan} &= \begin{bmatrix} 0 & 1 & \dots & 1 & 1 \end{bmatrix}^\top \\ \mathcal{C}^{Jordan} &= \begin{bmatrix} r_{1,1} & r_{1,2} & r_2 & \dots & r_{n_x} \end{bmatrix}\end{aligned} \tag{2.21}$$

Among all those possibilities, the Jordan form should be discarded since it is not straightforward to link bounds on the transfer function parameters $\theta^B = (\alpha, \beta)$ with bounds on poles and residuals. Moreover the problem would be badly conditioned due to the high sensitivity of the poles with respect to the denominator's coefficients. Instead, at a first look controllability and observability canonical forms may instead seem to be equally suitable for the selected task since both of them are easily built just by a suitable re-arrangement of the transfer function parameters without additional manipulations. Performing a deeper analysis about that, we can notice that

- In both cases, matrix $\mathcal{A}$ is sparse (i.e. contains many zeros) and also some terms are known to be exactly equal to one, without uncertainty. Sparsity allows to reduce the number of terms arising when forming the product $\mathcal{A}T$, while the presence of not-uncertain terms allows to simplify the expressions even if not reducing the number of terms.
- In controllability canonical form, matrix $\mathcal{B}$ is composed by just ones and zeros, while in observability canonical form matrix $\mathcal{C}$ is composed by all zeros and ones. Assuming now $A(\theta), B(\theta)$ and $C(\theta)$ affine in θ, it is possible to notice that while using controllability canonical form all equations (2.5) remains bilinear in the variables T, θ, α, β, using controllability canonical form equation $\mathcal{C}T = C(\theta)$ gets linear in the unknown variables. This is an important advantage in polynomial optimization since linear constraints are not relaxed and the solution is likely to be less conservative.

For those reasons, the following calculations are developed using the controllability canonical form. In fact, substituting $\mathcal{A}^{obs}, \mathcal{B}^{obs}$ and $\mathcal{C}^{obs}$ in place of the generic $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$ in (2.5), we find:

$$\mathcal{A}^{obs}(\alpha)T = TA(\theta) \quad \Leftrightarrow$$

$$\begin{bmatrix} 0 & 0 & \dots & 0 & -\alpha_{n_x} \\ 1 & 0 & \dots & 0 & -\alpha_{n_x-1} \\ 0 & 1 & \dots & 0 & -\alpha_{n_x-2} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & -\alpha_1 \end{bmatrix} \begin{bmatrix} t_{1,1} & t_{1,2} & \dots & t_{1,n_x-1} & t_{1,n_x} \\ t_{2,1} & t_{2,2} & \dots & t_{2,n_x-1} & t_{2,n_x} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ t_{n_x-1,1} & t_{n_x-1,2} & \dots & t_{n_x-1,n_x-1} & t_{n_x-1,n_x} \\ t_{n_x,1} & t_{n_x,2} & \dots & t_{n_x,n_x-1} & t_{n_x,n_x} \end{bmatrix} =$$

$$= \begin{bmatrix} t_{1,1} & t_{1,2} & \dots & t_{1,n_x-1} & t_{1,n_x} \\ t_{2,1} & t_{2,2} & \dots & t_{2,n_x-1} & t_{2,n_x} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ t_{n_x-1,1} & t_{n_x-1,2} & \dots & t_{n_x-1,n_x-1} & t_{n_x-1,n_x} \\ t_{n_x,1} & t_{n_x,2} & \dots & t_{n_x,n_x-1} & t_{n_x,n_x} \end{bmatrix} A(\theta)$$

Then, explicitly taking into account the assumption about affinity of $A(\theta)$ with respect to θ as

$$A(\theta) = A^{(0)} + \sum_{d=1}^{D} A^{(d)}\theta_d \tag{2.22}$$

the previous equation is re-written as

$$\begin{bmatrix} 0 & 0 & \dots & 0 & -\alpha_{n_x} \\ 1 & 0 & \dots & 0 & -\alpha_{n_x-1} \\ 0 & 1 & \dots & 0 & -\alpha_{n_x-2} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & -\alpha_1 \end{bmatrix} \begin{bmatrix} t_{1,1} & t_{1,2} & \dots & t_{1,n_x-1} & t_{1,n_x} \\ t_{2,1} & t_{2,2} & \dots & t_{2,n_x-1} & t_{2,n_x} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ t_{n_x-1,1} & t_{n_x-1,2} & \dots & t_{n_x-1,n_x-1} & t_{n_x-1,n_x} \\ t_{n_x,1} & t_{n_x,2} & \dots & t_{n_x,n_x-1} & t_{n_x,n_x} \end{bmatrix} =$$

$$= \begin{bmatrix} t_{1,1} & t_{1,2} & \dots & t_{1,n_x-1} & t_{1,n_x} \\ t_{2,1} & t_{2,2} & \dots & t_{2,n_x-1} & t_{2,n_x} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ t_{n_x-1,1} & t_{n_x-1,2} & \dots & t_{n_x-1,n_x-1} & t_{n_x-1,n_x} \\ t_{n_x,1} & t_{n_x,2} & \dots & t_{n_x,n_x-1} & t_{n_x,n_x} \end{bmatrix} \left[A^{(0)} + \sum_{d=1}^{D} A^{(d)}\theta_d \right]$$

Then, expanding row-by column matrix multiplications

$$\begin{aligned} -\alpha_{n_x} t_{n_x,j} &= \sum_{k=1}^{n_x} t_{1,k} \left(a_{k,j}^{(0)} + \sum_{d=1}^{D} a_{k,j}^{(d)}\, \theta_d \right) \\ &= \sum_{k=1}^{n_x} t_{1,k}\, a_{k,j}^{(0)} + \sum_{k=1}^{n_x}\sum_{d=1}^{D} a_{k,j}^{(d)}\, t_{1,k}\, \theta_d \end{aligned} \tag{2.23}$$

for what concerns the first row (all columns), while

$$\begin{aligned} t_{i-1,j} - \alpha_{n_x-i+1} t_{n_x,j} &= \sum_{k=1}^{n_x} t_{i,k} \left(a_{k,j}^{(0)} + \sum_{d=1}^{D} a_{k,j}^{(d)}\, \theta_d \right) \\ &= \sum_{k=1}^{n_x} t_{i,k}\, a_{k,j}^{(0)} + \sum_{k=1}^{n_x}\sum_{d=1}^{D} a_{k,j}^{(d)}\, t_{i,k}\, \theta_d \end{aligned} \tag{2.24}$$

for what concerns all the other elements. In above equations $a_{k,j}^{(d)}$ denotes element k, j of matrix $A^{(d)}$ with $d = 0,1,\dots,D$.
Similarly, we also substitute $\mathcal{B}^{obs}$ in place of $\mathcal{B}$ in $\mathcal{B} = TB(\theta)$ obtaining

$$\mathcal{B}^{obs}(\beta) = TB(\theta) \qquad \Leftrightarrow$$

$$\begin{bmatrix} \beta_{n_x} \\ \beta_{n_x-1} \\ \vdots \\ \beta_2 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} t_{1,1} & t_{1,2} & \dots & t_{1,n_x-1} & t_{1,n_x} \\ t_{2,1} & t_{2,2} & \dots & t_{2,n_x-1} & t_{2,n_x} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ t_{n_x-1,1} & t_{n_x-1,2} & \dots & t_{n_x-1,n_x-1} & t_{n_x-1,n_x} \\ t_{n_x,1} & t_{n_x,2} & \dots & t_{n_x,n_x-1} & t_{n_x,n_x} \end{bmatrix} B(\theta)$$

Then, assuming also $B(\theta)$ affine in θ as

$$B(\theta) = B^{(0)} + \sum_{d=1}^{D} B^{(d)}\theta_d \tag{2.25}$$

the matrix product can be written as

$$\begin{bmatrix} \beta_{n_x} \\ \beta_{n_x-1} \\ \vdots \\ \beta_2 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} t_{1,1} & t_{1,2} & \dots & t_{1,n_x-1} & t_{1,n_x} \\ t_{2,1} & t_{2,2} & \dots & t_{2,n_x-1} & t_{2,n_x} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ t_{n_x-1,1} & t_{n_x-1,2} & \dots & t_{n_x-1,n_x-1} & t_{n_x-1,n_x} \\ t_{n_x,1} & t_{n_x,2} & \dots & t_{n_x,n_x-1} & t_{n_x,n_x} \end{bmatrix} \left[B^{(0)} + \sum_{d=1}^{D} B^{(d)}\theta_d \right]$$

and expanding row-by-column matrix product we end up with

$$\begin{aligned} \beta_{n_x-i+1} &= \sum_{k=1}^{n_x} t_{i,k} \left(b_k^{(0)} + \sum_{d=1}^{D} b_k^{(d)}\, \theta_d \right) \\ &= \sum_{k=1}^{n_x} b_k^{(0)}\, t_{i,k} + \sum_{k=1}^{n_x} \sum_{d=1}^{D} b_k^{(d)}\, t_{i,k}\, \theta_d \end{aligned} \tag{2.26}$$

with $i = 1, \dots, n_x$.
Lastly, we also need to substitute $\mathcal{C}^{obs}$ in place of $\mathcal{C}$ in $\mathcal{C}T = C(\theta)$, obtaining

$$\mathcal{C}^{obs}T = C(\theta) \qquad \Leftrightarrow$$

$$\begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} t_{1,1} & t_{1,2} & \dots & t_{1,n_x-1} & t_{1,n_x} \\ t_{2,1} & t_{2,2} & \dots & t_{2,n_x-1} & t_{2,n_x} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ t_{n_x-1,1} & t_{n_x-1,2} & \dots & t_{n_x-1,n_x-1} & t_{n_x-1,n_x} \\ t_{n_x,1} & t_{n_x,2} & \dots & t_{n_x,n_x-1} & t_{n_x,n_x} \end{bmatrix} = C(\theta)$$

Then, assuming also $C(\theta)$ affine in θ as

$$C(\theta) = C^{(0)} + \sum_{d=1}^{D} C^{(d)}\theta_d \tag{2.27}$$

the matrix product can be written as

$$\begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} t_{1,1} & t_{1,2} & \dots & t_{1,n_x-1} & t_{1,n_x} \\ t_{2,1} & t_{2,2} & \dots & t_{2,n_x-1} & t_{2,n_x} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ t_{n_x-1,1} & t_{n_x-1,2} & \dots & t_{n_x-1,n_x-1} & t_{n_x-1,n_x} \\ t_{n_x,1} & t_{n_x,2} & \dots & t_{n_x,n_x-1} & t_{n_x,n_x} \end{bmatrix} = \left[C^{(0)} + \sum_{d=1}^{D} C^{(d)}\, \theta_d \right]$$

Expanding row-by-column products we end up with

$$t_{n_x,i} = c_i^{(0)} + \sum_{d=1}^{D} c_i^{(d)} \theta_d \qquad i = 1, \ldots, n_x \tag{2.28}$$

which is linear/affine in unknown variables $[T, \theta, \alpha, \beta]$, as expected.
Thanks to this procedure we are able to find bounds on parameters (α, β) of a state space model $\{\mathcal{A}^{obs}(\alpha), \mathcal{B}^{obs}(\beta), \mathcal{C}^{obs}\}$ out of noisy samples of input and output sequences. Moreover it is remarkable that in this approach all the information of the black-box model is encoded in $2n$ parameters only, namely α and β, therefore in formulation of POPs (2.11) and (2.12) it is not even necessary to include as variables all the elements of $\mathcal{A}, \mathcal{B}$ and $\mathcal{C}$, but just α and β are sufficient. As a consequence, such POPs can be re-written as

$$\begin{aligned} \underline{\theta}_i \quad = \min_{\theta,\alpha,\beta,T} \quad & \theta_i \\ \text{subject to} \quad & \mathcal{A}^{obs}(\alpha) T = T A(\theta) \\ & \mathcal{B}^{obs}(\beta) = T B(\theta) \\ & \begin{bmatrix} 0 & \ldots & 0 & 1 \end{bmatrix} T = C(\theta) \\ & \underline{\alpha_i} \leq \alpha_i \leq \overline{\alpha_i} \quad i = 1, \ldots, n_x \\ & \underline{\beta_i} \leq \beta_i \leq \overline{\beta_i} \quad i = 1, \ldots, n_x \end{aligned} \tag{2.29}$$

and

$$\begin{aligned} \overline{\theta}_i \quad = \max_{\theta,\alpha,\beta,T} \quad & \theta_i \\ \text{subject to} \quad & \mathcal{A}^{obs}(\alpha) T = T A(\theta) \\ & \mathcal{B}^{obs}(\beta) = T B(\theta) \\ & \begin{bmatrix} 0 & \ldots & 0 & 1 \end{bmatrix} T = C(\theta) \\ & \underline{\alpha_i} \leq \alpha_i \leq \overline{\alpha_i} \quad i = 1, \ldots, n_x \\ & \underline{\beta_i} \leq \beta_i \leq \overline{\beta_i} \quad i = 1, \ldots, n_x \end{aligned} \tag{2.30}$$

Where constraints here written in matrix form for the aim of compactness, has to be replaced by (2.23), (2.24), (2.26) and (2.28). This is actually fundamental when coding the problem into a POP solver which is not able to handle directly matrix representations, such as sparsePOP.
From the comparison of POPs (2.11) with (2.29) and (2.12) with (2.30), we may also remark that the number of variables involved in the optimization procedures falls from $2n_x^2 + 2n_x + D$ to $n_x^2 + 2n_x + D$, therefore the solution is numerically more efficient.

2.6 One-step procedure

At this stage one can realize that, to make the long story short, what we have proposed in order to solve this problem is to adopt a multi-step procedure when two different but linked polynomial optimization problems are solved, one after the other. In particular the parameters uncertainty intervals (PUIs) on the transfer function parameters $\theta^B = (\alpha, \beta)$ are used as bounds constraint in the successive POP.
In other words one can realize that constraints

$$\begin{aligned} \underline{\alpha_i} \leq \alpha_i \leq \overline{\alpha_i} \qquad & i = 1, \ldots, n_x \\ \underline{\beta_i} \leq \beta_i \leq \overline{\beta_i} \qquad & i = 1, \ldots, n_x \end{aligned} \tag{2.31}$$

are equivalent to the intersection of all constraints of the first POP:

$$\begin{aligned} & \tilde{y}(k) - \eta(k) + \sum_{i=0}^{n-1} \alpha_i \tilde{y}(k-i) + \alpha_i \eta(k-i) = \\ & \quad = \sum_{i=1}^{n} \beta_i \tilde{u}(k-i) + \beta_i \epsilon(k-i) \qquad k = n_x + 1, \ldots, N \\ & \underline{\Delta_\eta}(k) \leq \eta(k) \leq \overline{\Delta_\eta}(k) \qquad k = 1, \ldots, N \\ & \underline{\Delta_\epsilon}(k) \leq \epsilon(k) \leq \overline{\Delta_\epsilon}(k) \qquad k = 1, \ldots, N \end{aligned} \tag{2.32}$$

Therefore it could be a good idea to replace constraints (2.31) on α and β in the second POP with constraints (2.32).
By means of this observation, we are able to solve the problem by means of just one class of POPs, which are

$$\begin{aligned} & \underline{\theta_d} = \min_{\theta, \alpha, \beta, \eta, \epsilon, T} \theta_d \qquad d = 1, \ldots, D \\ & \text{subject to} \\ & \mathcal{A}(\alpha) T = T A(\theta) \\ & \mathcal{B}(\beta) = T B(\theta) \\ & \begin{bmatrix} 0 & \ldots & 0 & 1 \end{bmatrix} T = C(\theta) \\ & \tilde{y}(k) = \eta(k) + \sum_{j=1}^{n_x} \beta_j \tilde{u}(k-j) - \sum_{j=1}^{n_x} \beta_j \epsilon(k-j) + \\ & \quad - \sum_{j=1}^{n_x} \alpha_j \tilde{y}(k-j) + \sum_{j=1}^{n_x} \alpha_j \eta(k-j) \quad k = n_x + 1, \ldots, N \\ & \underline{\Delta_\eta}(k) \leq \eta(k) \leq \overline{\Delta_\eta}(k) \qquad k = 1, \ldots, N \\ & \underline{\Delta_\epsilon}(k) \leq \epsilon(k) \leq \overline{\Delta_\epsilon}(k) \qquad k = 1, \ldots, N \end{aligned} \tag{2.33}$$

and

$$
\begin{aligned}
&\overline{\theta_d} = \max_{\theta,\alpha,\beta,\eta,\epsilon,T} \theta_d = \min_{\theta,\alpha,\beta,\eta,\epsilon,T} -\theta_d \qquad d = 1,\dots,D \\
&\text{subject to} \\
&\mathcal{A}(\alpha)T = TA(\theta) \\
&\mathcal{B}(\beta) = TB(\theta) \\
&\begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} T = C(\theta) \\
&\tilde{y}(k) = \eta(k) + \sum_{j=1}^{n_x} \beta_j \tilde{u}(k-j) - \sum_{j=1}^{n_x} \beta_j \epsilon(k-j) + \\
&\quad - \sum_{j=1}^{n_x} \alpha_j \tilde{y}(k-j) + \sum_{j=1}^{n_x} \alpha_j \eta(k-j) \quad k = n_x+1,\dots,N \\
&\underline{\Delta_\eta}(k) \le \eta(k) \le \overline{\Delta_\eta}(k) \qquad k = 1,\dots,N \\
&\underline{\Delta_\epsilon}(k) \le \epsilon(k) \le \overline{\Delta_\epsilon}(k) \qquad k = 1,\dots,N
\end{aligned}
\tag{2.34}
$$

Above optimizations allows to find the solution without explicitly find the bounds on the black-box transfer function model as an intermediate step.
Moreover, such "1-step" approach also shows two very nice characteristics when compared with the above "2-steps" procedure:

1. **Computational efficiency.** The "2-steps" approach requires at first the solution of $4n_x$ POPs with $2n_x + 2N$ variables which in general can be quite computationally intensive due to the fact that the number of samples N is large, and then the solution of others $2D$ POPs with $2n_x + n_x^2 + D$ variables, that is generally much less intensive due to the relatively low number of variables involved. On the other hand the "1-step" approach requires the solution of $2D$ POPs with $2N + 2n_x + n_x^2 + D$ variables. In this case the number of variables involved is again large and each problem turns out to be approximately as computationally intensive as the fist step of the former approach. The good new is that since generally it holds that

$$D \ll 2n_x$$

   the whole procedure is supposed to be less computationally intensive due to the need of solving much less POPs.

2. **Improved precision**. When solving POPs (2.15) and (2.16) in the two-steps approach, our aim is to give a description of the projection $\mathcal{X}^{(\alpha,\beta)}$ of the feasible set described by (2.32) onto the subspace defined by variables $[\alpha, \beta]$. As it is well known, since such feasible set is described by a set of nonlinear (polynomial) and non-convex equations, then it is also non-convex and what

we need to do when solving the optimization problems is applying convex-relaxation techniques in order to find a outer-approximation $\mathcal{X}_{relaxed}^{(\alpha,\beta)}$ of the desired solution. Moreover, when just describing the projection by means of some intervals on each variables, further conservativeness is added since $\mathcal{X}_{relaxed}^{(\alpha,\beta)}$ is approximated with an hyper-rectangle $\mathcal{X}_{PUI}^{(\alpha,\beta)}$. In a two-dimensional space, the situation can be sketched as in figure 2.1. On the other hand, when

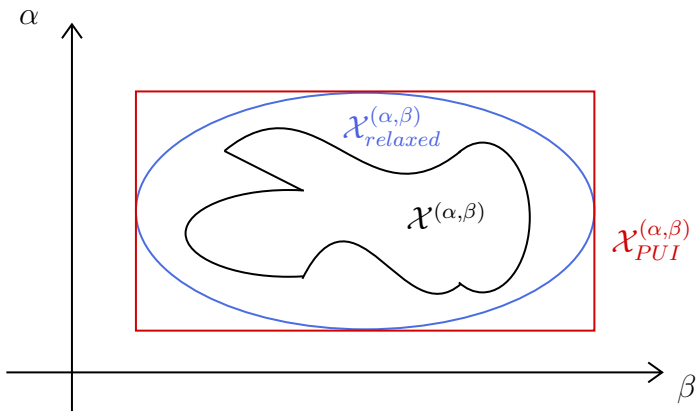

Figure 2.1: Projection of the feasible set of 2.15 and 2.16 into the transfer-function parameters space and his relaxations

solving the 1-step problem, all the information that has been relaxed in the latter procedure is instead retained, and as a consequence more precise results can be obtained (i.e. smaller PUIs on elements of variable θ).

Above considerations allows us to state that this "1-step" approach to the solution of the discrete time gray-box set-membership problem is much more convenient with respect to the two-steps one presented in the previous sections.

2.7 Numerical experiments

In this section a particular problem is presented and numerical results are presented. The discrete-time system described by gray-box model

$$\mathcal{M}_G = \begin{cases} x(k+1) = \begin{bmatrix} -0.22-\theta_2 & \theta_1 \\ 0 & 0.74 \end{bmatrix} x(k) + \begin{bmatrix} 2\theta_2 \\ 3 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 1.1 & 0 \end{bmatrix} x(k) \end{cases}$$

Data are generated my means of a simulation experiment for a time of $N = 50$ samples using values for the unknown parameters

$$\theta = \begin{bmatrix} -0.5 \\ 0.2 \end{bmatrix}$$

and as input $u(k)$ an uniformly distributed white between 0 and 1 random signal has been applied. Then some bounded additive output noise $\eta(k)$ is added to the output samples only (OE noise structure), and the bounds of such noise is chosen in order to obtain tests under different signal-to-noise-ratio (SNR) conditions.
It can be useful to report here also the transfer function corresponding to the true system:

$$H(z) = C(zI_2 - A)^{-1}B = \frac{0.44z - 1.976}{z^2 - 0.32z - 0.3108}$$

The problem is solved by means of both the two-steps and the one-step procedures, and for different SNR evaluated starting from generated signal samples as

$$SNR_{\text{dB}} = 20 \log_{10} \frac{\|y\|_2}{\|\eta\|_2} \tag{2.35}$$

For each parameter results are given in terms of

- absolute value of upper and lower bound as $\overline{\theta_i}$ and $\underline{\theta_i}$ respectively.
- the true value of the parameter $(\theta_i)_{\text{true}}$, in order to allow an immediate check of the correctness of the results
- the central estimate of the parameter, a point-wise estimated defined as

$$(\theta_i)^{\text{central}} = \frac{\overline{\theta_i} - \underline{\theta_i}}{2} \tag{2.36}$$

- relative uncertainty, evaluated through

$$\text{relative uncertainty}_{\%} = \frac{\overline{\theta_i} + \underline{\theta_i}}{(\theta_i)^{central}} \cdot 100 \tag{2.37}$$

Running the optimization in different SNR conditions, we obtained results in tables 2.1, 2.2, 2.3, 2.4, 2.5, 2.6. Time required for obtaining such results are reported in table 2.7.

Results in tables from 2.1 to 2.6 suggest the following observations:

	$\underline{\theta_i^B}$	$\left(\theta_i^B\right)_{\text{true}}$	$\overline{\theta_i^B}$	$\left(\theta_i^B\right)^{\text{central}}$	relative uncertainty$_{\%}$
α_1	-0.339270	-0.320000	-0.303743	-0.321507	11.050194%
α_2	-0.322651	-0.310800	-0.291509	-0.307080	10.141284%
β_1	0.377441	0.440000	0.497520	0.437481	27.447862%
β_2	-2.012101	-1.975600	-1.931779	-1.971940	4.073275%
	$\underline{\theta_i^G}$	$\left(\theta_i^G\right)_{\text{true}}$	$\overline{\theta_i^G}$	$\left(\theta_i^G\right)^{\text{central}}$	relative uncertainty$_{\%}$
θ_1	-0.520568	-0.500000	-0.478820	-0.499694	8.354642%
θ_2	0.180730	0.200000	0.216015	0.198372	17.787486%

Table 2.1: Results for 2-steps solution with SNR = 30 dB OE noise

	$\underline{\theta_i^G}$	$\left(\theta_i^G\right)_{\text{true}}$	$\overline{\theta_i^G}$	$\left(\theta_i^G\right)^{\text{central}}$	relative uncertainty$_{\%}$
θ_1	-0.502489	-0.500000	-0.495045	-0.498767	1.492442%
θ_2	0.190494	0.200000	0.206332	0.198413	7.982059%

Table 2.2: Results for 1-step solution with SNR = 30 dB OE noise

1. Results from the "1-step approach" are significantly better with respect to the ones obtained by means of the "2-steps approach" in terms of relative uncertainty of each parameter. This is true independently from the level of the noise.

2. The "1-step approach" provides his results in about 40 seconds, while the "2-steps approach" requires about 2 minutes in this example. This can be explained by noticing that the computational effort required for finding each bound $\underline{\alpha_i}, \underline{\beta_i}, \overline{\alpha_i}, \overline{\beta_i}$ is approximately the same required for finding bounds $\underline{\theta_i}, \overline{\theta_i}$ since the number of variables involved is almost the same. Then, due to the fact that in the "2-steps approach" I needed to solve 8 optimizations, while in the "1-step" approach I needed to solve half of them, also the required time is (approximately) halved.

3. As an average behavior with increasing level of the noise, the relative uncertainty gets worse both in the black-box model parameters estimations and in the gray-box one. Anyway this is not guaranteed to happen on each parameter since the results also depends on the realization of the noise which is in general different when the noise level changes. As an example, the relative uncertainty on β_1 is better in the 15 dB case with respect to the 20 dB case.

	$\underline{\theta_i^B}$	$\left(\theta_i^B\right)_{\text{true}}$	$\overline{\theta_i^B}$	$\left(\theta_i^B\right)^{\text{central}}$	relative uncertainty$_{\%}$
α_1	-0.349515	-0.320000	-0.152626	-0.251070	78.419832%
α_2	-0.438907	-0.310800	-0.284868	-0.361887	42.565233%
β_1	0.063547	0.440000	0.641093	0.352320	163.926286%
β_2	-2.239820	-1.975600	-1.718828	-1.979324	26.321697%
	$\underline{\theta_i^G}$	$\left(\theta_i^G\right)_{\text{true}}$	$\overline{\theta_i^G}$	$\left(\theta_i^G\right)^{\text{central}}$	relative uncertainty$_{\%}$
θ_1	-0.594627	-0.500000	-0.377097	-0.485862	44.772052%
θ_2	0.170485	0.200000	0.291406	0.230946	52.358862%

Table 2.3: Results for 2-steps solution with SNR = 20 dB OE noise

	$\underline{\theta_i^G}$	$\left(\theta_i^G\right)_{\text{true}}$	$\overline{\theta_i^G}$	$\left(\theta_i^G\right)^{\text{central}}$	relative uncertainty$_{\%}$
θ_1	-0.519869	-0.500000	-0.494729	-0.507299	4.955596%
θ_2	0.193452	0.200000	0.258400	0.225926	28.747620%

Table 2.4: Results for 1-step solution with SNR = 20 dB OE noise

	$\underline{\theta_i^B}$	$\left(\theta_i^B\right)_{\text{true}}$	$\overline{\theta_i^B}$	$\left(\theta_i^B\right)^{\text{central}}$	relative uncertainty$_{\%}$
α_1	-0.528619	-0.320000	-0.227592	-0.378106	79.614556%
α_2	-0.397758	-0.310800	-0.185553	-0.291656	72.758744%
β_1	0.182517	0.440000	0.882945	0.532731	131.478809%
β_2	-2.311370	-1.975600	-1.553103	-1.932237	39.242975%
	$\underline{\theta_i^G}$	$\left(\theta_i^G\right)_{\text{true}}$	$\overline{\theta_i^G}$	$\left(\theta_i^G\right)^{\text{central}}$	relative uncertainty$_{\%}$
θ_1	-0.659487	-0.500000	-0.326383	-0.492935	67.575742%
θ_2	0.082962	0.200000	0.292408	0.187685	111.594284%

Table 2.5: Results for 2-steps solution with SNR = 15 dB OE noise

	$\underline{\theta_i^G}$	$\left(\theta_i^G\right)_{\text{true}}$	$\overline{\theta_i^G}$	$\left(\theta_i^G\right)^{\text{central}}$	relative uncertainty$_{\%}$
θ_1	-0.551245	-0.500000	-0.464214	-0.507729	17.141258%
θ_2	0.119757	0.200000	0.256687	0.188222	72.749547%

Table 2.6: Results for 1-step solution with SNR = 15 dB OE noise

	2-steps	1-step
SNR = 30 dB	105.925 s + 0.939 s	48.171 s
SNR = 20 dB	124.608 s + 0.965 s	35.765 s
SNR = 15 dB	117.725 s + 0.970 s	39.492 s

Table 2.7: Time required for obtaining results of example in section 2.7

Chapter 3

Continuous Time Problem

Most of the systems which can be modeled by means of the so-called first principles of physics are naturally described by differential equations, and therefore as CT gray-box models. For this reason in this chapter we consider the CT gray-box identification problem.
When solving the gray-box identification problem in the SM framework in Chapter 2, the starting point is a reliable solution of the corresponding black-box problem in the SM framework. Similarly, in order to solve the corresponding problem for CT systems, a solution for the CT black-box identification problem in the SM framework is needed.
Moreover, the black-box continuous-time problem is interesting in itself especially as a technique for identification finalized to robust control strategies such as H_∞ optimization or μ-synthesis [25], [26].
For these reasons in this chapter lot of effort is devoted to the black-box CT system identification problem. At first classical solutions are mentioned and discussed, then two new methods based on the formulation of the problem in the SM framework are presented.
Finally, given the results about SM identification of black-box CT models, the solution of the CT gray-box identification problem straight forwardly follows.

3.1 Black-Box Problem Statement

Problem of identifying a continuous time system thanks to sampled data [27] is formulated as follows:
Problem: given a set of samples of input and output signals

$$\{\tilde{y}(k), \tilde{u}(k)\} \qquad k = 1, \ldots, N \tag{3.1}$$

collected out of the true continuous time input and output signals $u(t), y(t)$ by means of a constant rate sampling with sampling period T_s

$$\begin{aligned} u(k) &\doteq u(t = kT_S) \qquad k = 1, \dots, N \\ y(k) &\doteq y(t = kT_S) \qquad k = 1, \dots, N \end{aligned} \tag{3.2}$$

and corrupted by some noise, we want to estimate a transfer function

$$H(s) = \frac{\beta^c_{n-1}s^{n-1} + \cdots + \beta^c_1 s + \beta^c_0}{s^n + \alpha^c_{n-1}s^{n-1} + \cdots + \alpha^c_1 s + \alpha^c_0} = \frac{\sum_{i=0}^{n-1} \beta^c_i s^i}{s^n + \sum_{i=0}^{n-1} \alpha^c_i s^i} \tag{3.3}$$

describing the input-output behaviour of the system.
Although most of the literature about system identification focuses on discrete-time systems, the problem of identifying a CT system by means of sampled data has been studied in the past, as presented in [28] and [27]. As also discussed in such articles, solutions to the problem can be divided in two classes: direct approaches and indirect ones. An overview about such solution is the subject of the following sections.

3.2 Direct Approaches

Direct approaches rely on algorithms specifically designed in order to give an estimate of $H(s)$ directly from sampled data without deriving any discrete-time model. In the following we will present some of the results from this class. A detailed description of all the methods in this family is beyond the objective of this thesis.
The main observation behind direct CT system identification is that the CT transfer function (3.3) links input and output signals by means of

$$\begin{aligned} Y(s) = H(s)U(s) \Rightarrow y(t) + \alpha^c_1 y^{(1)}(t) + \cdots + \alpha^c_n y^{(n)}(t) = \\ = \beta^c_1 u^{(1)}(t) + \cdots + \beta^c_n u^{(n)}(t) \end{aligned} \tag{3.4}$$

where $y^{(i)}(t)$ and $u^{(i)}(t)$ for $i = 1, \dots, n$ represents $i-$th time derivatives of output and input signals respectively. Then given an estimate of the time-derivatives of input and output signals $\hat{u}$ and $\hat{y}$, it is possible to build some loss function

$$\begin{aligned} \mathcal{L} = \sum_{k=1}^{N} (\hat{y}(kT_s) + \alpha_1 \hat{y}^{(1)}(kT_s) + \cdots + \alpha_n \hat{y}^{(n)}(kT_s) + \\ - \beta_1 \hat{u}^{(1)}(kT_s) - \cdots - \beta_n \hat{u}^{(n)}(kT_s))^2 \end{aligned} \tag{3.5}$$

and formulate the optimization problem

$$(\theta^{B,c})^{opt} = \arg\min_{\theta^{B,c}} \mathcal{L} \qquad \theta^{B,c} = [\alpha^c, \beta^c] \tag{3.6}$$

which is a convex quadratic program that can be easily minimized by any standard solver. Alternatively, since the problem can also be recognized as a standard least squares problem, it can also be solved by means of the normal system of equations

$$(\theta^{B,c})^{opt} = (\Phi^\top \Phi)^{-1} \Phi^\top (-\tilde{y}) \tag{3.7}$$

where

$$\Phi = \left[\hat{y}^{(1)}, \dots, \hat{y}^{(n)}, \hat{u}^{(1)}, \dots, \hat{u}^{(n)}\right] \tag{3.8}$$

Estimation of time-derivatives of input and output signals is usually done by means of the application of a suitably-tuned filter $F_i(s)$ to the input and output signals as $X^{(i)}(s) \approx F_i(s)X(s)$. This can be interpreted as a numerical differentiation pre-processing operation. A simple choice consists in the application of some LTI filter as

$$F_i(s) = \left(\frac{s}{1+s\tau}\right)^i \tag{3.9}$$

which takes the ideal "multi-differentiator" filter $F_i^{ideal}(s) = s^i$ and adds some poles at high frequency to make it physically realizable. Such method, introduces approximation errors in the estimation of the derivatives. In any case it is possible to modify this problem, in order to avoid approximation errors.
In [29], the following idea is proposed. A causal linear operator is defined and used in order to replace the time-derivative, therefore avoiding the approximation problem. Such operator can be chosen to be for instance the low-pass

$$\lambda = F(s) = \frac{1}{1+s\tau} \quad \Rightarrow \quad s = \frac{1-\lambda}{\lambda\tau} \tag{3.10}$$

which allows to transform the initial transfer function $H(s|\alpha^c, \beta^c)$ in (3.3) into a new input-output model $H_0(\lambda|\alpha^c, \beta^c, \tau)$ which now explains input-output mapping in terms of λ operator. Then, it is now possible to evaluate filtered signals $\lambda^i u(t)$ and $\lambda^i y(t)$ and construct some loss function to me minimized thanks to which estimates of the original transfer function parameters $\hat{\alpha^c}, \hat{\beta^c}$ can be determined. Such approach is referred to as the State Variable Filter (SVF) approach and was developed by Young in 1969.
Other methods based on the very same idea of the application of analog filter, are:

- Generalized Poisson Moment Functionals (GPMF) approach: very similar to the SVF above, uses filters

$$F_i(s) = \left(\frac{b}{s+a}\right)^{i+1} \qquad a, b \text{ tunable parameters} \tag{3.11}$$

- Integral Methods: based on the idea of avoiding issues related to the differentiation performing n consecutive integrations. Among such family of methods, it is possible to find the Integral Method over a Stretched Window that can be interpreted as the application of

$$F_i(s) = s^{-i} \tag{3.12}$$

  or the integral method over a moving window (LIF) that can be interpreted as the application of

$$F_i(s) = \left(\frac{1 - e^{-lT_s s}}{s}\right)^i \tag{3.13}$$

  which has the advantage with respect to the previous integral method of making initial condition disappear.

- Modulating Functions approach: firstly proposed by Shinbrot in 1957, the filtering $F(s)$ corresponds to the multiplication by both sides of equation (3.4) by some function called modulating function. Then the derivative operator is applied to such functions and integration over the period is performed different. The usage of many different modulating functions has been studied, as instance Fourier basis functions, Hermite functions and Hartley-based functions. Advantages of such approach are that thanks to Fourier Hartley-based functions the problem can be posed in frequency domain and that the of effect of initial conditions disappears.

Since the analysis of direct approaches to continuous-time system identification is beyond the scope of this thesis, we do not enter into more details. Instead, it is useful to make some quite general considerations about all such approaches.
The analysis of the above mentioned algorithms is carried out either in the noise-free and noisy case. In this last case, the noise is modeled as an additive Gaussian random variable in the equation (3.4). This assumption allows to prove results about convergence of the solution when the estimate for the system parameters is evaluated with some least squares approach as in (3.6) and (3.7).
Another important feature is that all such derivations are formally true when input and output signals $u(t), y(t)$ are filtered by means of some analog filter with the given transfer function $F(s)$. In practice this is a problematic point, in fact

- If it is possible to include some analog device during the experiment, then one could build the filters $F_i(s)$ thanks to some operational-amplifier based circuit and collect outputs signals $\lambda^i u(t)$ and $\lambda^i y(t)$ with the same sampling frequency used for non-filtered input and output. For example, considering the filtering of $u(t)$ in (3.10), the filter can be implemented as sketched in Figure 3.1.

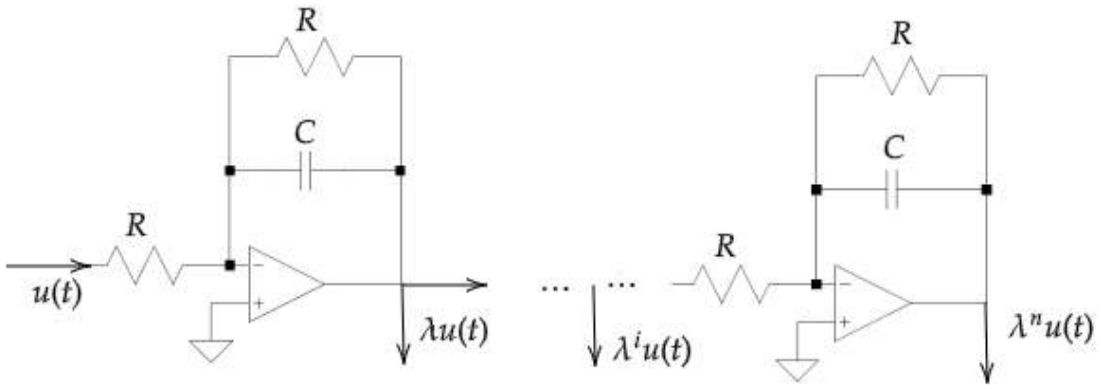

Figure 3.1: Analog implementation for filters $F_i(s)$ in SVF approach

In the same way we obtain $\lambda^i y(t)$ when feeding $y(t)$ as input of the filter in Figure 3.1.
Such solution, even if reasonable and apparently correct, presents a number of factors introducing errors in the process

- Resistor's and (even worse) capacitor's tolerances does not allow to get the very same τ in all the stages of the filter. This problem can be attenuated choosing components with small tolerances and from the same batch, but never eliminated.
- Non-ideality of the operational amplifiers introduces additional errors such as offset voltages due to input bias currents, offset input differential voltage, finite bandwidth and finite gain...

Moreover additional costs are added to the design due to the need of realizing the filters and including sample and hold devices for the sampling of each filtered signal.

- If we assume instead, that the only data we can collect are samples of input and output signals $u(kT_s), y(kT_s)$, then the only way to get estimates of $\lambda^i u(kT_s)$ and $\lambda^i y(kT_s)$ is by means of the application of some digital filter $F(z)$ obtained by discretization of the analog $F(s)$.

$$\lambda^i x(k) = F_i(s)x(t)|_{t=kT_s} \approx \mathcal{Z}\{F_i(s)\}x(k) = F_i(z)x(k) \tag{3.14}$$

Also this approach introduces some error, this time entirely due to the discretization. But the good new is that differently from the error arising in the analog case, this one is proved to tend to zero as the sampling frequency tends to infinity. In practice we can assume that this error is very small when the sampling frequency is much larger than both the bandwidth of the signals, of the system to be identified and that of the filters.

This is the reason why, in Section 3.5, the problem is tackled in a SM framework by estimating filtered signals through (3.14).

3.3 Indirect Approaches

Indirect approaches are based on the idea of estimating a discrete-time model my means of any discrete-time system identification algorithm, as

- classical algorithms for the estimation of discrete-time transfer function models: ARX, ARMAX, OE, BJ [ref ??];
- subspace identification based algorithms, such as N4SID [30];
- the SM approach presented in 2.5.

Afterwards, the obtained discrete-time model is converted into an equivalent continuous-time model. This last step is more crucial due to the fact the conversion from discrete-time and continuous-time is not unique. For this reason we now focus on the main techniques allowing to perform such a conversion.
Conversion of a discrete-time system into a continuous-time one is the inverse problem of the well-known discretization problem. Discretization of a continuous-time system can be formalized as
Problem: Given a continuous-time system $G^c(s)$, we need to find some discrete-time system $G^d(z)$ such that the input-output mapping is preserved.
Such analysis has been carried out under many assumptions, bringing to different solutions

- Forward Euler discretization: is maybe the simplest idea in discretization of differential equations. The differentiated signals are approximated by their Taylor series expansion truncated at the first order. For a signal $x(t)$ we have

$$\begin{aligned} x(t_0 + T_s) &= x(t_0) + T_s \left.\frac{dx(t)}{dt}\right|_{t=t_0} + \mathcal{O}(T_s^2) \\ &\approx x(t_0) + T_s \left.\frac{dx(t)}{dt}\right|_{t=t_0} \end{aligned} \tag{3.15}$$

As a consequence

$$\frac{dx(t)}{dt} \approx \frac{x((k+1)T_s) - x(kT_s)}{T_s} \tag{3.16}$$

Now, applying the Laplace-transform of the left-hand side and the $\mathcal{Z}$-transform on the right-hand side suggests the substitution

$$s = \frac{z-1}{T_s} \tag{3.17}$$

A geometrical interpretation for this approach can be found when thinking about the integrator

$$y((k+1)T_s) = y(kT_s) + \int_{t_0}^{t_0+T_s} u(t) \xrightarrow{\mathcal{L}} Y(s) = \frac{U(s)}{s} \tag{3.18}$$

In this case, applying Forward Euler discretization brings to

$$Y(z) = \frac{T_s}{z-1}U(z) \xrightarrow{\mathcal{Z}^{-1}} y((k+1)T_s) = y(kT_s) + T_s u(kT_s) \tag{3.19}$$

This means that the approximation is error-free only when the input signal is constant over the sampling interval, since from a time-domain point of view it is assumed that

$$\int_{t_0}^{t_0+T_s} u(t) \approx T_s u(kT_s) \tag{3.20}$$

- Backward Euler discretization: consists into a technique very similar to the forward Euler one, where the integrator is approximated with

$$y((k+1)T_s) = y(kT_s) + \int_{t_0}^{t_0+T_s} u(t) \approx y(kT_s) + T_s u((k+1)T_s) \tag{3.21}$$

Notice that in this case it is assumed

$$\int_{t_0}^{t_0+T_s} u(t) \approx T_s u((k+1)T_s) \tag{3.22}$$

which is a very similar assumption with respect to the latter technique, and brings to the substitution

$$s = \frac{z-1}{zT_s} = \frac{1-z^{-1}}{T_s} \tag{3.23}$$

- Tustin's discretization: is based on the idea of approximating the integral between two samples with a trapezoidal approximation, in formulas:

$$\begin{aligned} y((k+1)T_s) &= y(kT_s) + \int_{t_0}^{t_0+T_s} u(t) \\ &\approx y(kT_s) + T_s \frac{(u(kT_s) + u((k+1)T_s))}{2} \end{aligned} \tag{3.24}$$

In this case, the discretization is therefore performed by means of the substitution

$$s = \frac{2}{T_s}\frac{z-1}{z+1} \tag{3.25}$$

In most cases, such approximation brings to better results with respect to the previous ones for the same sampling time T_s, as long as the signals involved in the description of the evolution of the system are smooth enough. A graphical comparison between approximation of the integrals computations in Euler methods and Tustin one is presented in figure 3.2

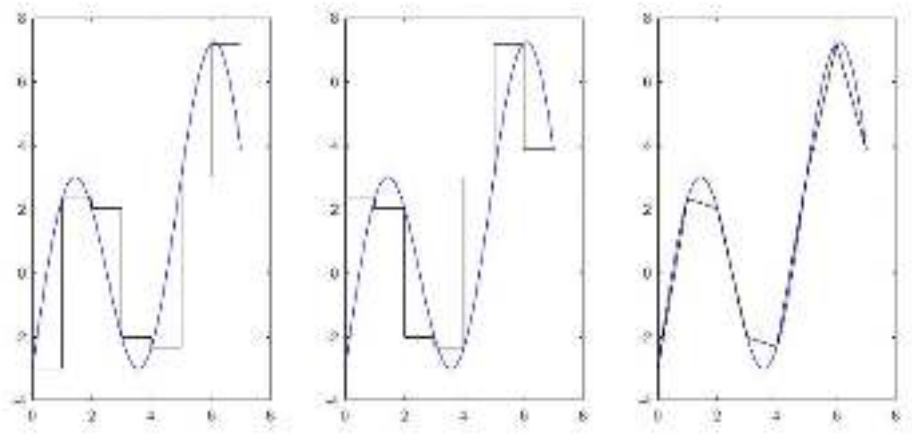

Figure 3.2: Comparison between approximation introduced by forward Euler method (left), backward Euler method (center) and Tustin method (right)

- ZOH. This assumption is based on the practical case of the digital design of a control system for a continuous-time plant. In such a circumstance the input signal for the plant to be discretized is a discrete-time signal coming from the digital controller with sampling time T_s and is filtered by a zero-order-hold (ZOH) device, characterized by

$$h_{ZOH}(t) = \epsilon(t) - \epsilon(t - T_s) \Rightarrow G_{ZOH}(s) = \frac{1 - e^{-sT_s}}{s} \tag{3.26}$$

where $\epsilon(t)$ is the Heaviside step function.
A block-diagram description of this setting is shown in figure 3.3.

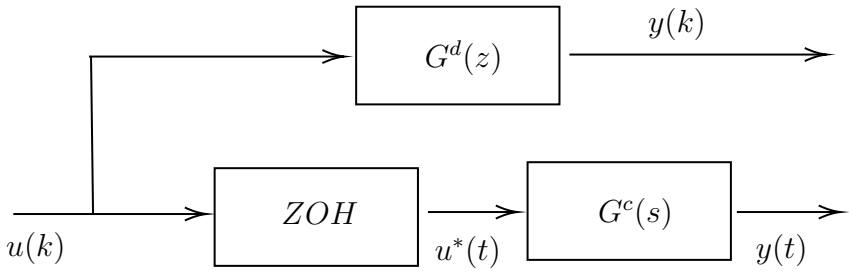

Figure 3.3: Block-diagram of ZOH discretization

In this situation, the discrete-time model $G^d(z)$ can be obtained as

$$\begin{aligned} G^d(z) &= \mathcal{Z}\left\{G^c(s)G_{ZOH}(s)\right\} = \mathcal{Z}\left\{G^c(s)\frac{1-e^{-sT_s}}{s}\right\} = \\ &= \mathcal{Z}\left\{\frac{G^c(s)}{s}\right\} - \mathcal{Z}\left\{\frac{G^c(s)}{s}e^{-sT_s}\right\} = (1-z^{-1})\mathcal{Z}\left\{\frac{G^c(s)}{s}\right\} \end{aligned} \tag{3.27}$$

- Matched: the method consists into mapping each pole and zero in the s-domain of the original CT transfer function into the corresponding pole and zero in the z-plane through the mapping

$$z_i = e^{s_i T_s} \tag{3.28}$$

This method has the advantage of preserving quite well the frequency behaviour of the system. For this reason it is mainly used to convert models for controllers designed in continuous-time into discrete-time controllers in control systems design.

3.4 SM Problem Statement

In the following, we will deal with the CT black-box identification problem in a SM framework. For this reason it is necessary to give further details about the statement of the problem in such context. Firstly, the following assumption on the noise is performed:
Assumption the samples of the noise affecting the input and output signals are bounded as

$$\begin{aligned} \underline{\Delta_\eta}(k) \leq \eta(k) \leq \overline{\Delta_\eta}(k) \\ \underline{\Delta_\epsilon}(k) \leq \epsilon(k) \leq \overline{\Delta_\epsilon}(k) \end{aligned} \tag{3.29}$$

and enters the problem according to the EIV noise structure, represented in figure 3.4

$$\begin{aligned} \tilde{u}(k) &= u(k) + \epsilon(k) \qquad k = 1, \ldots, N \\ \tilde{y}(k) &= y(k) + \eta(k) \qquad k = 1, \ldots, N \end{aligned} \tag{3.30}$$

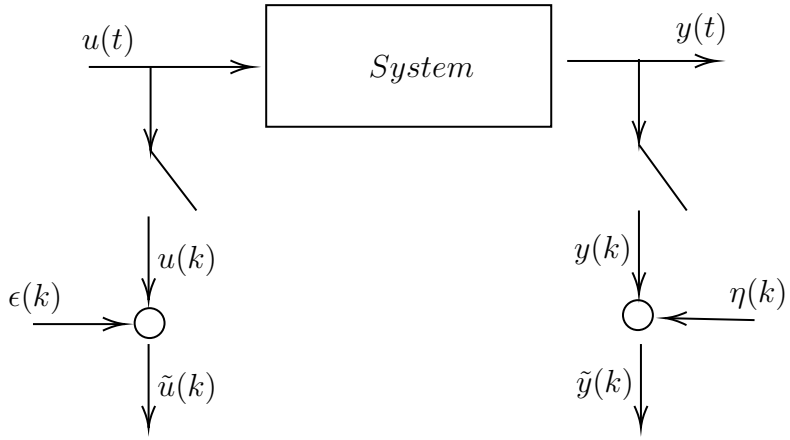

Figure 3.4: Block diagram of CT EIV noise structure

Under such condition, we are interested in finding bounds $\underline{\alpha_i^c}, \overline{\alpha_i^c}, \underline{\beta_i^c}, \overline{\beta_i^c}$ on the parameters α^c, β^c of the continuous-time transfer function (3.3).
In Sections 3.5 and 3.6, two solutions of this problem are proposed. Both of them are based on ideas from classical approaches discussed in the previous sections and are formulated as sets of polynomial optimization problems.

3.5 State Variable Filters SM approach

A possible idea is to establish a procedure based on the SVF technique, as presented in Section 3.2. In this setting, the main equation relating samples from the involved signals and the system parameters is

$$y(t) = H_0(\lambda|\alpha^c, \beta^c, \tau)u(t) \tag{3.31}$$

where $H_0(\lambda|\alpha^c, \beta^c, \tau)$ is obtained from $H(s|\alpha^c, \beta^c)$ by means of the substitution (3.10).
Let us now enter into more details about the shape of such $H_0(\lambda)$ in the (almost general) case when the continuous-time transfer function is a strictly proper transfer function of order n.

$$H(s|\alpha^c, \beta^c) = \frac{\sum_{i=0}^{n-1} \beta_i^c s^i}{s^n + \sum_{i=0}^{n-1} \alpha_i^c s^i} \tag{3.32}$$

Substituting the differential operator s with the low-pass filtering operator λ according to

$$s = \frac{1-\lambda}{\lambda\tau} \tag{3.33}$$

we get

$$\begin{aligned} H_0(\lambda|\alpha^c,\beta^c,\tau) &= \frac{\sum_{i=0}^{n-1}\beta_i^c\left(\frac{1-\lambda}{\lambda\tau}\right)^i}{\left(\frac{1-\lambda}{\lambda\tau}\right)^n + \sum_{i=0}^{n-1}\alpha_i^c\left(\frac{1-\lambda}{\lambda\tau}\right)^i} \\ &= \frac{\sum_{i=0}^{n-1}\beta_i^c\,(1-\lambda)^i\,(\lambda\tau)^{n-i}}{(1-\lambda)^n + \sum_{i=0}^{n-1}\alpha_i^c\,(1-\lambda)^i\,(\lambda\tau)^{n-i}} \\ &= \frac{\sum_{j=1}^{n}\tilde{b}_j\lambda^j}{1+\sum_{j=1}^{n}\tilde{a}_j\lambda^j} = H_0(\lambda|\tilde{a},\tilde{b}) \end{aligned} \tag{3.34}$$

where $\bar{a_i}$ and $\bar{b_i}$ are linked to α^c, β^c and τ. In order to get a closed form expression for this relationship, we need to focus on the expression

$$\sum_{i=0}^{n-1}\gamma_i\,(1-\lambda)^i\,(\lambda\tau)^{n-i} = \sum_{j=1}^{n}\tilde{c}_j\lambda^j \tag{3.35}$$

In fact, when γ is set to β^c we get the expression of the numerator, while when γ is set to α^c we the main term involved in the expression of the denominator.
We notice that $\bar{a_i}$ and $\bar{b_i}$, are the coefficients of a polynomial of order n, which is given as the product between many monomials. Then recall that coefficients of a polynomial $p(x) = \sum_i p_i x^i$ which can be written as the product of other two polynomials $q(x) = \sum_i q_i x^i$ and $r(x) = \sum_i r_i x^i$ can be obtained my means of the discrete convolution operation between vectors containing the coefficients of $q(x)$ and $r(x)$, as

$$p = q * r \qquad \text{i.e.} \quad p_i = \sum_{k=0}^{n} q_k r_{i-k} \tag{3.36}$$

Therefore, in this particular problem, it can be useful to introduce the vectors

$$c^{(i)} = \underbrace{\begin{bmatrix}-1\\1\end{bmatrix} * \cdots * \begin{bmatrix}-1\\1\end{bmatrix}}_{i \text{ times}} \underbrace{\begin{bmatrix}1\\0\end{bmatrix} * \cdots * \begin{bmatrix}1\\0\end{bmatrix}}_{n-i \text{ times}} \in \mathbb{R}^n \tag{3.37}$$

that corresponds to the coefficients of the i-th polynomial. In such vectors, the element $n-j$ represents the coefficient of λ^j in the corresponding polynomial.

Thanks to the definition of these vectors, we re-write the left-hand side of expression (3.35) as

$$\sum_{i=0}^{n-1} \gamma_i \tau^{n-i} \sum_{j=1}^{n} c_{n-j}^{(i)} \lambda^j \tag{3.38}$$

Then, thanks to the linearity of the summation operator, this expression can be further re-written as

$$\sum_{j=1}^{n} \sum_{i=0}^{n-1} \gamma_i \tau^{n-i} c_{n-j}^{(i)} \lambda^j \tag{3.39}$$

Comparing this last expression with the right-hand side of (3.35), it is finally possible to state that

$$\tilde{c}_j = \sum_{i=0}^{n-1} \gamma_i \tau^{n-i} c_{n-j}^{(i)} \tag{3.40}$$

and in particular

$$\tilde{b}_j = \tilde{b}_j(\beta^c) = \sum_{i=0}^{n-1} \beta_i^c \tau^{n-i} c_{n-j}^{(i)} \tag{3.41}$$

Moreover, since the coefficients of polynomial $(1-\lambda)^n$ appearing at the denominator are computed as

$$\underbrace{\begin{bmatrix} -1 \\ 1 \end{bmatrix} * \cdots * \begin{bmatrix} -1 \\ 1 \end{bmatrix}}_{n \text{ times}} = c^{(n)} \tag{3.42}$$

then

$$\tilde{a}_j = \tilde{a}_j(\alpha^c) = c_{n-j}^{(n)} + \sum_{i=0}^{n-1} \alpha_i^c \tau^{n-i} c_{n-j}^{(i)} \tag{3.43}$$

From equations (3.41) and (3.43), we notice that the parameters $\tilde{a}$ and $\tilde{b}$ appearing in $H_0(\lambda|\tilde{a}, \tilde{b})$ are linearly dependent from the parameters α^c, β^c we want to estimate. As a consequence it is easy to lead back the problem of finding upper and lower bounds on α^c, β^c to the problem of finding upper and lower bounds on $\tilde{a}$ and $\tilde{b}$. Next, in order to estimate upper and lower bounds on $\tilde{a}$ and $\tilde{b}$, we adopt a procedure inspired from the solution of the DT identification problem in the SM framework. In fact, given samples $u(k), y(k)$ of signals $u(t), y(t)$ and samples of their filtered versions

$$\begin{aligned} u^{(i)}(k) &\doteq \lambda^i u(t)\Big|_{t=kT_s} \\ y^{(i)}(k) &\doteq \lambda^i y(t)\Big|_{t=kT_s} \end{aligned} \tag{3.44}$$

it holds the equation

$$\begin{aligned} y(t) &= H_0(\lambda|\tilde{a}, \tilde{b}) u(t) \\ &\Rightarrow y(t) + \sum_{j=1}^{n} \tilde{a_j} \lambda^j y(t) = \sum_{j=1}^{n} \tilde{b_j} \lambda^j u(t) \qquad \forall t \end{aligned} \tag{3.45}$$

Since this equation holds for any time, then in particular it must be true also at any sampling instant. Then we can formulate optimization problems

$$\begin{aligned} &\underline{\tilde{\theta}_j} = \min \tilde{\theta}_j \qquad \tilde{\theta} = [\tilde{a}, \tilde{b}] \\ &\text{subject to} \\ &y(k) + \sum_{j=1}^{n} \tilde{a}_j y^{(i)}(k) = \sum_{j=1}^{n} \tilde{b}_j u^{(i)}(k) \qquad k = 1, \ldots, N \end{aligned} \tag{3.46}$$

$$\begin{aligned} &\overline{\tilde{\theta}_j} = \max \tilde{\theta}_j \qquad \tilde{\theta} = [\tilde{a}, \tilde{b}] \\ &\text{subject to} \\ &y(k) + \sum_{j=1}^{n} \tilde{a}_j y^{(i)}(k) = \sum_{j=1}^{n} \tilde{b}_j u^{(i)}(k) \qquad k = 1, \ldots, N \end{aligned} \tag{3.47}$$

Still it is not possible to solve the formulated optimization problems since the only available data are noisy samples $\tilde{u}(t)$ and $\tilde{y}(t)$ of $u(t)$ and $y(t)$ only, together with bounds on the noise affecting them, while filtered signals are not available. How to find such samples of filtered signals has already been discussed in 3.2 and now an analysis of errors affecting the samples in the analog filters implementation versus the digital filtering one is performed.

1. Analog filtering would in principle be the better option as long as we were able to build $2n$ filters all exactly equal in terms of transfer function they implement. As already discussed this is not possible in practice and moreover due to the non-ideality of components building the filters some additional error is introduced, so in such a condition a realistic model for sampled values we would measure could be

$$\begin{aligned} \tilde{y}^{(i)}(k) &= F_i(s)\tilde{y}(t)|_{t=kT_s} + \eta^{(i)}(k) \\ &= F_i(s)y(t)|_{t=kT_s} + F_i(s)\eta(t)|_{t=kT_s} + \eta^{(i)}(k) \end{aligned} \tag{3.48}$$

   Therefore the total error affecting the sample would be the sum of a component coming from the original noise affecting the signal we want to filter plus another component due to the non-ideal filtering itself, and as the reader may guess, bounds on errors $F_i(s)\eta(t)|_{t=kT_s} + \eta^{(i)}(k)$ either are not easy to find and in principle can be larger than the original Δ_η.

2. Usage of a digital filter $F(z)$ obtained through some discretization of $F(s)$ is an effective alternative. Such approach introduces an error $\delta(k)$ due to the discretization as

$$y^{(i)}(k) = F(z)y^{(i-1)}(k) + \delta(k) \tag{3.49}$$

   where

$$F(z) = \mathcal{Z}\{F(s)\} = \frac{f_1 + f_2 z^{-1}}{1 + f_3 z^{-1}} \tag{3.50}$$

Combining equations (3.49) and (3.50), we discover that samples of filtered and non-filtered signals are related by a simple linear relation

$$y^{(i)}(k) + f_3 y^{(i)}(k-1) = f_1 y^{(i-1)}(k) + f_2 y^{(i-1)}(k-1) + \delta(k) \tag{3.51}$$

and this hold for all $i = 0, \dots, n$ where $y^{(0)}(k) \doteq y(k)$.
Even more importantly, the uncertainty due to $\delta(k)$ can be neglected with respect to the one introduced by the noises $\eta(k), \epsilon(k)$ as long as the sampling time is small enough. This is true because when the sampling frequency approaches infinity, the DT filters $F(z)$ are able to reproduce the output of the ideal CT signals applied to the true signals.

For all the reasons above, we can assume that all information about the samples of filtered signals can be obtained thanks to the linear equations

$$\begin{aligned} y^{(i)}(k) + f_3 y^{(i)}(k-1) &= f_1 y^{(i-1)}(k) + f_2 y^{(i-1)}(k-1) \\ u^{(i)}(k) + f_3 u^{(i)}(k-1) &= f_1 u^{(i-1)}(k) + f_2 u^{(i-1)}(k-1) \end{aligned} \tag{3.52}$$

where $i = 0, \dots, n$ and $k = 1, \dots, N$.
Therefore, bounds on $\tilde{a}$ and $\tilde{b}$ are obtained by the solution of optimization problems

$$\begin{aligned} \underline{\tilde{\theta}_j} = \min_{\tilde{a},\tilde{b},y,\dots,y^{(n)},u,\dots,u^{(n)}} & \tilde{\theta}_j \qquad \tilde{\theta} = [\tilde{a}, \tilde{b}] \\ \text{subject to} & \\ y(k) + \sum_{j=1}^{n} \tilde{a}_j y^{(j)}(k) &= \sum_{j=1}^{n} \tilde{b}_j u^{(j)}(k) \qquad k = n+1, \dots, N \\ y^{(j)}(k) + f_3 y^{(j)}(k-1) &= f_1 y^{(j-1)}(k) + f_2 y^{(j-1)}(k-1) \\ j = 1, \dots, n, \quad k &= 1, \dots, N \\ u^{(j)}(k) + f_3 u^{(j)}(k-1) &= f_1 u^{(j-1)}(k) + f_2 u^{(j-1)}(k-1) \\ j = 1, \dots, n, \quad k &= 1, \dots, N \\ y(k) - \Delta_\eta \le y(k) \le y(k) + \Delta_\eta & \qquad k = 1, \dots, N \\ u(k) - \Delta_\epsilon \le u(k) \le u(k) + \Delta_\epsilon & \qquad k = 1, \dots, N \end{aligned} \tag{3.53}$$

$$
\begin{aligned}
&\overline{\tilde{\theta}_j} = \max_{\tilde{a},\tilde{b},y,\ldots,y^{(n)},u,\ldots,u^{(n)}} \tilde{\theta}_j \qquad \tilde{\theta} = [\tilde{a},\tilde{b}] \\
&\text{subject to} \\
&y(k) + \sum_{j=1}^{n} \tilde{a}_j y^{(j)}(k) = \sum_{j=1}^{n} \tilde{b}_j u^{(j)}(k) \qquad k = n+1,\ldots,N \\
&y^{(j)}(k) + f_3 y^{(j)}(k-1) = f_1 y^{(j-1)}(k) + f_2 y^{(j-1)}(k-1) \\
&\qquad j = 1,\ldots,n, \quad k = 1,\ldots,N \\
&u^{(j)}(k) + f_3 u^{(j)}(k-1) = f_1 u^{(j-1)}(k) + f_2 u^{(j-1)}(k-1) \\
&\qquad j = 1,\ldots,n, \quad k = 1,\ldots,N \\
&y(k) - \Delta_\eta \le y(k) \le y(k) + \Delta_\eta \qquad k = 1,\ldots,N \\
&u(k) - \Delta_\epsilon \le u(k) \le u(k) + \Delta_\epsilon \qquad k = 1,\ldots,N
\end{aligned}
\tag{3.54}
$$

Finally, we can further modify the optimization problems including variables α^c and β^c and constraints (3.41) and (3.43) in order to directly solve for α^c and β^c, which is our final objective.

$$
\begin{aligned}
&\underline{\tilde{\theta}_j} = \min_{\alpha^c,\beta^c,\tilde{a},\tilde{b},y,\ldots,y^{(n)},u,\ldots,u^{(n)}} \tilde{\theta}_j \qquad \tilde{\theta} = [\alpha^c,\beta^c] \\
&\text{subject to} \\
&\tilde{b}_j = \sum_{i=0}^{n-1} \beta_i^c \tau^{n-i} c_{n-j}^{(i)} \qquad j = 1,\ldots,n \\
&\tilde{a}_j = c_{n-j}^{(n)} + \sum_{i=0}^{n-1} \alpha_i^c \tau^{n-i} c_{n-j}^{(i)} \qquad j = 1,\ldots,n \\
&y(k) + \sum_{j=1}^{n} \tilde{a}_j y^{(j)}(k) = \sum_{j=1}^{n} \tilde{b}_j u^{(j)}(k) \qquad k = n+1,\ldots,N \\
&y^{(j)}(k) + f_3 y^{(j)}(k-1) = f_1 y^{(j-1)}(k) + f_2 y^{(j-1)}(k-1) \\
&\qquad j = 1,\ldots,n, \quad k = 1,\ldots,N \\
&u^{(j)}(k) + f_3 u^{(j)}(k-1) = f_1 u^{(j-1)}(k) + f_2 u^{(j-1)}(k-1) \\
&\qquad j = 1,\ldots,n, \quad k = 1,\ldots,N \\
&y(k) - \Delta_\eta \le y(k) \le y(k) + \Delta_\eta \qquad k = 1,\ldots,N \\
&u(k) - \Delta_\epsilon \le u(k) \le u(k) + \Delta_\epsilon \qquad k = 1,\ldots,N
\end{aligned}
\tag{3.55}
$$

and

$$
\begin{aligned}
\underline{\tilde{\theta}_j} = & \max_{\alpha^c,\beta^c,\tilde{a},\tilde{b},y,\ldots,y^{(n)},u,\ldots,u^{(n)}} \tilde{\theta}_j \qquad \tilde{\theta} = [\alpha^c, \beta^c] \\
& \text{subject to} \\
& \tilde{b}_j = \sum_{i=0}^{n-1} \beta_i^c \tau^{n-i} c_{n-j}^{(i)} \qquad j = 1, \ldots, n \\
& \tilde{a}_j = c_{n-j}^{(n)} + \sum_{i=0}^{n-1} \alpha_i^c \tau^{n-i} c_{n-j}^{(i)} \qquad j = 1, \ldots, n \\
& y(k) + \sum_{j=1}^{n} \tilde{a}_j y^{(j)}(k) = \sum_{j=1}^{n} \tilde{b}_j u^{(j)}(k) \qquad k = n+1, \ldots, N \\
& y^{(j)}(k) + f_3 y^{(j)}(k-1) = f_1 y^{(j-1)}(k) + f_2 y^{(j-1)}(k-1) \\
& \qquad j = 1, \ldots, n, \quad k = 1, \ldots, N \\
& u^{(j)}(k) + f_3 u^{(j)}(k-1) = f_1 u^{(j-1)}(k) + f_2 u^{(j-1)}(k-1) \\
& \qquad j = 1, \ldots, n, \quad k = 1, \ldots, N \\
& y(k) - \Delta_\eta \leq y(k) \leq y(k) + \Delta_\eta \qquad k = 1, \ldots, N \\
& u(k) - \Delta_\epsilon \leq u(k) \leq u(k) + \Delta_\epsilon \qquad k = 1, \ldots, N
\end{aligned}
\tag{3.56}
$$

Remarks

- Optimization problems (3.55) and (3.56) are polynomial optimization problems (POP) due to the fact that equations

$$
y(k) + \sum_{j=1}^{n} \tilde{a}_j y^{(j)}(k) = \sum_{j=1}^{n} \tilde{b}_j u^{(j)}(k) \tag{3.57}
$$

  involves products between optimization variables. In particular this POP has degree 2 and so can be solved by means of SDP relaxation techniques with order of relaxation starting from $\delta^{min} = 1$.

- The procedure based on the solution of POPs (3.41) and (3.43) is able to provide good results thanks to the fact that the only uncertainty is due to the noise on the original noise affected samples $u(k), y(k)$. All the samples of filtered signals are assumed to be uncertain but strictly correlated with realization of samples from the other sequences.
  In other words, computing bounds on $\tilde{y}^{(j)}(k)$ and $\tilde{u}^{(j)}(k)$ thanks to some linear program based on equations (3.52) and then use such information the POPs (3.46) and (3.47) (constraining $u^{(i)}, y^{(i)}\, i = 0, \ldots, n$ in their admissible interval) leads to much worse results, since in that case the correlation relating samples of different sequences would have been lost.

- The computational effort required to solve above optimization problems is quite high due to the number of variables involved

$$4n + 2(n+1)N$$

In fact this number is dominated by the product of the number of samples times the order of the system which in general is quite high in any identification procedure. This is a problem that and can quickly become dramatic, especially for system of order $n \geq 3$. In order to keep computational effort under control the only way is using a reduced number of samples but this brings to worse results.

3.5.1 Numerical Example

In this section we present a numerical experiment performed in order to verify the performances of the method described in the latter section.
Data are obtained from a simulation experiment on the plant $H(s)$ defined as

$$H(s) = \frac{15s + 22.4}{(s+10)(s+13)} = \frac{15s + 22.4}{s^2 + 23s + 130}$$

Then the obtained input and output signals are sampled with a sampling period of $T_s = 0.01\,\text{s}$ and corrupted by an additive bounded OE noise $|\eta(k)| \leq \Delta_\eta$ on each sample. The value of Δ_η was adjusted in order to obtain some desired signal-to-noise ratio.
Results are here reported in terms of:

- lower and upper bounds: $\underline{\theta_i^B}$ and $\overline{\theta_i^B}$.
- true value of the parameter $(\theta_i^B)_{\text{true}}$
- the central estimate, defined as

$$(\theta_i^B)^{\text{central}} = \frac{\overline{\theta_i^B} + \underline{\theta_i^B}}{2}$$

- relative uncertainty, evaluated as

$$\text{relative uncertainty}_{\%} = \frac{\underline{\theta_i^B} - \overline{\theta_i^B}}{(\theta_i^B)^{\text{central}}} \cdot 100$$

The simulation total time was set to 1.1 s, therefore the total number of samples turned out to be

$$N = \frac{T_{\text{sim}}}{T_s} = \frac{1.1\,\text{s}}{0.01\,\text{s}} = 110$$

for both input and output signals. Finally $\tau = 0.1\,\text{s}$ has been chosen as hyperparameter.
For the first solution, setting $\Delta_\eta = 0.22$ lead to a signal-to-noise ratio SNR $= 30\,\text{dB}$, and solving POPs (3.55) and (3.56), results in 3.1 are obtained.

While setting $\Delta_\eta = 0.7$, corresponding to SNR $= 20\,\text{dB}$, and solving POPs (3.55)

	$\underline{\theta_i}$	$(\theta_i)_{\text{true}}$	$\overline{\theta_i}$	$(\theta_i)^{\text{central}}$	relative uncertainty$_{\%}$
α_1	22.54	23	23.35	22.945	3.5294%
α_2	127.63	130	133.17	130.4	4.2496%
β_1	14.69	15	15.266	14.978	3.8456%
β_2	21.332	22.4	23.661	22.497	10.355%

Table 3.1: Results for SVF approach with SNR $= 30\,\text{dB}$ OE noise

and (3.56), results in 3.2 are obtained.

The amount of time required for the solution of the POPs is significant, in fact

	$\underline{\theta_i}$	$(\theta_i)_{\text{true}}$	$\overline{\theta_i}$	$(\theta_i)^{\text{central}}$	relative uncertainty$_{\%}$
α_1	21.756	23	23.879	22.818	9.305%
α_2	113.91	130	137.45	125.68	18.728%
α_1	14.126	15	15.665	14.896	10.332%
β_2	16.101	22.4	25.661	20.881	45.782%

Table 3.2: Results for SVF approach with SNR $= 20\,\text{dB}$ OE noise

there were necessary 1,467.96 s in the 20 dB case and 936.16 s in the 30 dB one. This is due to the fact that the number of variables involved in the optimizations is significantly high ($\approx 6N = 660$).

3.6 Tustin Discretization SM Approach

Similar results with a noticeable reduction in the computational effort with respect to what is achieved by means of the technique presented in 3.5, can be obtained exploiting some indirect procedure.
Among all the available techniques for discretizing a dynamic system, summarized in 3.3, each has his own advantages and disadvantages:

- Forward and Backward Euler methods can provide simple relationships between the parameters of the continuous-time transfer function $H(s)$ and their discrete-time counterpart's parameters, but the error they introduce is quite large

when compared to other methods for equal sampling periods.

- ZOH method is the best option when discretizing a model fed with piece-wise constant input signals. This is most of the times reasonable when dealing with plants of control systems but in general is not true. Moreover it does not provide a simple mapping from continuous-time and discrete-time transfer functions parameters.

- Matched method is able to preserve the frequency behaviour of the system but this is not what I need in order to develop my procedure which is not based on the frequency domain characteristics of the system. As for ZOH, also this method does not provide a simple mapping from continuous-time and discrete-time transfer functions parameters

- Tustin transform provides both simple relationships between the parameters of the continuous-time transfer function $H(s)$ and their discrete-time counterpart's parameters and a relatively low approximation when the involved signals $(u(t), y(t))$ are smooth enough.

On the basis of these observations, Tustin (or bilinear) approximation is definitely the most reasonable choice.
Up to now we stated that the relation between parameters of the transfer function $H(s|\alpha^c, \beta^c)$ we want to identify and parameters of his Tustin approximation $H(z|\alpha^d, \beta^d, T_s)$ is simple. But let's now enter into more details about this, deriving some explicit relation between α^d, β^d and α^c, β^c.
Consider the continuous time transfer function:

$$H(s|\alpha^c, \beta^c) = \frac{\sum_{i=0}^{n-1} \beta_i^c s^i}{s^n + \sum_{i=0}^{n-1} \alpha_i^c s^i} \tag{3.58}$$

Applying the Tustin transform

$$s = \frac{2}{T_s} \frac{z-1}{z+1} \tag{3.59}$$

we get

$$H(z|\alpha^c, \beta^c, T_s) = \frac{\sum_{i=0}^{n-1} \beta_i^c \left(\frac{2}{T_s} \frac{z-1}{z+1}\right)^i}{\left(\frac{2}{T_s} \frac{z-1}{z+1}\right)^n + \sum_{i=0}^{n-1} \alpha_i^c \left(\frac{2}{T_s} \frac{z-1}{z+1}\right)^i} \tag{3.60}$$

Then, multiplying both numerator and denominator by $T_s^n(z+1)^n$

$$H(z|\alpha^c,\beta^c,T_s)=\frac{\sum_{i=0}^{n-1}\beta_i^c 2^i(z-1)^i T_s^{n-i}(z+1)^{n-i}}{2^n(z-1)^n+\sum_{i=0}^{n-1}\alpha_i^c 2^i(z-1)^i T_s^{n-i}(z+1)^{n-i}} \tag{3.61}$$

At this point, it is possible to use black-box LTI discrete-time system identification in the SM framework in order to find bounds on parameters α^d,β^d of the discrete-time transfer function

$$H(z|\alpha^d,\beta^d)=\frac{\sum_{j=0}^{n}\beta_j^d z^j}{z^n+\sum_{j=0}^{n-1}\alpha_j^d z^j} \tag{3.62}$$

In general, the DT counterpart of the original CT transfer function obtained by means of Tustin discretization can be non strictly proper even if the original CT transfer function we want to identify is assumed to be strictly proper.
Then, by comparison of equations (3.61) and (3.62) we are able to determine the relationship between α^c,β^c and α^d,β^d. In order to do so we need to find the coefficients $\tilde{c}_j$ of the polynomial

$$\sum_{j=0}^{n}\tilde{b}_j z^j=\sum_{i=0}^{n-1}\gamma_i^c 2^i(z-1)^i T_s^{n-i}(z+1)^{n-i} \tag{3.63}$$

Exploiting again the convolution in order to find the coefficients of a polynomial expressed as product of monomials, we can define

$$c^{(i)}=\underbrace{\begin{bmatrix}1\\-1\end{bmatrix}*\cdots*\begin{bmatrix}1\\-1\end{bmatrix}}_{i\text{ times}}*\underbrace{\begin{bmatrix}1\\1\end{bmatrix}*\cdots*\begin{bmatrix}1\\1\end{bmatrix}}_{n-i\text{ times}} \tag{3.64}$$

which are vectors of $n+1$ elements containing the coefficients of the polynomial $(z-1)^i(z+1)^{n-i}$. In particular coefficient of z^j is at index $n+1-j$.
Thanks to this definition, (3.63) can be rewritten as

$$\sum_{j=0}^{n}\tilde{b}_j z^j=\sum_{i=0}^{n-1}\gamma_i^c 2^i T_s^{n-i}\sum_{j=0}^{n}c_{n+1-j}^{(i)}z^j \tag{3.65}$$

Then, thanks to the linearity of the summation operator,

$$\begin{aligned}\sum_{j=0}^{n}\tilde{b}_j z^j&=\sum_{i=0}^{n-1}\sum_{j=0}^{n}\gamma_i^c 2^i T_s^{n-i}c_{n+1-j}^{(i)}z^j\\&=\sum_{j=0}^{n}\sum_{i=0}^{n-1}\gamma_i^c 2^i T_s^{n-i}c_{n+1-j}^{(i)}z^j\end{aligned} \tag{3.66}$$

Finally, comparison of the right-hand side with the left-hand side of this equation allows to state that

$$\tilde{b}_j(\gamma^c) = \sum_{i=0}^{n-1} \gamma_i^c 2^i T_s^{n-i} c_{n+1-j}^{(i)} \tag{3.67}$$

Thanks to this result, we are immediately able to write (3.61) as

$$H(z|\alpha^c, \beta^c, T_s) = \frac{\sum_{j=0}^{n} \tilde{b}_j(\beta^c) z^j}{2^n (z-1)^n + \sum_{j=0}^{n} \tilde{b}_j(\alpha^c) z^j} \tag{3.68}$$

Then, given also that

$$2^n (z-1)^n = 2^n \sum_{j=0}^{n} c_{n+1-j}^{(n)} z^j \tag{3.69}$$

We can also define $\tilde{a}_j(\alpha^c)$ such that

$$\begin{aligned} & 2^n (z-1)^n + \sum_{j=0}^{n} \tilde{b}_j(\alpha^c) z^j = \\ & = 2^n \sum_{j=0}^{n} c_{n+1-j}^{(n)} z^j + \sum_{j=0}^{n} \tilde{b}_j(\alpha^c) z^j = \\ & = \sum_{j=0}^{n} \left(2^n c_{n+1-j}^{(n)} + \tilde{b}_j(\alpha^c) \right) z^j = \\ & = \sum_{j=0}^{n} \tilde{a}_j(\alpha^c) z^j \end{aligned} \tag{3.70}$$

Then,

$$\tilde{a}_j(\alpha^c) = 2^n c_{n+1-j}^{(n)} + \tilde{b}_j(\alpha^c) \tag{3.71}$$

At this point, given the definition of $\tilde{a}(\alpha^c)$ and $\tilde{b}(\beta^c)$ it holds

$$H(z|\alpha^c, \beta^c, T_s) = \frac{\sum_{j=0}^{n} \tilde{b}_j(\beta^c) z^j}{\sum_{j=0}^{n} \tilde{a}_j(\alpha^c) z^j} \tag{3.72}$$

This transfer function must be compared with (3.62) in which the denominator is normalized to have coefficient multiplying z^n equal to 1. Therefore the equations linking $\theta^d = [\alpha^d, \beta^d]$ with $\theta^c = [\alpha^c, \beta^c]$ are

$$\begin{aligned} \beta_j^d &= \frac{\tilde{b}_j(\beta^c)}{\tilde{a}_n(\alpha^c)} \qquad j = 0, \ldots, n \\ \alpha_j^d &= \frac{\tilde{a}_j(\alpha^c)}{\tilde{a}_n(\alpha^c)} \qquad j = 0, \ldots, n-1 \end{aligned} \tag{3.73}$$

which can be re-written as

$$\begin{aligned} \beta_j^d[\tilde{a}_n(\alpha^c)] &= \tilde{b}_j(\beta^c) \qquad j = 0, \dots, n \\ \alpha_j^d[\tilde{a}_n(\alpha^c)] &= \tilde{a}_j(\alpha^c) \qquad j = 0, \dots, n-1 \end{aligned} \tag{3.74}$$

Given the definitions of functions $\tilde{a}_j(\cdot)$ and $\tilde{b}_j(\cdot)$ in (3.71) and (3.67) we can notice that they are linear or affine with respect to the respective variable γ^c, therefore since in my final expressions (3.74) involves products between elements of $\theta^d = [\alpha^d, \beta^d]$ and given functions, the final expression is bilinear in the variable $\theta = [\theta^c, \theta^d]$.
Thanks to such result, we can finally motivate the following procedure allowing to solve the continuous-time black-box system identification problem in the SM approach. This consists into a two-steps procedure:

- Thanks to available samples of data

$$\{\tilde{y}(kT_s), \tilde{u}(kT_s)\} \doteq \{\tilde{y}(k), \tilde{u}(k)\} \qquad k = 1, \dots, N \tag{3.75}$$

identify bounds on the parameters of the discrete-time transfer function model (3.62)

$$H(z|\alpha^d, \beta^d) = \frac{\sum_{j=0}^{n} \beta_j^d z^{j-n}}{1 + \sum_{j=0}^{n-1} \alpha_j^d z^{j-n}} \tag{3.76}$$

by means of the SM approach summarized in 2.5.
In this particular case since the system is non strictly proper we need to solve POPs

$$\begin{aligned} &\underline{\theta_j^d} = \min_{\alpha^d, \beta^d, \eta, \epsilon} \theta_j^d \qquad \theta^d = [\alpha^d, \beta^d] \\ &\text{subject to} \\ &\tilde{y}(k) - \eta(k) + \sum_{i=0}^{n-1} \left[\alpha_i^d \tilde{y}(k+i-n) - \alpha_i^d \eta(k+i-n)\right] = \\ &\qquad = \sum_{i=0}^{n} \left[\beta_i^d \tilde{u}(k+i-n) - \beta_i^d \epsilon(k+i-n)\right] \qquad k = n+1, \dots, N \\ &\underline{\Delta_\eta}(k) \le \eta(k) \le \overline{\Delta_\eta}(k) \qquad k = 1, \dots, N \\ &\underline{\Delta_\epsilon}(k) \le \epsilon(k) \le \overline{\Delta_\epsilon}(k) \qquad k = 1, \dots, N \end{aligned} \tag{3.77}$$

and

$$\begin{aligned}
&\overline{\theta_j^d} = \max_{\alpha^d,\beta^d,\eta,\epsilon} \theta_j^d \qquad \theta^d = [\alpha^d, \beta^d] \\
&\quad \text{subject to} \\
&\quad \tilde{y}(k) - \eta(k) + \sum_{i=0}^{n-1} \left[\alpha_i^d \tilde{y}(k+i-n) - \alpha_i^d \eta(k+i-n)\right] = \\
&\qquad = \sum_{i=0}^{n} \left[\beta_i^d \tilde{u}(k+i-n) - \beta_i^d \epsilon(k+i-n)\right] \qquad k = n+1, \ldots, N \\
&\quad \underline{\Delta_\eta}(k) \le \eta(k) \le \overline{\Delta_\eta}(k) \qquad k = 1, \ldots, N \\
&\quad \underline{\Delta_\epsilon}(k) \le \epsilon(k) \le \overline{\Delta_\epsilon}(k) \qquad k = 1, \ldots, N
\end{aligned} \tag{3.78}$$

In the following we will denote obtained lower and upper bounds on α_j as $\underline{\alpha_j}$ and $\overline{\alpha_j}$ respectively. Similarly, we denote lower and upper bounds on β_j as $\underline{\beta_j}$ and $\overline{\beta_j}$.

- Second step of the procedure consists obviously into converting obtained bounds on the discrete-time transfer function into bounds on the bounds of the corresponding continuous-time transfer function parameters.
  As already shown, discrete-time parameters α^d, β^d and continuous-time ones α^c, β^c are related by means of bilinear transform by bilinear equalities (3.74), therefore it is possible to obtain bounds on continuous-time transfer function parameters by means of the solution of some additional polynomial optimization problems.

$$\begin{aligned}
&\underline{\theta_j^c} = \min_{\alpha^c,\beta^c,\alpha^d,\beta^d} \theta_j^c \qquad \theta^c = [\alpha^c, \beta^c] \\
&\quad \text{subject to} \\
&\quad \beta_i^d[\tilde{a}_n(\alpha^c)] = \tilde{b}_i(\beta^c) \qquad i = 0, \ldots, n \\
&\quad \alpha_i^d[\tilde{a}_n(\alpha^c)] = \tilde{a}_i(\alpha^c) \qquad i = 0, \ldots, n-1 \\
&\quad \underline{\alpha_i^d} \le \alpha_i^d \le \overline{\alpha_i^d} \qquad i = 0, \ldots, n \\
&\quad \underline{\beta_i^d} \le \beta_i^d \le \overline{\beta_i^d} \qquad i = 0, \ldots, n-1
\end{aligned} \tag{3.79}$$

and

$$\begin{aligned}
&\overline{\theta_j^c} = \max_{\alpha^c,\beta^c,\alpha^d,\beta^d} \theta_j^c \qquad \theta^c = [\alpha^c, \beta^c] \\
&\quad \text{subject to} \\
&\quad \beta_i^d[\tilde{a}_n(\alpha^c)] = \tilde{b}_i(\beta^c) \qquad i = 0, \ldots, n \\
&\quad \alpha_i^d[\tilde{a}_n(\alpha^c)] = \tilde{a}_i(\alpha^c) \qquad i = 0, \ldots, n-1 \\
&\quad \underline{\alpha_i^d} \le \alpha_i^d \le \overline{\alpha_i^d} \qquad i = 0, \ldots, n \\
&\quad \underline{\beta_i^d} \le \beta_i^d \le \overline{\beta_i^d} \qquad i = 0, \ldots, n-1
\end{aligned} \tag{3.80}$$

In which $\tilde{a}_k(\cdot)$ and $\tilde{b}_k(\cdot)$ are defined as in (3.71) and (3.67).

3.6.1 One-step approach

At this point, one can state that a new procedure for identifying continuous-time transfer functions in a SM framework has been established. Similarly with respect to what is done in 2.6, it is possible to notice that constraints

$$\begin{aligned} \underline{\alpha_i^d} \leq \alpha_i^d \leq \overline{\alpha_i^d} \qquad & i = 0, \dots, n \\ \underline{\beta_i^d} \leq \beta_i^d \leq \overline{\beta_i^d} \qquad & i = 0, \dots, n-1 \end{aligned} \tag{3.81}$$

are equivalent to the intersection of all constraints of the POP solved as first step of the solution:

$$\begin{aligned} &\tilde{y}(k) - \eta(k) + \sum_{i=0}^{n-1} \left[\alpha_i^d \tilde{y}(k+i-n) - \alpha_i^d \eta(k+i-n)\right] = \\ &\quad = \sum_{i=0}^{n} \left[\beta_i^d \tilde{u}(k+i-n) - \beta_i^d \epsilon(k+i-n)\right] \qquad k = n+1, \dots, N \\ &\underline{\Delta_\eta}(k) \leq \eta(k) \leq \overline{\Delta_\eta}(k) \qquad k = 1, \dots, N \\ &\underline{\Delta_\epsilon}(k) \leq \epsilon(k) \leq \overline{\Delta_\epsilon}(k) \qquad k = 1, \dots, N \end{aligned} \tag{3.82}$$

Therefore it could be a good idea to directly replacing constraints (3.81) on α^d and β^d in the second POP with constraints (3.82) involving the very same parameters α^d and β^d, the samples of the data and the ones of the noise.
Doing so, given samples of input and output signals $\{\tilde{y}(k), \tilde{u}(k)\}, \quad k = 1, \dots, N$ and bounds on the noise, upper and lower bounds on the parameters of $H(s|\alpha^c, \beta^c)$ are found solving POPs

$$\begin{aligned} &\underline{\theta_j^c} = \min_{\alpha^c, \beta^c, \alpha^d, \beta^d, \eta, \epsilon} \theta_j^c \qquad \theta^c = [\alpha^c, \beta^c] \\ &\quad \text{subject to} \\ &\quad \beta_i^d[\tilde{a}_n(\alpha^c)] = \tilde{b}_i(\beta^c) \qquad i = 0, \dots, n \\ &\quad \alpha_i^d[\tilde{a}_n(\alpha^c)] = \tilde{a}_i(\alpha^c) \qquad i = 0, \dots, n-1 \\ &\quad \tilde{y}(k) - \eta(k) + \sum_{i=0}^{n-1} \left[\alpha_i^d \tilde{y}(k+i-n) - \alpha_i^d \eta(k+i-n)\right] = \\ &\qquad = \sum_{i=0}^{n} \left[\beta_i^d \tilde{u}(k+i-n) - \beta_i^d \epsilon(k+i-n)\right] \qquad k = n+1, \dots, N \\ &\quad \underline{\Delta_\eta}(k) \leq \eta(k) \leq \overline{\Delta_\eta}(k) \qquad k = 1, \dots, N \\ &\quad \underline{\Delta_\epsilon}(k) \leq \epsilon(k) \leq \overline{\Delta_\epsilon}(k) \qquad k = 1, \dots, N \end{aligned} \tag{3.83}$$

and

$$
\begin{aligned}
&\overline{\theta_j^c} = \max_{\alpha^c,\beta^c,\alpha^d,\beta^d,\eta,\epsilon} \theta_j^c \qquad \theta^c = [\alpha^c, \beta^c] \\
&\text{subject to} \\
&\beta_i^d[\tilde{a}_n(\alpha^c)] = \tilde{b}_i(\beta^c) \qquad i = 0, \ldots, n \\
&\alpha_i^d[\tilde{a}_n(\alpha^c)] = \tilde{a}_i(\alpha^c) \qquad i = 0, \ldots, n-1 \\
&\tilde{y}(k) - \eta(k) + \sum_{i=0}^{n-1} \left[\alpha_i^d \tilde{y}(k+i-n) - \alpha_i^d \eta(k+i-n)\right] = \\
&\qquad = \sum_{i=0}^{n} \left[\beta_i^d \tilde{u}(k+i-n) - \beta_i^d \epsilon(k+i-n)\right] \qquad k = n+1, \ldots, N \\
&\underline{\Delta_\eta}(k) \leq \eta(k) \leq \overline{\Delta_\eta}(k) \qquad k = 1, \ldots, N \\
&\underline{\Delta_\epsilon}(k) \leq \epsilon(k) \leq \overline{\Delta_\epsilon}(k) \qquad k = 1, \ldots, N
\end{aligned}
\tag{3.84}
$$

When comparing this "1-step" approach with the "2-steps" presented at the beginning of this section I can make the following observations

- As discussed in 2.6 when comparing the 1-step and the two-steps approaches in gray-box SM problem in terms of conservativeness, the results obtained by means of a 1-step optimization instead of a multi-step procedure are improving. This happens because information from certain constraints which is described in terms of equations defining a semi-algebraic set is approximated by some hyper-rectangle when passing from one step to the following. Make reference to diagram 2.1 for a graphical interpretation.

- Both optimizations in the "2-steps" procedures are polynomial optimization problems involving equality constraints of degree at maximum 2, therefore it is possible to solve both with a low order of relaxation (i.e. starting from $\delta_{min} = 1$). When passing to the "1-step" approach the same order of relaxation can be used since the maximum degree is the same for all POPs. This is important since otherwise the computational effort would have increased.

- Computational effort of the two approaches is similar: with the 2-steps approach we need to solve $2n + 1$ POPs with $2N + 2n + 1$ variables plus $2n$ POPs with $4n + 1$ variables, while in the "1-step" approach we need to solve $2n$ POPs with $2N + 4n + 1$ variables.
  In fact, assuming that as usually happens $N \gg n$, the computational complexity of each optimization of the first POP of the 2-steps procedure is approximately the same as each optimization of the 1-step procedure.

3.6.2 Experiment Design and Convergence

In order to effectively identify the continuous-time model by means of (3.83) and (3.84), it is necessary to select some parameters and correctly choose the input signal to be applied, then collect data accordingly. In this section we discuss what are and how to select such parameters. A particular attention is devoted to convergence intended as the behaviour of the algorithm when the sampling time increases and we motivate why a finite one is sufficient to correctly handle the problem.

- the sampling time T_s, probably the most important parameter in this context. It must me chosen according to the criterion explained below in this section where I discuss the matter of convergence.

- the length of the experiment T_{exp}. It is of course related to the number of input and output samples collected by means of the relation

$$T_{exp} = N \cdot T_s \tag{3.85}$$

  The longer it lasts the experiment, the more are the available samples and therefore constraints in the POP. This leads to a reduction of the final uncertainty of the estimate in terms of width of the PUI. On the other hand, the computational effort required to solve the POPs grows.
  A good empirical staring point one can use can be evaluated by

$$T_{exp} \approx \frac{2\pi}{p_{min}} \tag{3.86}$$

  where p_{min} is some initial guess on the slower pole of the system to be identified.

The input signal to be fed into the system during the experiment is not so easy to be chosen as in the discrete-time case where it was sufficient to generate a white random sequence. In fact, in continuous-time we need to generate some continuous-time signal properly exciting the system. A reasonable input signal can be chosen to properly excite the system to be identified is

$$u(t) = \sum_{f=0}^{f_{max}} A_f \cos 2\pi f t \tag{3.87}$$

where amplitudes of each sinusoidal component A_f can be randomly generated numbers and the values of f linearly spaced between 0 and the maximum value f_{max} which is reasonably taken to be similar to the expected bandwidth of the system.
This class of input signals is differentiable infinite times (belongs to class C^∞) and

therefore the Tustin approximation is a good choice when approximating each derivative of this signal with the corresponding piece-wise linear approximate in the sampling interval. Moreover, if the input signal is chosen in this way, also the output signal is smooth enough so that the Tustin approximation holds with a reasonably small error.
Last, but probably more crucial topic, let's now discuss the matter of convergence. The question naturally arises when one makes an approximating assumption such as the one that the continuous-time transfer function is liked to the discrete-time one by means of Tustin discretization, is: *under which condition this assumption holds?*
In order to answer this question let consider the "2-step" approach to the solution and me analyze two different situations:

1. If no noise affects the data, optimizations (3.77) and (3.78) either turns out to be infeasible or the feasible set for α^d, β^d is represented by just one point (upper and lower bounds coincide).
   Then, when using (3.79) and (3.80) in order to map that point from the space of discrete-time parameters to the space of continuous-time parameters assuming Tustin transform, then two possible conditions may arise:

   - if the identified discrete-time system can be exactly described as the Tustin discretization of some continuous-time system, then also feasible set for α^c, β^c is just a point and the identification procedure is successful
   - if instead the identified discrete-time system cannot be described as the Tustin discretization of some strictly proper continuous-time system, the POPs (3.79) and (3.80) turn out out be infeasible.

   Notice that in this second step we are mapping $[\alpha^d, \beta^d] \in \mathbb{R}^{2n+1}$ into the lower dimensional space $[\alpha^c, \beta^c] \in \mathbb{R}^{2n}$, so at least in principle we have more equations then unknowns to be determined. This is due to the assumption that the continuous-time system is strictly proper and makes very likely that infeasibility occurs in this second step when no noise is present.
   In practice, all numerical experiments performed resulted in infeasibility of the POPs when $\Delta_\eta = 0$ and $\Delta_\epsilon = 0$ even if the collected data samples are noise-free.
   Moreover it is remarkable that repeating the procedure for decreasing sampling intervals T_s what happens is that on one side the max(constraint violation) gets smaller and smaller, and on the other side the returned value (even if not a solution, because the problem is always infeasible) is closer and closer to the true value.

2. If some noise affects the collected data optimizations (3.77) and (3.78) could in general be such that the feasible set for α^d, β^d is some non-convex subset of

$\mathbb{R}^{2n+1}$, this is the most realistic condition.
When such set is mapped to another feasible set for α^c, β^c using (3.79) and (3.80) then what could happen is

- there is no point inside the feasible set of α^d, β^d such that all equations (3.74) holds, i.e. none of the allowed discrete-time systems can be interpreted as a discretization of some strictly proper continuous-time system through Tustin transform. In this case the feasible set of problems (3.79) and (3.80) turns out to be infeasible. This is likely to happen when the sampling time is too large and as a consequence equations obtained from Tustin transformation are not respected due to the bad quality of the discretization.

- some points inside the feasible set of α^d, β^d satisfy all constraints and the problem is therefore feasible. In this case the obtained PUI on the continuous time parameters contains the true parameters. It is possible to interpret this situation as the case when the error due to the Tustin approximation is much smaller than the error due to the noise affecting the data.

A graphical two-dimensional representation can be visualized as follows

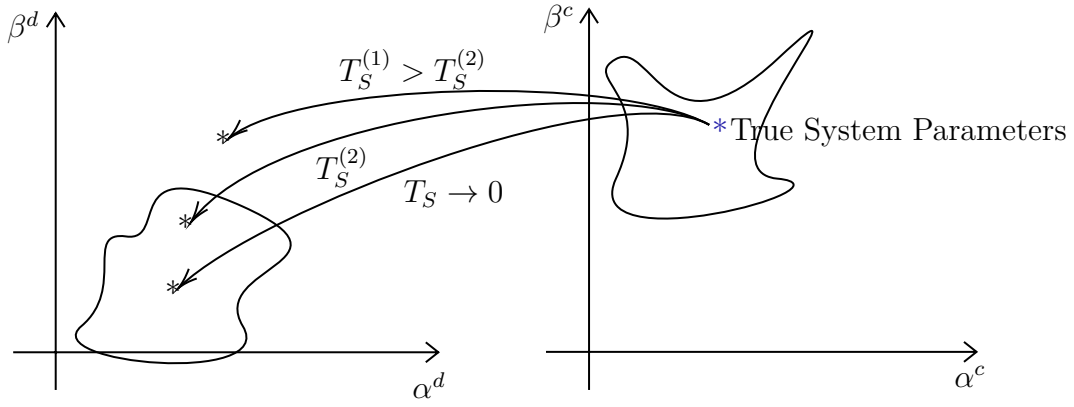

Figure 3.5: Mapping from CT to DT parameters for different T_s

All such observations suggests an empirical algorithm can be used to select the correct sampling time.

Algorithm 1: Sampling time selection

Result: PUI^c

Collect experimental data oversampling;
do
    decimate data in order to obtain some desired equivalent T_s;
    Solve POPs in order to obtain PUIs;
while *Obtained feasible solution*;

Oversampling during the experiment is suggested in order to avoid making multiple experiments on the physical plant, which may be time consuming. An initial guess on the starting equivalent sampling interval T_s can be chosen as $4\pi f_{max}$ according to the Nyquist–Shannon sampling theorem.
An interesting observation is that the minimum required sampling time is somehow related to the bounds $\Delta_\eta, \Delta_\epsilon$ which are assumed on the noise affecting the data. In particular the higher the noise bounds, the higher is the minimum required T_s which gives a feasible solution. This consideration makes sense as long as we give the interpretation that the error introduced by the Tustin approximation must be much smaller than the uncertainty due to the noise in order to get a non-empty feasible set.
Lastly, notice how all above considerations are true also in the case of the "1-step" procedure. This is true because above considerations are made on the feasible sets of the two POPs one after the other, but the feasible set of the second problem is just the intersection of the two previous ones.

3.6.3 Numerical Example

In this section some results from numerical experiments performed in order to verify the effectiveness of the proposed procedure are reported.
Considered experiments are performed on the very same system defined in 3.5.1, while results are obtained by means of the new procedure based on the solution of POPs (3.83) and (3.84). Also the other conditions are kept as in the previous numerical experiments: in particular the same data-set (same data, affected by the same noise) is used and performances are evaluated according to the same criterion. When SNR = 30 dB, we obtain results in table 3.3 in a total time of 84.64 s, while when SNR = 20 dB we obtain results in table 3.4 in a total time of 93.23 s.

It is possible to notice by comparison of tables 3.1 and 3.2 with 3.3 and 3.4 that the numerical results in terms of bounds values and uncertainty are very close each other. This can be explained thinking that the two approaches are using data affected by the same uncertainty.
The main difference between the two approaches is not in the numerical results but in the required computational effort: the approach based on state variables

	$\underline{\theta_i}$	$(\theta_i)_{\text{true}}$	$\overline{\theta_i}$	$(\theta_i)^{\text{central}}$	relative uncertainty$_{\%}$
α_1	22.565	23	23.35	22.958	3.4181%
α_2	127.63	130	132.99	130.31	4.1141%
β_1	14.714	15	15.266	14.99	3.6826%
β_2	21.332	22.4	23.604	22.468	10.113%

Table 3.3: Results for Tustin approach with SNR = 30 dB OE noise

	$\underline{\theta_i}$	$(\theta_i)_{\text{true}}$	$\overline{\theta_i}$	$(\theta_i)^{\text{central}}$	relative uncertainty$_{\%}$
α_1	21.954	23	23.879	22.917	8.4025%
α_2	113.91	130	136.2	125.06	17.819%
β_1	14.304	15	15.665	14.985	9.0817%
β_2	16.101	22.4	25.167	20.634	43.935%

Table 3.4: Results for Tustin approach with SNR = 20 dB OE noise

filters is doubtless much more computationally intensive due to the huge difference in the number of variables.
For this reason, it is possible to state that it is in all cases, it is preferable to approach the problem with the procedure based on Tustin discretization. For this reason from now on whenever a SM solution to the black-box CT identification problem is needed (e.g. as an intermediate step when solving the gray-box CT problem), we always consider this latter approach.

3.7 Gray-Box Problem

In this last section, we deal with the problem of identifying continuous-time gray-box model parameters. For sure this problem is much more motivated with respect to his discrete-time counterpart because the first principles of physics mostly allows to get continuous-time models, expressed in terms of differential equations.

3.7.1 Problem Statement

The CT SM gray-box identification problem can be formulated as follows:
Given the CT state-space model $\mathcal{M}_G$

$$\mathcal{M}_G : \begin{cases} \dot{x}(t) = A(\theta)x(t) + B(\theta)u(t) \\ y(t) = C(\theta)x(t) \end{cases} \tag{3.88}$$

where matrices $A(\theta), B(\theta)$ and $C(\theta)$ depends in an affine way by theta as

$$A(\theta) = A^{(0)} + \sum_{d=1}^{D} A^{(d)}\theta_d \qquad \text{similarly for } B(\theta), C(\theta) \tag{3.89}$$

and a set of samples of input and output signals

$$\{\tilde{y}(kT_s), \tilde{u}(kT_s)\} \doteq \{\tilde{y}(k), \tilde{u}(k)\} \qquad k = 1, \dots, N \tag{3.90}$$

corrupted by some unknown but bounded noise $\eta(k), \epsilon(k) \quad k = 1, \dots, N$ as in (3.29) according to the EIV noise structure (3.30) represented in figure 3.4, we need to find upper and lower bounds on each element of the parameters vector θ.

3.7.2 Solving the problem

In order to solve this problem, it is possible to use the same approach used in the discrete time case, which is again a multi-step procedure

1. From available data, find bounds on the parameters of the system's transfer function $H(s|\alpha^c, \beta^c)$. This requires the solution of POPs (3.83) and (3.84). The whole chapter so far was dedicated to the discussion about how to deal with this step.

2. Given bounds on the transfer function parameters $\underline{\alpha^c}, \overline{\alpha^c}, \underline{\beta^c}, \overline{\beta^c}$ it is now possible to build the corresponding state space black-box model in observability canonical form.

$$\mathcal{M}_B : \begin{cases} \dot{x}(t) = \mathcal{A}^{obs}x(t) + \mathcal{B}^{obs}u(t) \\ y(t) = \mathcal{C}^{obs}x(t) \end{cases} \tag{3.91}$$

   where

$$\begin{aligned} \mathcal{A}^{obs} = \mathcal{A}^{obs}(\alpha^c) &= \begin{bmatrix} 0 & 0 & \dots & 0 & -\alpha_0 \\ 1 & 0 & \dots & 0 & -\alpha_1 \\ 0 & 1 & \dots & 0 & -\alpha_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & -\alpha_{n_x-1} \end{bmatrix} \\ \mathcal{B}^{obs} = \mathcal{B}^{obs}(\beta^c) &= \begin{bmatrix} \beta_0 & \beta_1 & \dots & \beta_{n_x-2} & \beta_{n_x-1} \end{bmatrix}^\top \\ \mathcal{C}^{obs} &= \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \end{bmatrix} \end{aligned} \tag{3.92}$$

3. Given that black-box model (3.91) and the gray-box one (3.88) must be realization of the same system, then there must exist a matrix T such that

the similarity transformation equations

$$\begin{aligned}\mathcal{A}^{obs}T &= TA(\theta)\\ \mathcal{B}^{obs} &= TB(\theta)\\ \mathcal{C}^{obs}T &= C(\theta)\end{aligned}$$

Moreover, under the additional information that each matrix of the gray-box model depends in an affine way from the gray-box parameters, above matrix products can be explicitly re-written as already did in 2.5.2, obtaining:

$$\begin{aligned}
-\alpha_0^c t_{n_x,j} &= \sum_{k=1}^{n_x} t_{1,k}\, a_{k,j}^{(0)} + \sum_{k=1}^{n_x}\sum_{d=1}^{D} a_{k,j}^{(d)}\, t_{1,k}\,\theta_d \\
&\quad j = 1,\ldots,n_x \\
t_{i-1,j} - \alpha_{i-1}^c t_{n_x,j} &= \sum_{k=1}^{n_x} t_{i,k}\, a_{k,j}^{(0)} + \sum_{k=1}^{n_x}\sum_{d=1}^{D} a_{k,j}^{(d)}\, t_{i,k}\,\theta_d \\
&\quad i = 2,\ldots,n_x,\ j = 1,\ldots,n_x \\
\beta_{i-1}^c &= \sum_{k=1}^{n_x} b_k^{(0)}\, t_{i,k} + \sum_{k=1}^{n_x}\sum_{d=1}^{D} b_k^{(d)}\, t_{i,k}\,\theta_d \qquad i = 1,\ldots,n_x \\
t_{n_x,i} &= c_i^{(0)} + \sum_{d=1}^{D} c_i^{(d)}\theta_d \qquad i = 1,\ldots,n_x
\end{aligned} \tag{3.93}$$

So finally the problem can be solved by means of the solution of POPs

$$\begin{aligned}
\underline{\theta}_j \quad &= \min_{\theta,\alpha^c,\beta^c,T} \theta_j \qquad j = 1,\ldots,D \\
&\text{subject to} \\
&\mathcal{A}^{obs}(\alpha^c)T = TA(\theta) \\
&\mathcal{B}^{obs}(\beta^c) = TB(\theta) \\
&\begin{bmatrix}0 & \ldots & 0 & 1\end{bmatrix} T = C(\theta) \\
&\underline{\alpha_i^c} \le \alpha_i^c \le \overline{\alpha_i^c} \qquad i = 1,\ldots,n \\
&\underline{\beta_i^c} \le \beta_i^c \le \overline{\beta_i^c} \qquad i = 1,\ldots,n
\end{aligned} \tag{3.94}$$

and

$$\begin{aligned}
\overline{\theta}_j \quad &= \max_{\theta,\alpha^c,\beta^c,T} \theta_j \qquad j = 1,\dots,D \\
&\text{subject to} \\
&\mathcal{A}^{obs}(\alpha^c)T = TA(\theta) \\
&\mathcal{B}^{obs}(\beta^c) = TB(\theta) \\
&\begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} T = C(\theta) \\
&\underline{\alpha_i^c} \le \alpha_i^c \le \overline{\alpha_i^c} \qquad i = 1,\dots,n \\
&\underline{\beta_i^c} \le \beta_i^c \le \overline{\beta_i^c} \qquad i = 1,\dots,n
\end{aligned} \tag{3.95}$$

Which are very similar to POPs (2.29) and (2.30), with the difference that variables α, β are now substituted by α^c, β^c.

3.7.3 One-step approach

Continuing along the lines of what we did for the continuous time problem, also in this case it possible to reformulate the problem as a one-step procedure allowing to obtain considerable advantages.
In this case we notice that constraints

$$\begin{aligned}
&\underline{\alpha_i^c} \le \alpha_i^c \le \overline{\alpha_i^c} \qquad i = 1,\dots,n \\
&\underline{\beta_i^c} \le \beta_i^c \le \overline{\beta_i^c} \qquad i = 1,\dots,n
\end{aligned}$$

can be replaced by the feasible set of POPs (3.83) and (3.84) from which they are evaluated

$$\begin{aligned}
&\beta_i^d[\tilde{a}_n(\alpha^c)] = \tilde{b}_i(\beta^c) \qquad i = 0,\dots,n \\
&\alpha_i^d[\tilde{a}_n(\alpha^c)] = \tilde{a}_i(\alpha^c) \qquad i = 0,\dots,n-1 \\
&\tilde{y}(k) - \eta(k) + \sum_{i=1}^{n} \left[\alpha_{n-i}^d \tilde{y}(k-i) - \alpha_{n-i}^d \eta(k-i)\right] = \\
&\qquad = \sum_{i=0}^{n} \left[\beta_{n-i}^d \tilde{u}(k-i) - \beta_{n-i}^d \epsilon(k-i)\right] \qquad k = n+1,\dots,N \\
&\underline{\Delta_\eta}(k) \le \eta(k) \le \overline{\Delta_\eta}(k) \qquad k = 1,\dots,N \\
&\underline{\Delta_\epsilon}(k) \le \epsilon(k) \le \overline{\Delta_\epsilon}(k) \qquad k = 1,\dots,N
\end{aligned}$$

Therefore the gray-box system identification problem in the SM framework can be solved by the solution of the following POPs

$$
\begin{aligned}
\underline{\theta}_j \quad &= \min_{\theta,\alpha^c,\beta^c,T,\eta,\epsilon,\alpha^d,\beta^d} \theta_j \qquad j = 1,\dots,D \\
&\text{subject to} \\
&\mathcal{A}^{obs}(\alpha^c)T = TA(\theta) \\
&\mathcal{B}^{obs}(\beta^c) = TB(\theta) \\
&\begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} T = C(\theta) \\
&\beta_i^d[\tilde{a}_n(\alpha^c)] = \tilde{b}_i(\beta^c) \qquad i = 0,\dots,n \\
&\alpha_i^d[\tilde{a}_n(\alpha^c)] = \tilde{a}_i(\alpha^c) \qquad i = 0,\dots,n-1 \\
&\tilde{y}(k) - \eta(k) + \sum_{i=1}^{n} \left[\alpha_{n-i}^d \tilde{y}(k-i) - \alpha_{n-i}^d \eta(k-i)\right] = \\
&\qquad = \sum_{i=0}^{n} \left[\beta_{n-i}^d \tilde{u}(k-i) - \beta_{n-i}^d \epsilon(k-i)\right] \qquad k = n+1,\dots,N \\
&\underline{\Delta_\eta}(k) \le \eta(k) \le \overline{\Delta_\eta}(k) \qquad k = 1,\dots,N \\
&\underline{\Delta_\epsilon}(k) \le \epsilon(k) \le \overline{\Delta_\epsilon}(k) \qquad k = 1,\dots,N
\end{aligned}
\tag{3.96}
$$

and

$$
\begin{aligned}
\overline{\theta}_j \quad &= \max_{\theta,\alpha^c,\beta^c,T,\eta,\epsilon,\alpha^d,\beta^d} \theta_j \qquad j = 1,\dots,D \\
&\text{subject to} \\
&\mathcal{A}^{obs}(\alpha^c)T = TA(\theta) \\
&\mathcal{B}^{obs}(\beta^c) = TB(\theta) \\
&\begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} T = C(\theta) \\
&\beta_i^d[\tilde{a}_n(\alpha^c)] = \tilde{b}_i(\beta^c) \qquad i = 0,\dots,n \\
&\alpha_i^d[\tilde{a}_n(\alpha^c)] = \tilde{a}_i(\alpha^c) \qquad i = 0,\dots,n-1 \\
&\tilde{y}(k) - \eta(k) + \sum_{i=1}^{n} \left[\alpha_{n-i}^d \tilde{y}(k-i) - \alpha_{n-i}^d \eta(k-i)\right] = \\
&\qquad = \sum_{i=0}^{n} \left[\beta_{n-i}^d \tilde{u}(k-i) - \beta_{n-i}^d \epsilon(k-i)\right] \qquad k = n+1,\dots,N \\
&\underline{\Delta_\eta}(k) \le \eta(k) \le \overline{\Delta_\eta}(k) \qquad k = 1,\dots,N \\
&\underline{\Delta_\epsilon}(k) \le \epsilon(k) \le \overline{\Delta_\epsilon}(k) \qquad k = 1,\dots,N
\end{aligned}
\tag{3.97}
$$

This approach is more convenient with respect to the multi-step one for at least three reasons. The first two reasons are the same as the ones presented in Section 2.6,

namely improved precision and reduced computational effort. Also the motivations for which they holds are the same.
The good new is that also another advantage arises here. When the sampling frequency is not big enough, the problems are even more likely to be infeasible because now additional constraints giving information about the structure of the system are present. For this reason any parameter value for which the resulting system does not have those characteristics is unfeasible.

3.7.4 Numerical Example

Consider the following CT gray-box model

$$\mathcal{M}_G = \begin{cases} x(k+1) = \begin{bmatrix} \theta_1^G & -0.9 \\ 12 & -2\theta_2^G \end{bmatrix} x(k) + \begin{bmatrix} 3\theta_2^G \\ 1 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(k) \end{cases}$$

Data are generated my means of a simulation experiment last for $T_{exp} = 4\,\mathrm{s}$ with true values of the unknown parameters

$$\theta = \begin{bmatrix} -1.3 \\ 2 \end{bmatrix}$$

Input signal is generated as the summation of sine waves with random amplitudes in the range of frequencies between $0\,\mathrm{rad\,s^{-1}}$ and $8\,\mathrm{rad\,s^{-1}}$.
Obtained data has been sampled with sampling time T_s and obtained samples corrupted by an additive OE noise, bounded by $\Delta_\eta = 0.7$ and $\Delta_\eta = 2$ in the case when $\mathrm{SNR} = 30\,\mathrm{dB}$ and $\mathrm{SNR} = 20\,\mathrm{dB}$ respectively.

- With sampling time $T_s = 0.2\,\mathrm{s}$, both the two-steps approach ((3.83) and (3.84) followed by (3.94) and (3.95)) and the one-step approach ((3.96) and (3.97)) are characterized by infeasibility of all optimization problems required.
- Reducing the sampling time down to $T_s = 0.05\,\mathrm{s}$, feasibility of the problem is now reached and we are able to obtain results in 3.5 and 3.6 when $\mathrm{SNR} = 30\,\mathrm{dB}$ using respectively the two-steps and the one-step approaches. The 2-steps case required a computational time of $42.05\,\mathrm{s} + 1.15\,\mathrm{s}$, while the one-step required only $18.41\,\mathrm{s}$. Similarly when $\mathrm{SNR} = 20\,\mathrm{dB}$, we obtain 3.7 and 3.8 in $52.18\,\mathrm{s} + 1.10\,\mathrm{s}$ and $26.01\,\mathrm{s}$ respectively.

As final remarks one can notice that

- When the sampling time is too small, infeasibility occurs. This is true as long as the number of samples used is maintained reasonably high, otherwise it

	$\underline{\theta_i^B}$	$(\theta_i^B)_{\text{true}}$	$\overline{\theta_i^B}$	$(\theta_i^B)^{\text{central}}$	relative uncertainty$_{\%}$
α_1	5.217543	5.300000	5.378295	5.297919	3.034252%
α_2	15.455244	16.000000	16.792887	16.124065	8.295946%
β_1	5.883344	6.000000	6.242111	6.062727	5.917581%
β_2	22.228817	23.100000	24.247296	23.238057	8.686091%
	$\underline{\theta_i^G}$	$(\theta_i^G)_{\text{true}}$	$\overline{\theta_i^G}$	$(\theta_i^G)^{\text{central}}$	relative uncertainty$_{\%}$
θ_1^G	−1.451565	−1.300000	−1.136953	−1.294259	24.3083%
θ_2^G	1.963365	2.000000	2.047246	2.005306	4.1829%

Table 3.5: Results for 2-steps solution with SNR = 30 dB OE noise

	$\underline{\theta_i^G}$	$(\theta_i^G)_{\text{true}}$	$\overline{\theta_i^G}$	$(\theta_i^G)^{\text{central}}$	relative uncertainty$_{\%}$
θ_1^G	−1.319069	−1.300000	−1.275358	−1.297213	3.369633%
θ_2^G	1.987902	2.000000	2.003932	1.995917	0.803128%

Table 3.6: Results for 1-step solution with SNR = 30 dB OE noise

could happen that some incorrect solution is found in a neighborhood of the true parameter. For this reason it is suggested to always use at least $N = 60$ samples.
If the maximum available sampling time is not large (unrealistic case nowadays), the experiment should last for a time sufficient to collect "many" samples, so that if the sampling time is too small the user can at least realize that.

- Increasing the sampling frequency causes an increase in the number of samples when the experiment time is fixed, and therefore an increase of the computational effort. In any case numerical experiments shows that computational effort never grows too large even when sampling time is taken, by purpose, very small with respect to the system's time constants.

- As in the discrete-time case, the 1-step approach is more convenient both in terms of speed and accuracy with respect to the 2-steps one.

	$\underline{\theta_i^B}$	$(\theta_i^B)_{\text{true}}$	$\overline{\theta_i^B}$	$(\theta_i^B)^{\text{central}}$	relative uncertainty$_{\%}$
α_1	5.063766	5.300000	5.674412	5.369089	11.373364%
α_2	13.938304	16.000000	17.126753	15.532528	20.527561%
β_1	5.735642	6.000000	6.578771	6.157206	13.693367%
β_2	20.092991	23.100000	25.172918	22.632955	22.444824%
	$\underline{\theta_i^G}$	$(\theta_i^G)_{\text{true}}$	$\overline{\theta_i^G}$	$(\theta_i^G)^{\text{central}}$	relative uncertainty$_{\%}$
θ_1^G	−1.654589	−1.300000	−0.894600	−1.274594	59.625954%
θ_2^G	1.911881	2.000000	2.084583	1.998232	8.642762%

Table 3.7: Results for 2-steps solution with SNR = 20 dB OE noise

	$\underline{\theta_i^G}$	$(\theta_i^G)_{\text{true}}$	$\overline{\theta_i^G}$	$(\theta_i^G)^{\text{central}}$	relative uncertainty$_{\%}$
θ_1^G	−1.430486	−1.300000	−1.207689	−1.319087	16.890267%
θ_2^G	1.953861	2.000000	2.058026	2.005944	5.192816%

Table 3.8: Results for 1-step solution with SNR = 20 dB OE noise

Chapter 4

Problem's Generalizations

In this chapter, some generalizations of the problem is considered. Such generalizations allows to relax some of the constraints under which above methods were developed.
Methods below can be also applied both in the discrete-time and continuous-time without particular modifications. It is sufficient to add variables α^c, β^c and constraints (3.74) to the optimization problem to solve the continuous-time case, while directly using discrete-time transfer function parameters in observability matrices poses the problem in discrete-time.

4.1 Polynomially structured problems

The less realistic assumption we did in all previous derivations was "matrices $A(\theta), B(\theta), C(\theta)$ are affine in θ".
This assumption can be easily relaxed to the more general case when the dependency of mentioned matrices with respect to θ is a sum of ratio of polynomials

$$[A]_{ij} = \sum_t \frac{n_t(\theta)}{d_t(\theta)} \qquad n_t, d_t \in \mathbb{R}[\theta],\ \forall t \tag{4.1}$$

and similarly for elements of B and C.
In such situation, it is possible to keep the previous solution and solve the new problem just by introducing additional variables and constraints. In fact, it is possible to introduce n_f additional fake variables $\theta_{D+1}, \ldots, \theta_{D+n_f}$ representing each ratio of polynomials and impose constraints enforcing this fact as

$$\theta_{D+t} = \frac{n_t(\theta)}{d_t(\theta)} \implies \theta_{D+t} d_t(\theta) = n_t(\theta) \qquad t = 1, \ldots, n_f \tag{4.2}$$

This of course requires that $d(\theta) \neq 0$, but generally this is evident from the physical meaning of the equations formulated when building the gray-box model.

Additional constraints (4.2) are by construction polynomial equality constraints of degree $\max\{\deg n(x), 1 + \deg d(x)\}$. For this reason they can be safely introduced in the polynomial optimizations (3.96) and (3.97) as long as the order of relaxation used to solve them by means of SDP relaxation is augmented at least above

$$\delta^{\min} = \left\lceil \frac{\max\{\deg n(x), 1 + \deg d(x)\}}{2} \right\rceil \tag{4.3}$$

This technique is exemplified in Section 4.4, where a real-word example is presented.

4.2 MISO systems

A further extension of this theory can be performed when dealing with Multiple Input Single Output (MISO) systems.
In this case, the gray-box description of the system (3.88) involves matrices $A(\theta) \in \mathbb{R}^{n_x,n_x}$, $C(\theta) \in \mathbb{R}^{n_x}$ as in the SISO case, while matrix $B(\theta)$ now has dimensions depending on both the order of the system and number of inputs n_u:

$$B(\theta) \in \mathbb{R}^{n_x,n_u}$$

When solving the SISO problem, the main intermediate step was to find bounds on the transfer function parameters. In this case, the transfer function of the system is now a vector of transfer functions

$$H(s) = \begin{bmatrix} H_1(s) & \dots & H_{n_u}(s) \end{bmatrix} \tag{4.4}$$

where

$$H_j(s) = \frac{\beta^{c,(j)}_{n_x-1} s^{n_x-1} + \dots + \beta^{c,(j)}_0}{s^{n_x} + \alpha^c_{n_x-1} s^{n_x-1} + \dots + \alpha^c_0} \qquad j = 1, \dots, n_u$$

It is possible to notice that denominator coefficients are assumed here to be all the same for each transfer function in the vector. This is reasonable since in the statement of the problem we are given one unique matrix A for the whole system which is also assumed to be minimal.
The output is obtained from the combination of the inputs through them as

$$Y(s) = H(s)U(s) = \sum_{j=1}^{n_u} H_j(s)U_j(s) \tag{4.5}$$

The idea in this case is to estimate parameters $\theta^c = [\alpha^c, \beta^{c,(1)}, \dots, \beta^{c,(n_u)}]$ by means of

1. Set-membership estimation of all parameters of discrete-time transfer functions $H_j(z)$ relating input and output samples. This can be performed by means of polynomial optimization as detailed in [31]. Here we just report optimization problems to be solved in order to find interesting bounds

$$
\begin{aligned}
&\underline{\theta_j^B} = \min_{\alpha^d,\beta^d,\eta,\epsilon,z} \theta_j^B \qquad \theta^B = [\alpha^d, \beta^{d,(1)}, \ldots, \beta^{d,(n_u)}] \\
&\quad \text{subject to} \\
&\quad z_l(k) + \sum_{i=1}^{n_x} \alpha_i^d z_l(k+i-n_x) = \sum_{i=1}^{n_x} \beta_i^{d,(l)} u_l(k+i-n_x) - \beta_i^{d,(l)} \epsilon_l(k+i-n_x) \\
&\qquad l = 1, \ldots, n_u, \quad k = n_x+1, \ldots, N \\
&\quad y(k) = z_1(k) + \cdots + z_{n_u}(k) + \eta(k) \qquad k = 1, \ldots, N \\
&\quad \underline{\Delta_\eta} \leq \eta(k) \leq \overline{\Delta_\eta} \qquad k = 1, \ldots, N \\
&\quad \underline{\Delta_\epsilon}^{(l)} \leq \epsilon_l(k) \leq \overline{\Delta_\epsilon}^{(l)} \qquad l = 1, \ldots, n_u, \quad k = 1, \ldots, N
\end{aligned}
\tag{4.6}
$$

and

$$
\begin{aligned}
&\overline{\theta_j^B} = \max_{\alpha^d,\beta^d,\eta,\epsilon,z} \theta_j^B \qquad \theta^B = [\alpha^d, \beta^{d,(1)}, \ldots, \beta^{d,(n_u)}] \\
&\quad \text{subject to} \\
&\quad z_l(k) + \sum_{i=1}^{n_x} \alpha_i^d z_l(k+i-n_x) = \sum_{i=1}^{n_x} \beta_i^{d,(l)} u_l(k+i-n_x) - \beta_i^{d,(l)} \epsilon_l(k+i-n_x) \\
&\qquad l = 1, \ldots, n_u, \quad k = n_x+1, \ldots, N \\
&\quad y(k) = z_1(k) + \cdots + z_{n_u}(k) + \eta(k) \qquad k = 1, \ldots, N \\
&\quad \underline{\Delta_\eta} \leq \eta(k) \leq \overline{\Delta_\eta} \qquad k = 1, \ldots, N \\
&\quad \underline{\Delta_\epsilon}^{(l)} \leq \epsilon_l(k) \leq \overline{\Delta_\epsilon}^{(l)} \qquad l = 1, \ldots, n_u, \quad k = 1, \ldots, N
\end{aligned}
\tag{4.7}
$$

2. Relationship between continuous-time and discrete-time transfer functions parameters obtained thanks to the assumption of Tustin discretization

$$\alpha_j^d \tilde{a}_n(\alpha^c) = \tilde{a}_j(\alpha^c) \quad j = 0, \ldots, n_x - 1 \tag{4.8}$$

$$\beta_j^{d,(l)} \tilde{a}_n(\alpha^c) = \tilde{b}_j(\beta^{c,(l)}) \quad j = 0, \ldots, n_x,\ l = 1, \ldots, n_u \tag{4.9}$$

At this point, observability canonical form of the MISO system can be built as

$$\begin{aligned}
\mathcal{A}^{obs} &= \begin{bmatrix} 0 & 0 & \dots & 0 & -\alpha_0^c \\ 1 & 0 & \dots & 0 & -\alpha_1^c \\ \vdots & \dots & \dots & \dots & \vdots \\ 0 & \dots & 0 & 1 & -\alpha_{n_x-1}^c \end{bmatrix} \\
\mathcal{B}^{obs} &= \begin{bmatrix} \beta_0^{c,(1)} & \beta_0^{c,(2)} & \dots & \beta_0^{c,(n_u)} \\ \vdots & \dots & \dots & \vdots \\ \beta_{n_x-1}^{c,(1)} & \beta_{n_x-1}^{c,(2)} & \dots & \beta_{n_x-1}^{c,(n_u)} \end{bmatrix} \\
\mathcal{C}^{obs} &= \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \end{bmatrix}
\end{aligned} \tag{4.10}$$

And expanding row-by-column matrix products of equations (2.5) we can state that on one hand $\mathcal{A}^{obs}$ and $\mathcal{C}^{obs}$ are unchanged with respect to the SISO case, so we still get (3.93). On the other hand $\mathcal{B}^{obs} = \mathcal{B}^{obs}(\beta^c)$ is now a matrix in $\mathbb{R}^{n_x,n_u}$. Therefore expanding the equation $\mathcal{B}^{obs}(\beta^c) = TB(\theta)$ gives

$$\begin{aligned}
\beta_{i-1}^{c,(j)} = \mathcal{B}_{i,j}^{obs} &= \sum_{k=1}^{n_x} t_{i,k} b_{k,j}(\theta) \\
&= \sum_{k=1}^{n_x} t_{i,k} \left(b_{k,j}^{(0)} + \sum_{d=1}^{D} b_{k,j}^{(d)} \theta_d \right) \\
&= \sum_{k=1}^{n_x} b_{k,j}^{(0)} t_{i,k} + \sum_{k=1}^{n_x} \sum_{d=1}^{D} b_{k,j}^{(d)} t_{i,k} \theta_d
\end{aligned} \tag{4.11}$$

Substitution of (4.11) in place of the third equation of (3.93) when converting bounds on the black-box transfer function parameters into bounds on the gray-box ones, allows to generalize the problem to the MISO case.
Overall, the one-step solution which allows to find bounds on gray-box parameters

in the MISO case is therefore

$$
\begin{aligned}
&\underline{\theta_j} = \min_{\theta,T,\alpha,\beta,\eta,\epsilon,z} \theta_j \\
&\quad \text{subject to} \\
&\quad \mathcal{A}^{obs}(\alpha^c)T = TA(\theta) \\
&\quad \mathcal{B}^{obs}(\beta^{c,(1)},\dots,\beta^{c,(n_u)}) = TB(\theta) \\
&\quad \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} T = C(\theta) \\
&\quad \alpha_j^d \tilde{a}_n(\alpha^c) = \tilde{a}_j(\alpha^c) \quad j = 0,\dots,n_x-1 \\
&\quad \beta_j^{d,(l)} \tilde{a}_n(\alpha^c) = \tilde{b}_j(\beta^{c,(l)}) \qquad j = 0,\dots,n_x,\ l = 1,\dots,n_u \\
&\quad z_l(k) + \sum_{i=1}^{n_x} \alpha_i^d z_l(k+i-n_x) = \sum_{i=1}^{n_x} \beta_i^{d,(l)} u_l(k+i-n_x) - \beta_i^{d,(l)} \epsilon_l(k+i-n_x) \\
&\qquad l = 1,\dots,n_u, \quad k = n_x+1,\dots,N \\
&\quad y(k) = z_1(k) + \dots + z_{n_u}(k) + \eta(k) \qquad k = 1,\dots,N \\
&\quad \underline{\Delta_\eta} \le \eta(k) \le \overline{\Delta_\eta} \qquad k = 1,\dots,N \\
&\quad \underline{\Delta_\epsilon}^{(l)} \le \epsilon_l(k) \le \overline{\Delta_\epsilon}^{(l)} \qquad l = 1,\dots,n_u, \quad k = 1,\dots,N
\end{aligned}
\tag{4.12}
$$

And similarly for the upper bounds $\overline{\theta_j}$, we just need to replace min with max in the previous optimization. Such problems are, as in the SISO case, polynomial optimization problems with bilinear constraints, therefore can be solved by means of SDP relaxation starting from order of relaxation $\delta^{\min} = 1$. What is different is the number of optimization variables, that increases to

$$D + n_x^2 + (1+n_u)n_x + (1+2n_u)N$$

and therefore the computational effort increases.

4.3 MIMO systems

Extension to Multiple-Input Multiple-Output (MIMO) systems is quite trivial as long as one is able to exploit the previous result for MISO systems.
In fact, a MIMO system can be seen as a set of n_y MISO systems, where n_y is the number of outputs. For this reason, what can be done is solving optimization problems where the feasible set can be described as

$$\mathcal{D}_\theta = \bigcap_{m=1}^{n_y} \mathcal{D}_\theta^{(m)} \tag{4.13}$$

where $\mathcal{D}_\theta^{(m)}$ is the feasible set described by constraints of POP (4.12) for output m. Entering into more details, we can state that in this MIMO case, the transfer

function of the system is a matrix, where each row is linked to an output.

$$H(s) = \begin{bmatrix} H_{1,1}(s) & \dots & H_{1,n_u}(s) \\ \vdots & \ddots & \vdots \\ H_{n_y,1}(s) & \dots & H_{n_y,n_u}(s) \end{bmatrix} \tag{4.14}$$

Denominators of all elements of the matrix transfer function can be taken as equal for the same reasons as in the MISO case.

$$H_{m,l}(s) = \frac{\beta_{n_x-1}^{c,(m,l)} s^{n_x-1} + \dots + \beta_0^{c,(m,l)}}{s^{n_x} + \alpha_{n_x-1} z^{n_x-1} + \dots + \alpha_0} \qquad m = 1, \dots, n_y,\ l = 1, \dots, n_u \tag{4.15}$$

Estimating bounds on parameters of the all such transfer functions parameters relating input and output samples in the discrete-time domain can be again formulated as a POP [31]. Then, as in the SISO and MISO case, such parameters can be linked to the corresponding continuous-time ones by means of the Tustin transform (3.71) and (3.67). Optimization problems to be solved to find such bounds are

$$\begin{aligned}
&\underline{\theta_j^B} = \min_{\alpha^c, \beta^c, \alpha^d, \beta^d, \eta, \epsilon, z} \theta_j^B \qquad \theta^B = [\alpha^c, \beta^{c,(\cdot,\cdot)}] \\
&\quad \text{subject to} \\
&\quad z_{m,l}(k) + \sum_{i=0}^{n_x-1} \alpha_i^d z_{m,l}(k+i-n) = \sum_{i=0}^{n_x} \beta_i^{d,(m,l)} u_l(k+i-n) - \beta_i^{d,(m,l)} \epsilon_l(k+i-n) \\
&\qquad m = 1, \dots, n_y, \quad l = 1, \dots, n_u, \quad k = n_x+1, \dots, N \\
&\quad y_m(k) = z_{m,1}(k) + \dots + z_{m,n_u}(k) + \eta_m(k) \\
&\qquad m = 1, \dots, n_y, \quad k = 1, \dots, N \\
&\quad \alpha_j^d \tilde{a}_n(\alpha^c) = \tilde{a}_j(\alpha^c) \quad j = 0, \dots, n_x - 1 \\
&\quad \beta_j^{d,(m,l)} \tilde{a}_n(\alpha^c) = \tilde{b}_j(\beta^{c,(m,l)}) \\
&\qquad m = 1, \dots, n_y\, j = 0, \dots, n_x,\ l = 1, \dots, n_u \\
&\quad \underline{\Delta_\eta}^{(m)} \le \eta_m(k) \le \overline{\Delta_\eta}^{(m)} \qquad m = 1, \dots, n_y, \quad k = 1, \dots, N \\
&\quad \underline{\Delta_\epsilon}^{(l)} \le \epsilon_l(k) \le \overline{\Delta_\epsilon}^{(l)} \qquad l = 1, \dots, n_u, \quad k = 1, \dots, N
\end{aligned} \tag{4.16}$$

and similarly for the upper bounds. Notice that this POPs allows to solve the black-box continuous time SM identification problem for MIMO systems.
The final objective is then to estimate parameters of a MIMO gray-box model. Building observability canonical form matrices (the MISO fashion) for each output

and adding the related constraints to the above POPs, we finally obtain

$$
\begin{aligned}
&\underline{\theta_j} = \min_{\theta,T,\alpha,\beta,\eta,\epsilon,z} \theta_j \\
&\quad \text{subject to} \\
&\quad \mathcal{A}^{obs}(\alpha^c)T^{(m)} = T^{(m)}A(\theta) \qquad m = 1,\dots,n_y \\
&\quad \mathcal{B}^{obs}(\beta^{c,(m,1)},\dots,\beta^{c,(m,n_u)}) = T^{(m)}B(\theta) \qquad m = 1,\dots,n_y \\
&\quad \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} T^{(m)} = C_m(\theta) \qquad m = 1,\dots,n_y \\
&\quad z_{m,l}(k) + \sum_{i=0}^{n_x-1} \alpha_i^d z_{m,l}(k+i-n) = \sum_{i=0}^{n_x} \beta_i^{d,(m,l)} u_l(k+i-n) - \beta_i^{d,(m,l)} \epsilon_l(k+i-n) \\
&\qquad m = 1,\dots,n_y, \quad l = 1,\dots,n_u, \quad k = n_x+1,\dots,N \\
&\quad y_m(k) = z_{m,1}(k) + \dots + z_{m,n_u}(k) + \eta_m(k) \\
&\qquad m = 1,\dots,n_y, \quad k = 1,\dots,N \\
&\quad \alpha_j^d \tilde{a}_n(\alpha^c) = \tilde{a}_j(\alpha^c) \quad j = 0,\dots,n_x-1 \\
&\quad \beta_j^{d,(m,l)} \tilde{a}_n(\alpha^c) = \tilde{b}_j(\beta^{c,(m,l)}) \\
&\qquad m = 1,\dots,n_y \, j = 0,\dots,n_x, \, l = 1,\dots,n_u \\
&\quad \underline{\Delta_\eta}^{(m)} \le \eta_m(k) \le \overline{\Delta_\eta}^{(m)} \qquad m = 1,\dots,n_y, \quad k = 1,\dots,N \\
&\quad \underline{\Delta_\epsilon}^{(l)} \le \epsilon_l(k) \le \overline{\Delta_\epsilon}^{(l)} \qquad l = 1,\dots,n_u, \quad k = 1,\dots,N
\end{aligned}
\tag{4.17}
$$

where $C_m(\theta)$ represents the m-th row of matrix $C(\theta) \in \mathbb{R}^{n_y,n_x}$. Upper bounds are obtained by means of maximization over the same feasible set.

4.4 Example

In this last section, we present a simulated example about gray-box identification of the parameters of a single-input two-outputs (SITO) model, which is a particular case of MIMO systems when $n_u = 1$ and $n_y = 2$. In this example the elements of the matrices constituting the gray-box model depends by the unknown parameters as ratios of polynomials, as discussed in 4.1.
The considered gray-box model is

$$
\mathcal{M}_G = \begin{cases} \dot{x}(t) = A(\theta)x(t) + B(\theta)u(t) \\ y(t) = C(\theta)u(t) \end{cases}
$$

where

$$
A(\theta) = \begin{bmatrix} 3-\theta_2^2 & 1.2 \\ \theta_2 & \dfrac{\theta_1}{\theta_2} \end{bmatrix} \qquad B(\theta) = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \qquad C(\theta) = \begin{bmatrix} \theta_2 & -2 \\ \dfrac{\theta_1}{\theta_2} & \theta_1 \end{bmatrix}
$$

The true values of the parameters are

$$\theta = \begin{bmatrix} 1.7 \\ -2.5 \end{bmatrix}$$

and using such values a simulation experiment is performed. Characteristics of the performed experiment are

- input signal, chosen as

$$u(t) = \sum_{\omega=0}^{\omega_{\max}/0.05} A_\omega \cos 0.05\omega t$$

  with $\omega_{\max} = 3\,\mathrm{rad\,s^{-1}}$ and A_ω normally distributed random numbers;

- duration of the experiment: $T_{\exp} = 8\,\mathrm{s}$;

- sampling time $T_\mathrm{s} = 20\,\mathrm{ms}$. As a consequence $N = \dfrac{T_{\exp}}{T_\mathrm{s}} = 400$ samples of each signal are considered.

After the simulation of the system, bounded and uniformly generated random noise was added on both output sequences.

$$\begin{aligned} \tilde{y}_1(k) &= y_1(k) + \eta_1(k) \qquad -\Delta_{\eta,1} \leq \eta_1(k) \leq \Delta_{\eta,1} \quad \forall k \\ \tilde{y}_2(k) &= y_2(k) + \eta_2(k) \qquad -\Delta_{\eta,2} \leq \eta_2(k) \leq \Delta_{\eta,2} \quad \forall k \end{aligned}$$

For this reason the considered noise structure is the output error.

In order to solve the given problem, it is possible to evaluate lower bounds on the parameters to be estimated θ_1, θ_2 by means of (4.17) and upper bounds by means of the corresponding maximization. Moreover, due to the rational and polynomial nature of the dependency of $A(\theta)$ and $C(\theta)$ by the parameter θ, it is necessary to define

$$\theta_3 = \frac{\theta_1}{\theta_2} \qquad \text{and} \qquad \theta_4 = \theta_2^2$$

and therefore solve the problem in the augmented unknown vector

$$\theta_{\mathrm{aug}} = [\theta_1, \theta_2, \theta_3, \theta_4]^\top$$

including additional constraints

$$\theta_3\theta_2 - \theta_1 = 0 \qquad \text{and} \qquad \theta_2^2 - \theta_4 = 0 \tag{4.18}$$

Doing so for different amplitudes of the additive noise, it is possible to obtain what follows.
When the noise is relatively small:

$$\mathrm{SNR}_1|_{\mathrm{dB}} = 20\log_{10}\frac{\|y_1\|_2}{\|\eta_1\|_2} = 30.4\,\mathrm{dB}$$
$$\mathrm{SNR}_2|_{\mathrm{dB}} = 20\log_{10}\frac{\|y_2\|_2}{\|\eta_2\|_2} = 26.5\,\mathrm{dB}$$

Then by means of POPs (4.17) (and corresponding maximization) on θ_1 and θ_2

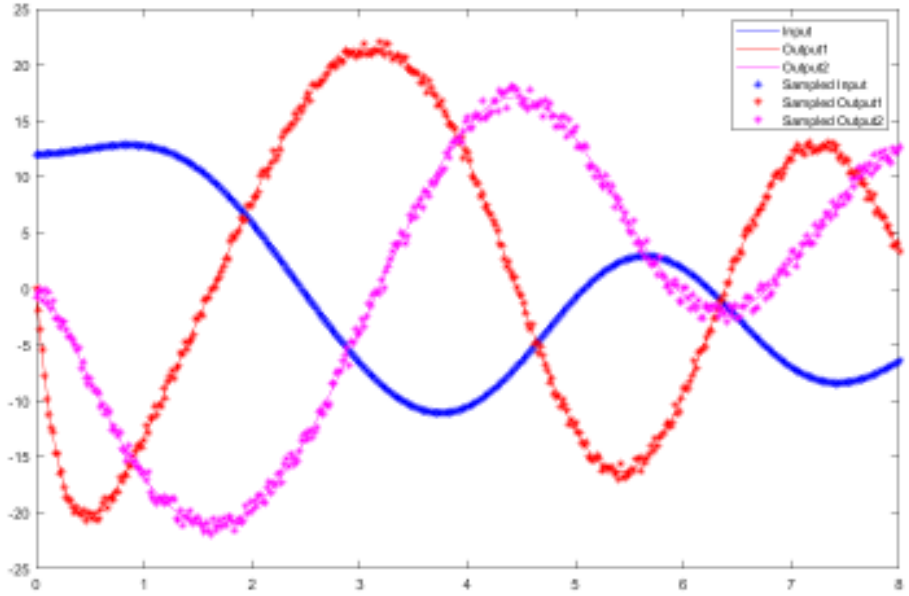

Figure 4.1: Data of simulation experiment when $\mathrm{SNR}_1 = 30.4\,\mathrm{dB}$ and $\mathrm{SNR}_2 = 26.5\,\mathrm{dB}$

using additional constraints (4.18), it is possible to obtain results in table 4.1 in a time of 279.027 s. For larger noise:

$$\mathrm{SNR}_1|_{\mathrm{dB}} = 20\log_{10}\frac{\|y_1\|_2}{\|\eta_1\|_2} = 19.4\,\mathrm{dB}$$
$$\mathrm{SNR}_2|_{\mathrm{dB}} = 20\log_{10}\frac{\|y_2\|_2}{\|\eta_2\|_2} = 16.1\,\mathrm{dB}$$

Then by means of POPs (4.17) (and corresponding maximization) on θ_1 and θ_2

	$\underline{\theta_i}$	$(\theta_i)_{\text{true}}$	$\overline{\theta_i}$	$(\theta_i)^{\text{central}}$	relative uncertainty$_{\%}$
θ_1	1.6934	1.7000	1.7859	1.7397	5.3201%
θ_2	−2.5061	−2.5000	−2.4981	−2.5021	0.31904%

Table 4.1: Results in low noise conditions

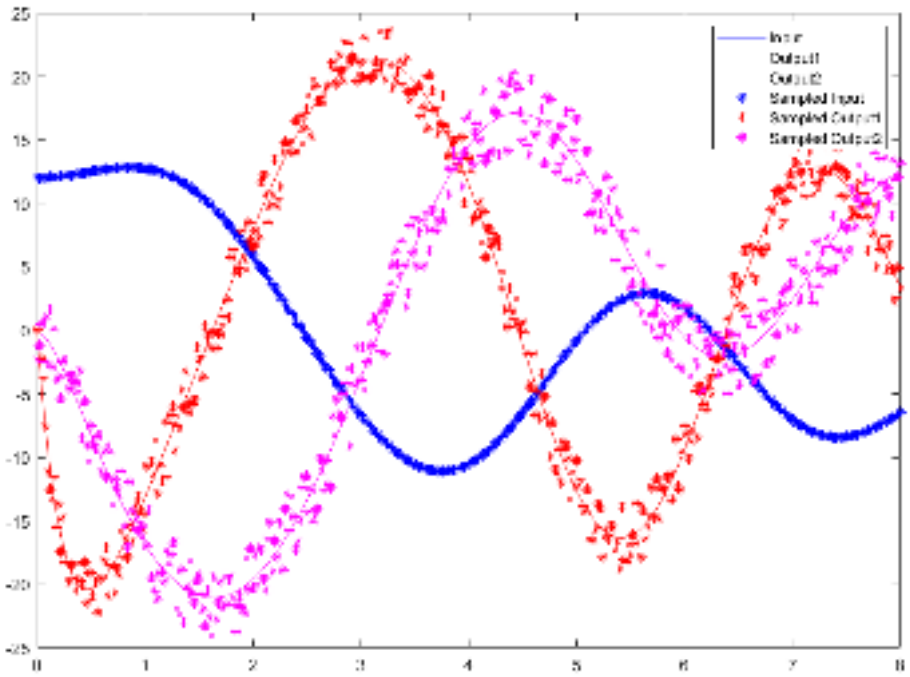

Figure 4.2: Data of simulation experiment when $\text{SNR}_1 = 19.4\,\text{dB}$ and $\text{SNR}_2 = 16.1\,\text{dB}$

using additional constraints (4.18), it is possible to obtain results in table 4.2 in a time of 308.47 s.

	$\underline{\theta_i}$	$(\theta_i)_{\text{true}}$	$\overline{\theta_i}$	$(\theta_i)^{\text{central}}$	relative uncertainty$_{\%}$
θ_1	1.6347	1.7000	1.7279	1.6813	5.5491%
θ_2	-2.5234	-2.5000	-2.4921	-2.5077	1.2513%

Table 4.2: Results in high noise conditions

Bibliography

[1] L. Ljung. *System Identification Theory for the User*. Prentice Hall, 1999 (cit. on pp. 1–4, 13).

[2] B. Wittenmark K.J. Aström. *Computer-controlled systems*. Prentice-Hall (cit. on pp. 1, 2).

[3] B. Francis J. Doyle and A. Tannenbaum. *Feedback Control Theory*. 1992 (cit. on pp. 1, 2).

[4] O. Criner. «Control Systems Identification In Finance And Economics». In: *WIT Transactions on Information and Communication Technologies* 41 (2008), pp. 2–12 (cit. on p. 1).

[5] Ronald C. Rosenberg Dean C. Karnopp Donald L. Margolis. *Modeling and Simulation of Mechatronic Systems*. 1999 (cit. on p. 2).

[6] Mario Milanese. *Teoria dei sistemi*. 1994 (cit. on p. 4).

[7] T. McKelvey. «Estimation of state space models from time and frequency data». PhD thesis. Linköping, Sweden: Linköping University, 1995 (cit. on p. 4).

[8] T. Kailath. *Linear Systems*. Prentice Hall, 1980 (cit. on p. 5).

[9] H. Piet-Lahanier M. Milanese J. Norton and E. Walter. *Bounding Approaches to System Identification*. 1996 (cit. on p. 5).

[10] A. Vicino M. Milanese. «Optimal estimation theory for dynamic systems with set membership uncertainty: An overview». In: *Automatica* 27 (1991), pp. 997–1009 (cit. on p. 5).

[11] D. Regruto V. Cerone D. Piga. «Set-Membership Error-in-Variables Identification Through Convex Relaxation Techniques». In: *IEEE Transactions on Automatic Control* 57 (2012), pp. 517–522 (cit. on pp. 7, 17).

[12] J.B. Lasserre. «Global optimization with polynomials and the problem of moments». In: *SIAM Journal on Optimization* 11 (2001), pp. 796–817 (cit. on pp. 8–10).

[13] L. El Ghaoui G.C. Calafiore. *Optimization Models*. 2014 (cit. on p. 8).

[14] M. Putinar. «Positive polynomials on compact semi-algebraic sets». In: *Indiana University Mathematics Journal* 42 (1993), pp. 969–984 (cit. on p. 10).

[15] M. Laurent. «Semidefinite representations for finite varieties». In: *Mathematical Programming* 109 (2007), pp. 1–26 (cit. on p. 10).

[16] M. Laurent J.B. Lasserre and P. Rostalski. «Semidefinite characterization and computation of real radical ideals». In: *Foundations of Computational Mathematics* 8 (2008), pp. 607–647 (cit. on p. 10).

[17] H. Waki S. Kim M. Kojima H. Sugimoto and M. Muramatsu. «SparsePOP a Sparse Semidefinite Programming Relaxation of Polynomial Optimization Problems». In: *ACM Transactions on Mathematical Software* 35 (2008). URL: `http://dx.doi.org/10.1145/1377612.1377619` (cit. on p. 11).

[18] M. Kojima H. Waki S. Kim and M. Muramatsu. «Sums of squares and semidefinite programming relaxation for polynomial optimization problems with structured sparsity». In: *SIAM Journal on Optimization* 17 (2006), pp. 218–242. URL: `http://dx.doi.org/10.1137/050623802` (cit. on p. 11).

[19] J. F. Sturm. «Using SeDuMi 1.02, a MATLAB Toolbox for optimization over symmetric cones». In: *Optim. Methods Software* 11 (1999), pp. 625–653 (cit. on p. 11).

[20] MOSEK. *Documentation and API reference*. 2019. URL: `www.mosek.com/documentation/` (cit. on p. 11).

[21] Olivier Prot and Guillaume Mercère. «Combining Linear Algebra and Numerical Optimization for Gray-Box Affine State Space Model Identification». In: *IEEE Transactions on Automatic Control* 65 (Aug. 2020), pp. 3272–3285 (cit. on pp. 12, 13).

[22] H.M. Möller and H.J. Stetter. «Multivariate polynomial equations with multiple zeros solved by matrix eigenproblems». In: *Numerische Mathematik* 70 (1995), pp. 311–329 (cit. on p. 14).

[23] Vito Cerone. «Parameter bounds for ARMAX models from records with bounded error in variables». In: 57 (1993), pp. 225–235 (cit. on p. 17).

[24] Vito Cerone. «Feasible Parameter Set for Linear Models with Bounded Errors in All Variables». In: *Automatica* 29 (1993), pp. 1551–1555 (cit. on p. 17).

[25] David J.N. Limebeer Michael Green. *Linear Robust Control*. 2012 (cit. on p. 31).

[26] Kang-Zhi LiuYu Yao. *Robust Control: Theory and Applications*. 2016 (cit. on p. 31).

[27] M. Mensler H. Garnier and A. Richard. «Continuous-time model identification from sampled data: implementation issues and performance evaluation». In: 76 (2003), pp. 1337–1357 (cit. on pp. 31, 32).

[28] H. Unbehauen and G. P. Rao. «Continuous Time Approaches to System Identification - A Survey». In: 26 (1990), pp. 23–35 (cit. on p. 32).

[29] Rolf Johansson. *System Modeling and Identification*. Prentice-Hall International, Inc., 1993, pp. 280–285 (cit. on p. 33).

[30] *Subspace Identification for Linear Systems: Theory — Implementation — Applications*. 1996 (cit. on p. 36).

[31] Diego Regruto Tomalino Vito Cerone Valentino Razza. «Set-membership errors-in-variables identification of MIMO linear systems». In: *Automatica* 90 (2018), pp. 25–37 (cit. on pp. 70, 73).