



UNIVERSITÀ DEGLI STUDI DI SIENA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
DOTTORATO IN INGEGNERIA DELL'INFORMAZIONE - CICLO XV

**SET-MEMBERSHIP ESTIMATION:
AN ADVANCED TOOL
FOR SYSTEM IDENTIFICATION**

Candidato:

Marco Casini

Anno Accademico 2001/2002

Contents

Introduction	1
 I Set-Membership System Identification: Theoretical Developments	 5
 1 Set-membership estimation theory	 7
1.1 Problem formulation	8
1.1.1 Spaces and operators	8
1.1.2 Uncertain sets, admissible sets	11
1.1.3 Errors, radius of information	14
1.2 Estimation algorithm properties	15
1.2.1 Optimal algorithms	16
1.2.2 Other properties of algorithms	18
1.3 Pointwise estimators	20
1.4 Set estimators	23
1.5 Set-membership system identification	25

2	Pointwise Estimators	27
2.1	Linear problems	27
2.1.1	Central algorithm	28
2.1.2	Linear algorithms	31
2.1.3	Projection algorithms	32
2.2	Asymptotic properties of algorithms	38
2.3	Properties in the nonlinear case	40
3	Conditional estimation	43
3.1	Problem formulation	44
3.2	Conditional pointwise estimators	45
3.3	Conditional estimation with energy bounded noise	49
3.3.1	\mathcal{H}_2 set-membership identification	51
3.4	Properties of conditional algorithms	52
4	Orthonormal basis functions in conditional identification	57
4.1	Orthonormal basis functions	58
4.2	Classes of orthonormal basis functions	60
4.2.1	Laguerre functions	60
4.2.2	Optimal pole selection in Laguerre expansion	61
4.2.3	Kautz functions	63

4.2.4	Generalized orthonormal basis functions	65
4.3	Orthonormal basis functions in conditional identification	67
4.4	Suboptimal pole choice and error bounds	69
4.4.1	Error bounds for ellipsoidal feasible sets	76
4.5	Simulation examples	79
5	Optimal input design	85
5.1	Experiment design in system identification	86
5.2	Time and model complexity for fast identification	87
5.2.1	Separation of input design and model selection	88
5.2.2	Noise free optimal input design	90
5.2.3	Optimal affine representation	91
5.3	Optimal input for energy bounded noise	94
5.3.1	Problem formulation	94
5.3.2	Evaluation of worst-case error bounds	98
6	Example of application	103
6.1	Process description	103
6.2	Identification procedure	104
6.2.1	Set-membership identification	106
6.2.2	Statistical identification	110

II	ACT: a Remote Laboratory of Automatic Control and Identification	117
7	The <i>Automatic Control Telelab</i>	119
7.1	Remote laboratories; state of the art	120
7.2	Features of the Automatic Control Telelab	123
7.3	A Session Description	128
7.3.1	User-defined controller	130
7.3.2	The tank level and magnetic levitation examples	131
7.3.3	Running the experiments	134
7.4	Student competition overview	138
7.4.1	A competition session description	140
7.4.2	Teaching experiences	144
7.5	Remote system identification	148
7.5.1	An identification session description	148
7.6	The ACT Architecture	154
	Bibliography	157

Introduction

This dissertation is divided in two parts: the first one concerns set-membership theory for model estimation and identification, while the second one deals with remote laboratories of automatic control, and in particular with the *Automatic Control Telelab*, a remote lab developed at University of Siena which allows the remote identification and control of physical processes, and which has been used to apply set-membership techniques described in the first part on real systems.

In the first part, the set-membership approach to system estimation (identification) is addressed. The aim of the estimation problem is to obtain a dynamic model of the system from noisy input-output measurements. Depending on the hypothesis on the noise, it is possible to distinguish between a statistical and a deterministic approach. The main difference between the classical (statistical) estimation and the set-membership (deterministic) one lies in the fact that in statistical estimation noise is represented as a stochastic process (usually a filtered white noise), while in set-membership estimation noise is supposed to be unknown but bounded, i.e. the only knowledge about noise consists in its bounds evaluated in a given norm.

In statistical estimation, uncertainty is described in terms of confidence in-

tervals (soft bounds). On the contrary, in the deterministic approach a set of all admissible solutions is found (hard bounds). In this case, such a set contains all the feasible solutions of the problem, thus providing an evaluation of the uncertainty associated to the estimation problem. For this reason this approach is usually called set-membership. Moreover, while the statistical estimation deals with the average case, the deterministic theory usually considers the worst-case, that is the estimate that shows the best performance in a worst-case setting.

Set-membership theory was born at the end of the Sixties and was applied to problems of state estimation of dynamical systems [1, 2]. In the Eighties, this approach deserved interest due to the development of robust control theory; in fact, by giving hard bounds on the uncertainty, this theory provides models that are useful in the robust control context.

In the first part of the thesis the main aspects of the set-membership approach to system estimation and identification are described. The purpose is not to provide a complete survey of these topics, but to present a general description of the main concepts and a more detailed treatment of some specific problems of interest in set-membership identification. In particular, the main original contribution of this thesis regards conditional estimation problems and optimal input design. In the former case, the evaluation of the optimality degree of almost-optimal algorithms, for different model classes, is investigated. The optimal input design problem concerns the choice of the best input to apply when performing system identification in a set-membership context. In particular, the case of noise bounded in the ℓ_2 norm (energy bounded noise) is analyzed in depth.

In the following a short description of chapter contents is reported.

In Chapter 1 the general set-membership estimation problem is introduced, along with all definitions and main concepts which will be used in the following

development. Properties of different estimation algorithms are also described, and the classes of pointwise and set estimators are introduced.

In Chapter 2 a more detailed description of pointwise estimators is reported. Particular emphasis is given to the linear setting, for which the main classes of pointwise estimators are described. Optimality and asymptotic properties of such estimators are also outlined.

Chapter 3 is focused on the theory of conditional estimation (or restricted-complexity estimation), in which an estimate is restricted to belong to a pre-defined set.

In Chapter 4 an overview of orthonormal basis functions is reported. Such functions are commonly used to obtain restricted-complexity model classes, as described in Chapter 3. Particular emphasis is given to Laguerre functions, and to the optimal choice of pole location. New tight bounds on the optimality degree of suboptimal algorithms for different pole location of the basis function are reported.

The problem of optimal input design is described in Chapter 5. A general overview of such problem along with some results in the noise free case is addressed. New results about input design in the energy bounded noise case are also reported.

In Chapter 6 an example of application of set-membership and statistical identification is reported. The identification experiment has been performed by means of the *Automatic Control Telelab*, the remote laboratory described in Part 2.

The second part of this thesis is focused on remote laboratories of automatic control and in particular on the *Automatic Control Telelab (ACT)*, a remote lab developed at the Dipartimento di Ingegneria dell'Informazione of the University of Siena. By means of remote labs it is possible to perform remote

experiments, through the Internet or other kinds of networks, on real processes. This is the main feature which distinguishes remote labs from virtual labs which only provide software simulation of physical systems. The state of the art along with a comparison between virtual and remote labs is reported.

The Automatic Control Telelab is mainly used for educational purpose, with the aim of helping students to practice their knowledge of control systems in an easy way and from any computer connected to the Internet. A detailed description of the features of the ACT is reported. A key feature which distinguishes the ACT from other remote labs is the possibility to design a user-defined controller by means of a Simulink model in a very easy way.

A typical session description is reported, in which it is explained how to interact with remote processes. The user can choose the kind of controller to use among a set of predefined ones, or can design a new controller by himself. During the experiment, it is possible to change controller parameters as well as reference signals, and to view on-line input and output signals and a live video of the experiment.

A new feature of the ACT is the presence of a *student competition*, a mechanism which allows students to compete in designing the best controller for achieving some predefined performance. Performances are automatically evaluated and the designed controllers are ranked accordingly.

An important novelty, introduced by ACT is the presence of tools for performing system identification of the remote processes. Through this feature it is possible to choose the input to apply to the system in order to perform a system identification procedure by means of statistic and set-membership techniques.

Part I

Set-Membership System Identification: Theoretical Developments

Set-membership estimation theory

In this chapter the basic concepts about set-membership estimation theory are presented. The formalism is similar to that used in [3] to describe the optimal algorithms theory. Through this formalism it is possible to formulate a wide variety of problems in the same context.

The general estimation problem can be summarized as follows:

Given an unknown element x , find an estimate of the function $S(x)$, based on a priori information K and on measurements of the function $F(x)$ corrupted by additive noise e .

Of course the solution of this problem depends on the functions $S(\cdot)$ and $F(\cdot)$, on the a priori information and in particular on the assumptions on the noise e . Set-membership theory deals with unknown but bounded (UBB) noise, that is noise bounded in some norm.

The chapter is organized as follows. In the first section the estimation prob-

lem is formalized and its essential elements are described (spaces, operators, sets, etc.). In Section 1.2 the main properties of estimation algorithms are reported and the concepts of optimality and almost-optimality are introduced. In Sections 1.3 and 1.4 the main classes of pointwise and set estimators are described, while in Section 1.5 a specific set-membership system identification problem is cast in the considered framework.

1.1 Problem formulation

1.1.1 Spaces and operators

A generic estimation problem can be formulated in the following spaces:

- X : problem elements space;
- Y : measurements space;
- Z : solution space.

We suppose that X , Y and Z are normed linear spaces, of dimension respectively n , m , p . Moreover, we denote with $\|\cdot\|_X$, $\|\cdot\|_Y$ and $\|\cdot\|_Z$ the associated norms in such spaces.

We define *solution operator* $S : X \rightarrow Z$, the function which associates to every element $x \in X$, the quantity we want to estimate

$$z = S(x). \tag{1.1}$$

Estimation is based on available information on x , which is fundamentally of two different kinds.

1. *A priori information*

It is usually represented by a subset K of X , to which the problem element must belong, i.e.

$$x \in K \subseteq X. \quad (1.2)$$

The set K typically has some structural characteristics (i.e. convexity, symmetry, etc.). A common example is

$$K = \{x \in X : \|L(x - x_0)\|_X \leq 1\}$$

where L is a linear operator and $x_0 \in X$.

2. *A posteriori information (measurements)*

It is represented by the knowledge of a certain function $F(x)$, where $F : X \rightarrow Y$ is the *information operator*. Usually, the knowledge of $F(x)$ is not exact, but it is corrupted by noise on the measurement process. If we assume additive noise, the available observations $y \in Y$ are

$$y = F(x) + e. \quad (1.3)$$

A crucial aspect of the estimation problem is the hypothesis on the noise e . In fact, the procedure adopted to solve the problem as well as the characteristics of the solution (optimality, asymptotic behaviour, etc.) strongly depend on this hypothesis.

Unknown But Bounded (UBB) noise. The noise e is assumed to be bounded in norm

$$\|e\|_Y \leq \varepsilon \quad (1.4)$$

for some constant $\varepsilon > 0$.

We define an *estimation algorithm* (or *estimator*) as an operator $\Phi(\cdot)$, $\Phi : Y \rightarrow Z$, which provides an approximation $\Phi(y)$ of the quantity to be estimated $S(x)$. In other words, the target of the estimation problem is to determine

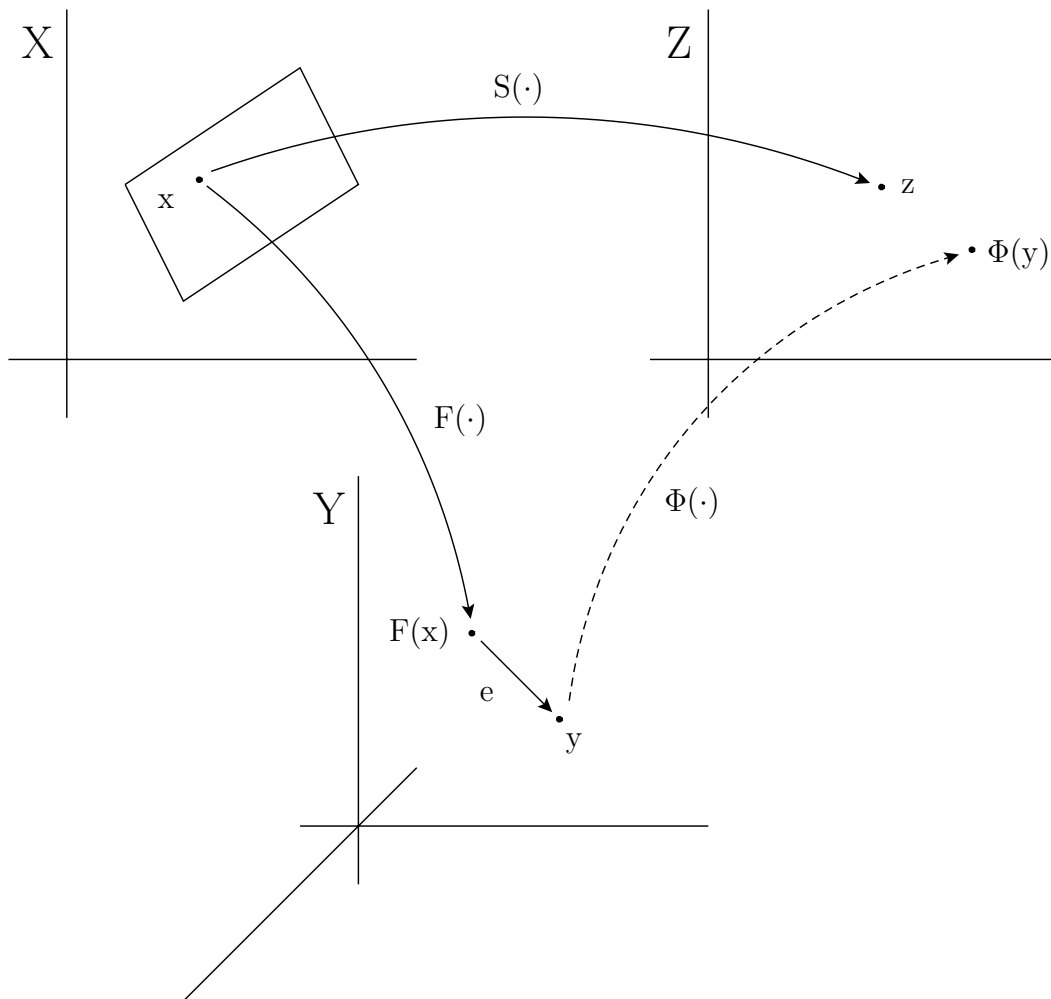


Figure 1.1: Illustrative picture of a generic estimation problem.

an algorithm Φ , such that $\Phi(y) \simeq S(x)$. A schematic representation of the problem previously described is reported in Fig.1.1.

In general, $\Phi(\cdot)$ can be a single- or multi-valued function. It is denoted as *pointwise estimator* if $\Phi(\cdot)$ associates one element (point) in the solution space to every vector in the measurements space. On the contrary, a *set estimator* is a function $\Phi(\cdot)$ which determines a set of elements in the Z space. In the following we will denote by $\Phi(\cdot)$ a generic pointwise estimator.

1.1.2 Uncertain sets, admissible sets

Due to the hypothesis of UBB noise, the main elements of the estimation problem can be characterized by means of suitable sets (set-membership approach). In the following, the sets which play an important role are defined.

- *Measurement Uncertain Set*

$$MUS_y = \{\tilde{y} \in Y : \|y - \tilde{y}\|_Y \leq \varepsilon\}. \quad (1.5)$$

It is the set containing all the measurements whose distance from the observation y is less or equal to ε . The MUS_y contains all the “exact” information that could have generated the noisy measurement y . In fact, if x is the unknown element to be estimated, and y is the available observation, it follows immediately that $F(x) \in MUS_y$.

- *Estimation Uncertain Set*

$$EUS_\Phi = \Phi(MUS_y). \quad (1.6)$$

For a fixed estimator Φ , it is the set containing all estimates that can be obtained from measurements belonging to MUS_y .

- *Feasible Parameter Set*

$$FPS_y = \{x \in K : \|y - F(x)\|_Y \leq \varepsilon\}. \quad (1.7)$$

It is the set of the problem elements which are compatible with all the available information: the structure of the operator $F(\cdot)$, the hypothesis of UBB noise, the a priori information K and the observed measurements y . Note that if FPS_y is empty, the observations are not consistent with the problem formulation, that is with $F(\cdot)$, K and ε .

- *Feasible Solution Set*

$$FSS_y = S(FPS_y). \quad (1.8)$$

It is the set of admissible solutions, compatible with the available information in the estimation problem.

In several problems of parametric and non parametric estimation, the solution operator $S(\cdot)$ coincides with the identity operator, that is the purpose of the estimation problem is the same element of the problem ($S(x) = x$). In this case, FPS_y coincides with FSS_y , and it is usually called *feasible set*.

In general, the previously defined sets may have a very complex structure (non-convex, non-connected, etc.). A special situation occurs when the information operator is linear, that is $F(x) = Fx$, as it happens in many problems of interest. Note that in this case, if $K = X$, FPS_y is the counterimage through F of the set $MUS_y \cap \mathcal{R}(F)$, where $\mathcal{R}(F)$ denotes the image of F . The structure of FPS_y depends on the used norm for the measurement space, under the UBB hypothesis, as described in Table 1.1¹.

$\ \cdot\ _Y$	FPS_y
ℓ_∞	polytope
ℓ_2	ellipsoid
ℓ_1	polytope

Table 1.1: Structure of FPS_y , depending on the Y norm (linear case).

¹In the following, it will be denoted by $\ell_\infty, \ell_2, \ell_1$ the usual norms on semi-infinite spaces, as well as the vectorial norms $\infty, 2, 1$ in spaces of finite dimension.

If $m < n$, FPS_y is not bounded. On the contrary, if $m \geq n$ (the most frequent situation in estimation problems), then FPS_y is bounded if and only if $\text{rank}(F) = n$. This condition is sometime denoted as *sufficient information* [4].

Analyzing these sets and their properties, some geometric features commonly used play an important role, like the concepts of *center* and *radius*. In the following some useful definitions are reported.

Definition 1.1 Consider a set $I \subset Z$. The Chebyshev center of I is defined as

$$\text{cen}(I) = \arg \inf_{z \in Z} \sup_{\tilde{z} \in I} \|z - \tilde{z}\|_Z. \quad (1.9)$$

The Chebyshev radius of I is given by

$$\text{rad}(I) = \sup_{\tilde{z} \in I} \|\text{cen}(I) - \tilde{z}\|_Z. \quad (1.10)$$

In other words, $\text{cen}(I)$ is the center of the minimum radius ball (in the Z norm) containing I . Note that in general:

- $\text{cen}(I)$ is not unique.
- $\text{cen}(I)$ may not belong to I (even if I is convex).

For example, if I is a polytope and $\|\cdot\|_Z = \ell_1$, it is possible to find examples where $\text{cen}(I) \notin I$ for spaces Z of dimension greater than 2. If the set I has a symmetry center, obviously $\text{cen}(I)$ coincides with it.

Definition 1.2 The diameter of a set $I \subset Z$ is given by

$$\text{diam}(I) = \sup_{z_1, z_2 \in I} \|z_1 - z_2\|_Z. \quad (1.11)$$

From the previous definitions it follows immediately that

$$\text{rad}(I) \leq \text{diam}(I) \leq 2\text{rad}(I), \quad \forall I \subset Z. \quad (1.12)$$

1.1.3 Errors, radius of information

Let Φ be a pointwise estimator. A measure of the quality of the estimation provided by Φ is given by the distance

$$\|S(x) - \Phi(y)\|_Z \quad (1.13)$$

which depends on the unknown element x as well as on the observation y . In the context of set-membership estimation, it is common to evaluate (1.13) with respect to the worst case problem element and/or the worst measurement. That is, we use error measures in a *worst-case* setting. It is possible to define the following errors.

- *Y-local error*

$$E_y(\Phi, \varepsilon) = \sup_{x \in FPS_y} \|S(x) - \Phi(y)\|_Z \quad (1.14)$$

- *X-local error*

$$E_x(\Phi, \varepsilon) = \sup_{y \in MUS_{F(x)}} \|S(x) - \Phi(y)\|_Z \quad (1.15)$$

- *Global error*

$$E(\Phi, \varepsilon) = \sup_{x \in X} E_x(\Phi, \varepsilon) = \sup_{y \in Y_0} E_y(\Phi, \varepsilon) \quad (1.16)$$

where $Y_0 = \{y \in Y : FPS_y \neq \emptyset\}$.

Remark 1.1 *In the following, we will denote the feasible sets FSS_y^m and the errors $E(\Phi, \varepsilon, m)$, when it is important to emphasize the dependance on the number of observations m .*

It is useful to stress the difference between *Y-local* error and *X-local* error. The former is an *a posteriori* measure of the goodness of the estimation, since it is based on the knowledge of the observed measurements and on the set FPS_y . On the contrary, the latter can be considered as an *a priori* measure,

since it is function of the problem element (usually unknown) and it is not based on the observations. The meaning of the previously considered errors will be deeply investigated in Section 1.2.

The minimum global error which can be achieved by an estimation algorithm is defined as *radius of information*

$$R(\varepsilon) = \inf_{\Phi} E(\Phi, \varepsilon). \quad (1.17)$$

The name “radius” comes from the fact that $R(\varepsilon)$ can be computed from the set of the feasible solutions in the following way [5].

Proposition 1.1

$$R(\varepsilon) = \sup_{y \in Y_0} \text{rad}(FSS_y).$$

Another variable commonly used to evaluate the performance of estimators is the *diameter of information*, defined as

$$D(\varepsilon) = \sup_{y \in Y_0} \sup_{x_1, x_2 \in FPS_y} \|S(x_1) - S(x_2)\|_Z = \sup_{y \in Y_0} \text{diam}(FSS_y). \quad (1.18)$$

From (1.12) it follows immediately

$$R(\varepsilon) \leq D(\varepsilon) \leq 2 R(\varepsilon). \quad (1.19)$$

1.2 Estimation algorithm properties

In this section some properties of pointwise estimators are reported. They mainly refer to the notion of estimation error and radius of information, previously introduced.

1.2.1 Optimal algorithms

Optimality of an algorithm is relative to the type of error which is minimized.

Definition 1.3 *An algorithm Φ^* is X -locally optimal if*

$$E_x(\Phi^*, \varepsilon) \leq E_x(\Phi, \varepsilon) \quad \forall x \in X, \forall \Phi.$$

Definition 1.4 *An algorithm Φ^* is Y -locally optimal if*

$$E_y(\Phi^*, \varepsilon) \leq E_y(\Phi, \varepsilon) \quad \forall y \in Y_0, \forall \Phi.$$

Definition 1.5 *An algorithm Φ^* is globally optimal if*

$$E(\Phi^*, \varepsilon) \leq E(\Phi, \varepsilon) \quad \forall \Phi.$$

From the last definition and from (1.17) it follows immediately that for a globally optimal algorithm, the global error is equal to the radius of information

$$E(\Phi^*, \varepsilon) = R(\varepsilon).$$

Note the different meaning of the concept of local optimality, in the Definitions 1.3 and 1.4. In an estimation problem where the measurements y are available, the best algorithm is the one which minimizes the estimation error related to the worst element x which is compatible with the observations, for all possible values of y (Y -local optimality). On the contrary, if measurements are not known, an algorithm X -locally optimal gives the best estimation related to the worst value of y , for all unknown elements $x \in X$.

It is useful to remark that local optimality is stronger than global optimality.

Proposition 1.2 *If Φ is a Y -locally optimal algorithm (X -locally optimal algorithm), then it is globally optimal too.*

Proof. Let Φ be Y -locally optimal. If

$$\bar{y} = \arg \sup_{y \in Y_0} E_y(\Phi, \varepsilon)$$

we have

$$E(\Phi^*, \varepsilon) = E_{\bar{y}}(\Phi^*, \varepsilon) \leq E_{\bar{y}}(\Phi, \varepsilon) \leq E(\Phi, \varepsilon) \quad \forall \Phi$$

where the first inequality derives from Y -local optimality, and the second one from the definition of global error. If Φ^* is X -locally optimal the proof is similar. \square

On the contrary, it is possible to prove that, in general, a globally optimal algorithm is not necessarily locally optimal.

Remark 1.2 *From Definition 1.3, an algorithm X -locally optimal is the algorithm which solves the problem*

$$\inf_{\Phi} E_x(\Phi, \varepsilon) \tag{1.20}$$

for all $x \in X$. However, being $E_x(\Phi, \varepsilon)$ a function of x , in general also the solution of problem (1.20) depends on x . On the other side, an estimation algorithm is a function of the observations y , but not of the problem element x which is unknown. Indeed an algorithm X -locally optimal exists only if $\inf_{\Phi} E_x(\Phi, \varepsilon)$ does not depend on x . A particular case occurs when $E_x(\Phi, \varepsilon)$ is independent of x , and so $E_x(\Phi, \varepsilon) = E(\Phi, \varepsilon)$ and X -local optimality coincides with global optimality.

In many applications, optimal algorithms can have a high computational burden. For this reason, almost-optimal algorithms are usually employed. For these kinds of algorithms it is possible to evaluate the worst-case performances.

Definition 1.6 *An algorithm Φ^* is globally almost-optimal (or optimal within a factor k) if*

$$E(\Phi^*, \varepsilon) \leq k \inf_{\Phi} E(\Phi, \varepsilon) = k R(\varepsilon).$$

Analogously it is possible to define X -local and Y -local almost-optimal algorithms.

1.2.2 Other properties of algorithms

In addition to the optimality concept, there are other properties that are usually requested for a pointwise estimator.

Definition 1.7 *An algorithm Φ is correct if*

$$\Phi(F(x)) = S(x) \quad \forall x \in X.$$

Correctness is the ability to reconstruct exactly the solution from observations not corrupted by noise. It is essentially related to the concept of unbiased estimation in the statistical theory of estimation. Note that the previous definition is valid only if $p \leq m$, that is if we have enough measurements with respect to the parameters to be estimated (this is the most common situation in estimation problems). In the linear case, we have the following property.

Proposition 1.3 *Let $m \geq p$ and $K = X$. If $F(x) = Fx$ and $S(x) = Sx$, an estimation algorithm is correct if and only if*

$$\ker(F) \subseteq \ker(S). \quad (1.21)$$

Proof. If (1.21) does not hold, there exists $\bar{x} \in X$ such that $F\bar{x} = 0$ and $S\bar{x} \neq 0$. On the other side, a correct algorithm must satisfy $\Phi(0) = 0$, and so we have $\Phi(F\bar{x}) = \Phi(0) = 0 \neq S\bar{x}$, which contradicts Definition 1.7.

On the contrary, we suppose that (1.21) holds, and let $y = Fx$, for some $x \in X$. Let $x = x_1 + x_2$, with $x_1 \in \ker(F)$ and $x_2 \in (\ker(F))^\perp$, so we have $y = Fx_2$ and by (1.21), $Sx_1 = 0$. Moreover, if $\dim(\ker(F)) = k$, we have $\text{rank}(F) = n - k$ and so it is possible to evaluate x_2 by solving the system $Fx_2 = y$. Fixing

$\Phi(y) = Sx_2$ we have $\Phi(Fx) = \Phi(y) = Sx_2 = Sx_1 + Sx_2 = Sx$ and so Φ is correct. \square

Note that if F defines an injective transformation, we have $\ker(F) = \emptyset$ and so (1.21) is always satisfied.

Another important property of an estimation algorithm is related to the ability to provide an estimate compatible with the available information.

Definition 1.8 *An estimation algorithm Φ is an interpolatory algorithm if*

$$\Phi(y) \in FSS_y.$$

Interpolatory algorithms are very important, also thanks to the following property.

Proposition 1.4 *An interpolatory algorithm is Y -locally (and then also globally) optimal within a factor 2.*

Proof. Let Φ_i be an interpolatory algorithm and let $x_y \in FPS_y$ such that $\Phi_i(y) = S(x_y)$. From the definition of Y -local error (1.14) it follows that

$$\begin{aligned} E_y(\Phi_i, \varepsilon) &= \sup_{x \in FPS_y} \|S(x) - \Phi_i(y)\|_Z = \\ &= \sup_{x \in FPS_y} \|S(x) - S(x_y)\|_Z \leq \\ &\leq \text{diam}(FSS_y) \leq 2 \text{rad}(FSS_y) \leq 2 E_y(\Phi, \varepsilon) \quad \forall \Phi, \forall y \end{aligned}$$

and then Φ_i is Y -locally almost-optimal. \square

Other important properties concern the asymptotic behaviour of algorithms, that is the limit value of estimation errors when the measurements number tends to infinity.

Definition 1.9 *An algorithm Φ is asymptotically convergent if*

$$\lim_{\varepsilon \rightarrow 0} \lim_{m \rightarrow \infty} E_x(\Phi, \varepsilon, m) = 0 \quad \forall x \in K.$$

Definition 1.10 *An algorithm Φ is robustly convergent if*

$$\lim_{\varepsilon \rightarrow 0} \lim_{m \rightarrow \infty} E(\Phi, \varepsilon, m) = 0.$$

From the definition of global error, it follows that robust convergence implies asymptotic one, but not conversely.

In many cases, robust convergence analysis can be related to that of the asymptotic behaviour of the diameter of information $D(\varepsilon)$. In fact, if we adopt globally optimal or almost-optimal algorithms, robust convergence turns out to be convergence of the radius of information, which from (1.19) holds if and only if

$$\lim_{\varepsilon \rightarrow 0} \lim_{m \rightarrow \infty} D(\varepsilon) = 0. \quad (1.22)$$

On the contrary, if (1.22) is not satisfied, there cannot exist any robustly convergent algorithm. To obtain a robustly convergent algorithm it is necessary to modify some element of the estimation problem (the experiment $F(\cdot)$, the a priori information K , etc.). The analysis of the asymptotic behaviour of the diameter is justified by the fact that in general the computation of $D(\varepsilon)$ (or an upper limit of it) is easier with respect to $R(\varepsilon)$.

It is worthwhile to remark that the convergence properties previously considered are useful especially when the X space has infinite dimension. We will see in the following (Section 2.2) that if N is finite, the convergence of the diameter of information is easily guaranteed by a sufficient information hypothesis.

1.3 Pointwise estimators

In this section two classes of pointwise estimators are presented: the *central algorithms* and the *projection algorithms*. These classes are important for their properties and because they are widely used in many estimation problems.

Since set-membership theory allows to characterize the set of all feasible solutions of an estimation problem, the definition of a pointwise estimator consists essentially in the selection of a representative element within a set. A typical choice is to choose the Chebyshev center of the set.

Definition 1.11 *A central algorithm Φ_c is defined as*

$$\Phi_c(y) = \text{cen}(FSS_y).$$

From this definition one obtains

$$E_y(\Phi_c, \varepsilon) = \text{rad}(FSS_y).$$

The characteristics of the central algorithm obviously depend on the structure of FSS_y . In general these algorithms are computationally hard and for this reason it is usually preferred to employ easier estimators. A very important class is that of projection algorithms.

Definition 1.12 *A projection algorithm is defined as*

$$\Phi_p(y) = S(x_p)$$

where

$$x_p = \arg \min_{x \in K} \|y - F(x)\|_Y. \quad (1.23)$$

In the linear case, if $K = X$, a projection algorithm provides, as selected element of the problem, the counterimage through F of the projection (in norm $\|\cdot\|_Y$) of y on the subspace $\mathcal{R}(F)$ (see Fig. 1.2).

It is interesting to look at the various interpretations of the projection algorithm depending on the norm in the Y space. If we use the ℓ_2 norm we obtain the popular *least squares* algorithm (Φ_{LS}), widely used in many estimation and

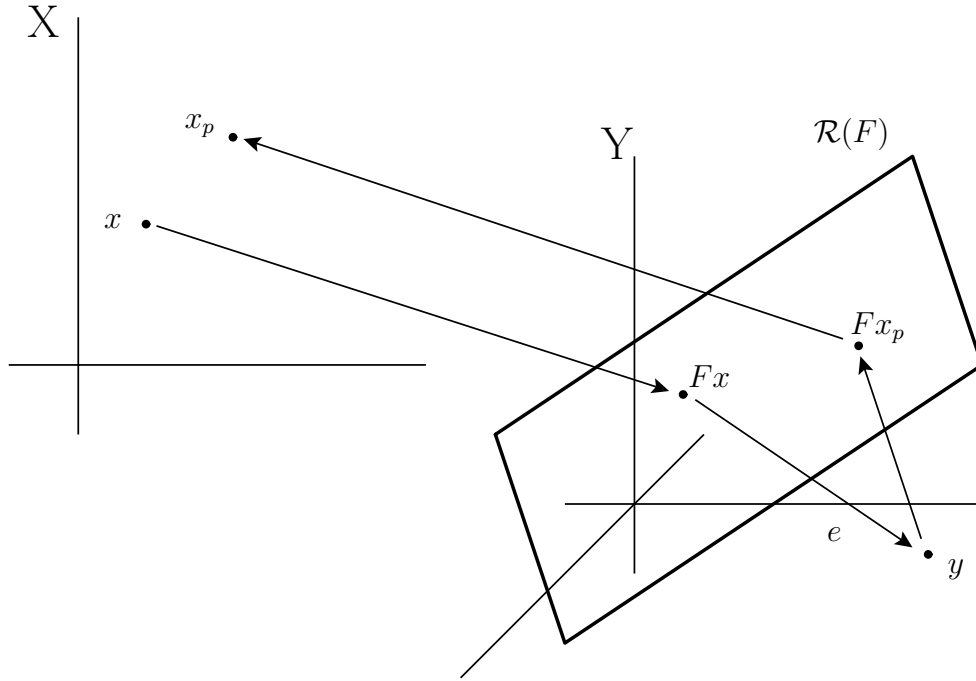


Figure 1.2: Graphical example of the projection algorithm when $\|\cdot\|_Y = \ell_2$ (orthogonal projection of y on $\mathcal{R}(F)$).

identification problems. On the contrary, using ℓ_∞ and ℓ_1 norms, we obtain respectively the *minimum maximum difference* estimator and the *minimum error sum* estimator.

It is useful to remark that projection algorithms have important properties also in other contexts, as for instance in the statistical theory of estimation. In this case, the projection algorithm in a given norm results to be the maximum likelihood estimator (MLE), under some hypotheses on the measurement noise distribution. The various interpretations in the statistical and deterministic contexts are reported in Table 1.2.

$\ \cdot\ _Y$	Deterministic estimation	Probabilistic estimation
ℓ_2	Least squares	MLE for gaussian noise
ℓ_∞	Minimum maximum difference	MLE for uniform noise
ℓ_1	Minimum error sum	MLE for Laplace noise

Table 1.2: Interpretations of projection algorithm depending on the Y norm.

1.4 Set estimators

In Section 1.1 we denoted as *set estimator* a function $\Phi(\cdot)$ which associates a set I of elements of the solution space to a vector y of the measurements space

$$\Phi(y) = I \subset Z.$$

Since in set-membership theory the information regarding the estimation problem is described by means of sets (see Section 1.1.2), a natural way to proceed is to represent the estimate through a set of admissible solutions. These estimators can be essentially classified in

- exact algorithms;
- approximated algorithms.

The former can describe exactly the set of feasible solutions, i.e.

$$\Phi(y) = FSS_y.$$

In many cases this approach is not computationally tractable, since the FSS_y is too complex (nonlinear, non convex, etc.). For this reason great attention has been devoted to approximated algorithms. They can be mainly divided into inner and outer approximations. The most common approaches use ellipsoids or orthotopes as approximating regions (see Fig. 1.3), and provide strategies

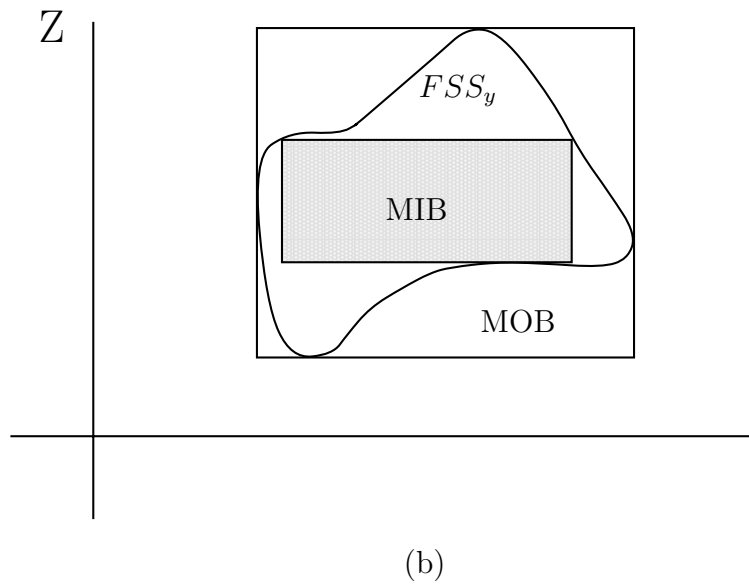
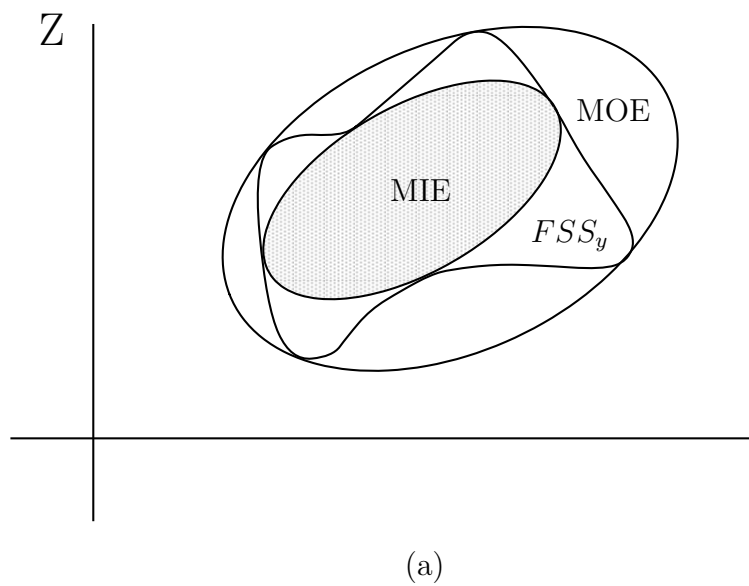


Figure 1.3: Inner and outer approximations of the FSS_y by means of ellipsoids (a) and orthotopes (b).

to compute in an efficient way the outer approximations of minimum volume (MOE, MOB), or the inner approximations of maximum volume (MIE, MIB).

Under linear hypotheses, using ℓ_∞ or ℓ_1 norm in the Y space, it is known that FSS_y is a polytope (see Table 1.1). Nevertheless also the description of a generic polytope can be computationally hard, due to the large number of faces and vertexes. In [6, 7] it has been shown that recursive approximation of FSS_y by means of parallelotopes is a viable compromise between quality of the approximation and required computational burden.

1.5 Set-membership system identification

In this section an application of set-membership theory is reported. Let us consider a LTI discrete time SISO system, described by the transfer function

$$H(z) = \sum_{k=0}^{\infty} h_k z^{-k}.$$

Let $h = \{h_i\}_{i=0}^{\infty}$ be the impulse response sequence of the system, and let h belong to a linear normed space X equipped with the norm $\|\cdot\|_X$. Note that in this case X is an infinite dimensional space.

Data consists of N input/output measurements $\{(u_k, y_k), k = 0, \dots, N-1\}$, related by

$$y_k = \sum_{i=0}^k h_i u_{k-i} + v_k \quad k = 0, 1, \dots, N-1. \quad (1.24)$$

The input sequence u is usually bounded in the ℓ_∞ norm, and so we can assume $\|u\|_\infty \leq 1$. The noise sequence v is assumed to be unknown but bounded, i.e.

$$\|v\|_Y \leq \varepsilon.$$

The aim of the problem is to identify the first p elements of the impulse response ($p \leq N$).

It is possible to rewrite the problem in vector form, by following the scheme reported in Section 1.1. In particular:

- X is the infinite dimensional space of impulse responses $x = \{h_k\}_{k=0}^{\infty}$. A priori knowledge on such system is expressed as $h \in K$, where K is a set contained in X . An example of this set may be (exponential decay)

$$K = \{h : |h_i| \leq M\rho^i, M > 0, |\rho| < 1, i = 0, \dots, N-1\}.$$

- Y is the N dimensional space of measurements $y = [y_0, \dots, y_{N-1}]'$, whereas the noise vector $e \in Y$ is $e = v = [v_0, \dots, v_{N-1}]'$ which must satisfy $\|e\|_Y \leq \varepsilon$.
- The generic element in the solution space Z is obtained by truncating the sequence x after the first p elements

$$z = [h_0, \dots, h_{p-1}]' = T^p x$$

and so the solution operator is the truncation operator $S(\cdot) = T^p$.

Now it is possible to rewrite the input/output equations (1.24) as in (1.3), choosing the information operator

$$F(x) = UT^N x$$

where U is the Toeplitz lower triangle matrix composed by the input elements

$$U = \begin{pmatrix} u_0 & 0 & \dots & 0 \\ u_1 & u_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ u_{N-1} & u_{N-2} & \dots & u_0 \end{pmatrix}. \quad (1.25)$$

Note that in this case the information and solution operators are both linear.

Pointwise Estimators

In this chapter an overview of results concerning pointwise estimators is reported. Particular attention is devoted to the central and projection algorithms introduced in Section 1.3. Some basic properties of pointwise estimators are discussed. For a thorough treatment on such topics, the reader is referred to several papers present in the literature, such as [8, 9, 5].

The chapter is organized as follows. Section 2.1 describes linear estimation problems, along with their solution and related performances in terms of optimality, correctness, feasibility, etc., while in Section 2.2 asymptotic properties are addressed. Finally, in Section 2.3 some extensions of previous results to nonlinear problems are reported.

2.1 Linear problems

In this section, linear estimation problems are considered, that is problems where $F(x) = Fx$, $F \in \mathbb{R}^{m \times n}$, and $S(x) = Sx$, $S \in \mathbb{R}^{p \times n}$. For simplicity we assume $K = X$, considering that many results are also valid when K is

a convex and balanced set. First, let us analyze properties of the central algorithm.

2.1.1 Central algorithm

A useful result deals with the characterization of the diameter of information [3].

Lemma 2.1 *For every norm in the Y and Z spaces, one has*

$$D(\varepsilon) = 2 \sup_{x \in FPS_0} \|Sx\|_Z.$$

Proof.

$$\begin{aligned} D(\varepsilon) &= \sup_{y \in Y_0} \sup_{x_1, x_2 \in FPS_y} \|S(x_1 - x_2)\|_Z \leq \\ &\leq \sup_{x_1, x_2: \|Fx_1 - Fx_2\|_Y \leq 2\varepsilon} \|S(x_1 - x_2)\|_Z = \\ &= \sup_{x_1, x_2: \|Fx_1 - Fx_2\|_Y \leq \varepsilon} \|2S(x_1 - x_2)\|_Z = 2 \sup_{x: \|Fx\|_Y \leq \varepsilon} \|Sx\|_Z \end{aligned}$$

where the inequality derives from

$$\|Fx_1 - Fx_2\|_Y \leq \|Fx_1 - y\|_Y + \|Fx_2 - y\|_Y$$

which holds for every y , and therefore, in particular, for that where the sup in $D(\varepsilon)$ is achieved. On the other side, if $y = 0$, choosing $x_1, x_2 \in FPS_0$ such that $x_2 = -x_1$ one obtains

$$D(\varepsilon) \geq \sup_{x_1, x_2 \in FPS_0} \|S(x_1 - x_2)\|_Z \geq 2 \sup_{x_1 \in FPS_0} \|Sx_1\|_Z$$

and the lemma is proved. \square

Lemma 2.1 states that the maximum diameter of the set of feasible solutions (that is the worst error which is possible to achieve by choosing the estimate inside the set) is reached when all measurements are null.

The next theorem summarizes the optimal properties of central algorithms.

Theorem 2.1 *For every couple of norms $\|\cdot\|_Y$ and $\|\cdot\|_Z$, a central algorithm Φ_c is*

i) *correct;*

ii) *Y -locally optimal (and hence globally optimal)*

$$E_y(\Phi_c, \varepsilon) \leq E_y(\Phi, \varepsilon) \quad \forall y, \forall \Phi \quad (2.1)$$

with Y -local error $E_y(\Phi_c, \varepsilon) = \text{rad}(FSS_y)$;

iii) *X -locally optimal within a factor 2 among correct estimators*

$$E_x(\Phi_c, \varepsilon) \leq 2 E_x(\hat{\Phi}, \varepsilon) \quad \forall x, \forall \hat{\Phi} \text{ correct}. \quad (2.2)$$

Moreover, if $\|\cdot\|_Y$ and $\|\cdot\|_Z$ are such that $D(\varepsilon) = 2 R(\varepsilon)$, Φ_c is

iv) *X -locally optimal among correct algorithms, that is*

$$E_x(\Phi_c, \varepsilon) \leq E_x(\hat{\Phi}, \varepsilon) \quad \forall x, \forall \hat{\Phi} \text{ correct}. \quad (2.3)$$

Proof.

i) By definition (1.7), we want to prove that $\text{cen}(FSS_{F_x}) = Sx$, $\forall x \in X$.

Since

$$FSS_{F_x} = \{z = S\bar{x} : \|F(\bar{x} - x)\|_Y \leq \varepsilon\}$$

it follows that if $z_1 = S(x + \bar{x}) \in FSS_{F_x}$, then also $z_2 = S(x - \bar{x}) \in FSS_{F_x}$

and then Sx is the symmetry center of FSS_{F_x} , for every Y norm. Hence,

for every norm $\|\cdot\|_Z$ used to compute $\text{cen}(FSS_{F_x})$, it coincides with Sx .

ii) By Definitions 1.1, 1.4 and 1.11.

iii) For a correct estimator $\hat{\Phi}$ one has

$$\begin{aligned} E_x(\hat{\Phi}, \varepsilon) &= \sup_{y \in MUS_{F_x}} \|Sx - \hat{\Phi}(y)\|_Z && \geq \\ &\geq \sup_{\bar{x}: \|F(\bar{x} - x)\|_Y \leq \varepsilon} \|Sx - \hat{\Phi}(F\bar{x})\|_Z && = \\ &= \sup_{\bar{x}: \|F(\bar{x} - x)\|_Y \leq \varepsilon} \|S(x - \bar{x})\|_Z && = \frac{D(\varepsilon)}{2} \quad \forall x \end{aligned} \quad (2.4)$$

when the last equality derives by Lemma 2.1. On the other side, from the global optimality of Φ_c and from (1.17) one has

$$E_x(\Phi_c, \varepsilon) \leq E(\Phi_c, \varepsilon) = \inf_{\Phi} E(\Phi, \varepsilon) = R(\varepsilon). \quad (2.5)$$

From (1.19) it follows that $R(\varepsilon) \leq D(\varepsilon)$, and so (2.4)–(2.5) implies (2.2).

iv) If $D(\varepsilon) = 2R(\varepsilon)$, from (2.4)–(2.5) one obtains (2.3). \square

It is worthwhile to note that the condition $D(\varepsilon) = 2R(\varepsilon)$ is verified in many cases of interest. In particular it holds if:

- $\|\cdot\|_Z = \ell_\infty$, whatever $\|\cdot\|_Y$;
- $\|\cdot\|_Y = \ell_2$, whatever $\|\cdot\|_Z$ (or, more in general, for every Y -norm such that FPS_y has a symmetry center);
- $p = 1$ (scalar solution), for every norm in Y and Z [10].

Theorem 2.1 shows that in a generic set-membership estimation problem, the central algorithm is the best pointwise estimator. Note that the central algorithm may not be unique.

In general a central algorithm is not linear and can be hard to compute, especially if FSS_y does not have a symmetry center. In the case when $\|\cdot\|_Z = \ell_\infty$, FSS_y is a polytope (see Table 1.1) and the following result holds [11].

Theorem 2.2 *Let $\|\cdot\|_Z = \ell_\infty$. Then the central algorithm Φ_c can be evaluated in the following way:*

$$\Phi_{c,i}(y) = cen_i(FSS_y) = \frac{\overline{z_i} + \underline{z_i}}{2} \quad i = 1, \dots, p \quad (2.6)$$

where

$$\begin{aligned} \overline{z_i} &= \sup_{z \in FSS_y} z_i = \sup_{x \in FPS_y} S_{i,\cdot} x & i = 1, \dots, p \\ \underline{z_i} &= \inf_{z \in FSS_y} z_i = \inf_{x \in FPS_y} S_{i,\cdot} x & i = 1, \dots, p \end{aligned} \quad (2.7)$$

and $S_{i,\cdot}$ is the i -th row of the matrix S . The Y -local error is

$$\text{rad}(FSS_y) = \max_{i=1,\dots,p} \frac{\bar{z}_i - \underline{z}_i}{2}.$$

From Theorem 2.2 it follows that a central algorithm Φ_c and its Y -local error $E_y(\Phi_c, \varepsilon)$ can be obtained by solving the $2p$ linear programming problems in (2.7).

2.1.2 Linear algorithms

In many applications, the computational burden required by a central algorithm can be unacceptable; in this case it is necessary to use suboptimal simpler algorithms. A possible choice is to force the estimation operator to be linear, i.e. $\Phi(y) = \Phi y$, $\Phi \in \mathbb{R}^{p \times m}$.

The next theorem shows that, if it is used the ℓ_∞ norm in the Y and Z spaces, it is always possible to design a correct linear algorithm, globally optimal and X -locally optimal among correct estimators [12, 11].

Theorem 2.3 *Let $\|\cdot\|_Z = \ell_\infty$, $\|\cdot\|_Y = \ell_\infty$ and $m > n$. Then there exists one linear estimator Φ_l , such that*

1. Φ_l is correct;
2. Φ_l is globally optimal

$$E(\Phi_l, \varepsilon) \leq E(\Phi, \varepsilon) \quad \forall \Phi; \quad (2.8)$$

3. Φ_l is X -locally optimal among correct estimators

$$E_x(\Phi_l, \varepsilon) \leq E_x(\Phi, \varepsilon) \quad \forall x, \forall \Phi \text{ correct}; \quad (2.9)$$

4. the global error and the Y -local error of Φ_l coincide and are equal to

$$E(\Phi_l, \varepsilon) = E_x(\Phi_l, \varepsilon) = \text{rad}(FSS_{F_x}) \quad \forall x. \quad (2.10)$$

The algorithm Φ_l described in Theorem 2.3 can be computed from the active constraints of the linear programming problems (2.7) with $y = 0$ (for further details see [12, 11]).

We will see in the following that if we use the ℓ_2 norm in the Y space, there exists a linear algorithm that enjoys the global optimality and X -local optimality properties (among correct algorithms) described in Theorem 2.3 and it is also a central algorithm and therefore Y -locally optimal (see Theorem 2.1). This algorithm is the celebrated *least squares algorithm*, and it belongs to the class of projection algorithms representing a very special case inside that class. The general properties of projection algorithm are discussed in the next section.

2.1.3 Projection algorithms

The first property of projection algorithms is a direct consequence of Definition 1.12.

Theorem 2.4 *A projection algorithm is always an interpolatory algorithm.*

Proof. Let Φ_p be a projection algorithm, such that $\Phi_p(y) = Sx_p$, with x_p defined in (1.23). If $y \in Y_0$ then

$$\|y - F(x_p)\|_Y \leq \|y - F(\bar{x})\|_Y \leq \varepsilon \quad \forall \bar{x} \in FPS_y$$

and so $x_p \in FPS_y$ and Φ_p is interpolatory. \square

An immediate consequence of the previous theorem is that, thanks to Proposition 1.4, projection algorithms are Y -locally almost-optimal within a factor 2. This is due to the fact that in the definition of projection algorithm and in that of FPS_y it has been used the same Y norm. In the following it will be shown that, if the projection is computed using a norm which differs from that used to evaluate the UBB noise, Theorem 2.4 does not hold anymore, and projection algorithms no more enjoy this suboptimality property.

An important feature of projection algorithms is their robustness against uncertainties about the noise level ε . The following result holds [3].

Theorem 2.5 *Let Φ_p be the projection algorithm shown in Definition 1.12. It follows that*

$$E_y(\Phi_p, \varepsilon) \leq 2 \operatorname{rad}(FSS_y) \leq 2 E_y(\Phi_c, \varepsilon) \quad \forall y, \forall \Phi, \forall \varepsilon. \quad (2.11)$$

Property (2.11) is called *robust almost-optimality* (Y -local). It is very important since it is usually difficult to know exactly the value of ε , while it is quite easier to estimate an upper bound. For this reason it is useful to guarantee the performance level of the estimation algorithm also when the noise level is overestimated. It is worthwhile to note that central algorithms do not enjoy this robustness property; a central algorithm computed for $\varepsilon = \varepsilon_0$ is, in general, no longer optimal if the real value of ε is less than ε_0 . Moreover, the central estimate Φ_c may not belong to the actual feasible set FSS_y and the Y -local error $E_y(\Phi_c, \varepsilon)$ can be larger than $2 \operatorname{rad}(FSS_y)$.

Performances of projection algorithms may improve if the set of feasible elements FPS_y has a special structure. In particular, if it has a symmetry center, it can be proved that the projection algorithm is optimal in every sense [13].

Theorem 2.6 *Let $m \geq n$. For every norm $\|\cdot\|_Y$ for which FPS_y has a symmetry center, the projection algorithm Φ_p is*

- linear and correct;
- central;
- Y -locally optimal (robustly);
- X -locally optimal (among correct estimators).

A special class of norms for which FPS_y has a central symmetry, is that of Hilbert norms [13].

Proposition 2.1 *If Y is a Hilbert space, then x_y defined by*

$$x_y = \arg \min_{x \in X} \|Fx - y\|_Y$$

is the symmetry center of FPS_y , that is $\forall y \in Y_0$ one has

$$x \in FPS_y \implies (2x_y - x) \in FPS_y.$$

Note that in the context defined by Proposition 2.1, the projection algorithm is defined by $\Phi_p(y) = Sx_y$. In particular, choosing $\|\cdot\|_Y = \ell_2$, the projection algorithm is the celebrated least squares algorithm Φ_{LS} . In the following its properties, which derive directly from Theorem 2.6, will be described;

Corollary 2.1 *Let $m \geq n$. The least squares algorithm Φ_{LS} is correct, linear, central, Y -locally optimal (robustly with respect to ε) and X -locally optimal (among correct algorithms). Moreover, if $\text{rank}(F) = n$, one has*

$$\Phi_{LS}(y) = S(F'F)^{-1}F'y. \quad (2.12)$$

As previously stressed in Section 1.3, in real applications projection algorithms are commonly used also in non-Hilbert spaces (ℓ_∞ , ℓ_1 , etc.). For this reason, it should be useful to extend optimality properties of Theorem 2.6 to situations where FPS_y does not have a symmetry center, as it happens if ℓ_∞ or ℓ_1 norms in the Y space are used. Unfortunately, in these cases it can only be shown that the projection algorithm is not optimal, neither X - nor Y - nor globally.

In the following an example which shows the difference among projection algorithms in ℓ_2 , ℓ_∞ and ℓ_1 norms is reported. For a complete overview see [13].

Example 2.1 Let $X = \mathbb{R}$, $Y = \mathbb{R}^2$. We want to estimate the parameter $x \in \mathbb{R}$ on the basis of measures of the function

$$F(x) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} x$$

corrupted by additive noise e , bounded in y norm, $\|e\|_Y \leq \varepsilon$. So $Z = X = \mathbb{R}$, $S(x) = x$, and the measures $y = [y_1 \ y_2]'$ are given by

$$\begin{aligned} y_1 &= x + e_1 \\ y_2 &= 2x + e_2. \end{aligned}$$

Let us consider the following three cases

$$i) \quad \|\cdot\|_Y = \ell_2, \quad e_1^2 + e_2^2 \leq \varepsilon^2$$

$$ii) \quad \|\cdot\|_Y = \ell_\infty, \quad |e_i| \leq \varepsilon \quad i = 1, 2$$

$$iii) \quad \|\cdot\|_Y = \ell_1, \quad |e_1| + |e_2| \leq \varepsilon$$

and compare performances of central algorithms (Φ_c) and projection ones (Φ_p).

A geometric description of the three situations is reported in Figures 2.1, 2.2, 2.3. Note that in case i) the projection algorithm (i.e. the least square algorithm) is also central, as stated in Corollary 2.1, and hence Y -locally optimal. On the contrary, in cases ii) and iii) the projection algorithm and the central algorithm give different estimates. Since in all considered situations the feasible set FPS_y is a segment of the real axis, the central estimate $\Phi_c(y)$ is unique and so the projection algorithm can not be Y -locally optimal.

Regarding the global optimality, the global errors of the algorithms Φ_c and Φ_p are reported in Table 2.1; for further details see [13]. Note that they coincide only in the case i).

Being Φ_c globally optimal (see Theorem 2.1) it is possible to state that if we use the ℓ_∞ or ℓ_1 norms on the Y space, the projection algorithm is not globally optimal.

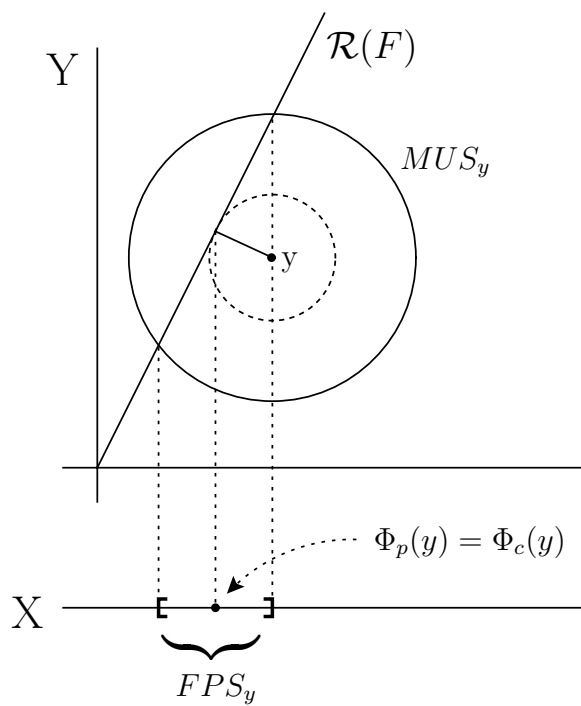


Figure 2.1: Example 2.1: case $\|\cdot\|_Y = \ell_2$.

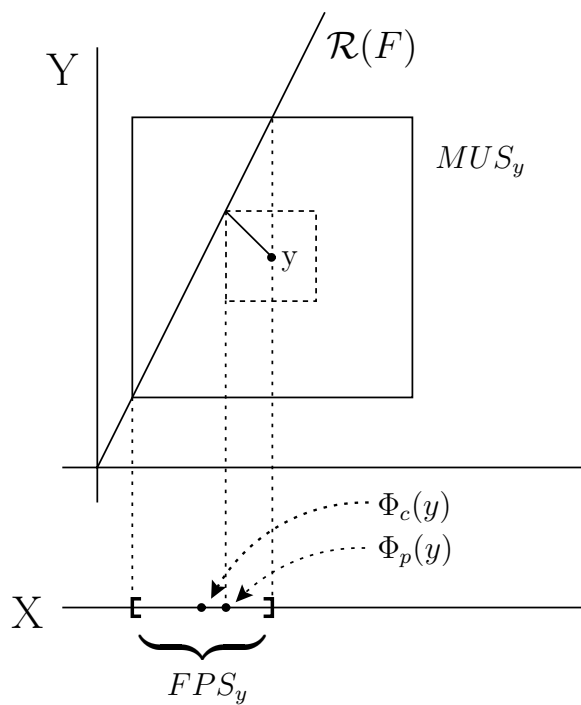
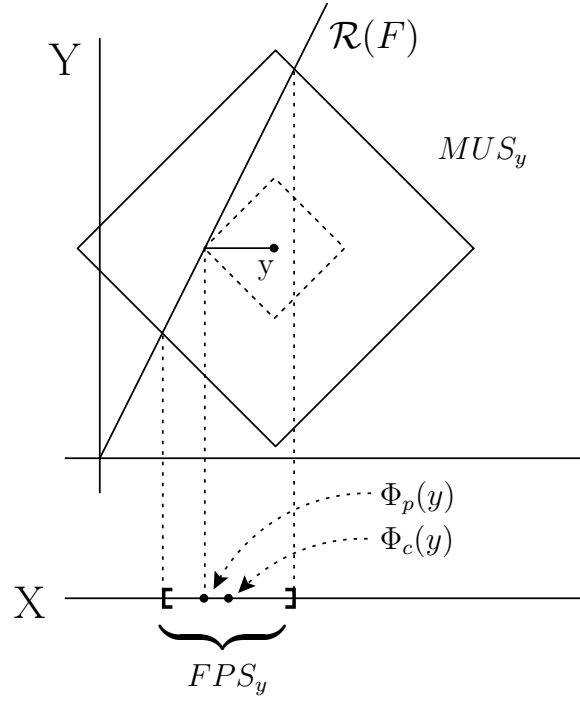


Figure 2.2: Example 2.1: case $\|\cdot\|_Y = \ell_\infty$.

Figure 2.3: Example 2.1: case $\|\cdot\|_Y = \ell_1$.

$\ \cdot\ _Y$	$E(\Phi_c, \varepsilon)$	$E(\Phi_p, \varepsilon)$
ℓ_2	$\frac{1}{\sqrt{5}}\varepsilon$	$\frac{1}{\sqrt{5}}\varepsilon$
ℓ_∞	$\frac{1}{2}\varepsilon$	$\frac{2}{3}\varepsilon$
ℓ_1	$\frac{1}{3}\varepsilon$	$\frac{1}{2}\varepsilon$

Table 2.1: Global error for different Y norms.

Since the least squares algorithm is optimal (in all senses) if the noise is energy bounded (ℓ_2), the properties of this algorithm have been investigated in the case of errors bounded in ℓ_∞ norm. It is easy to show that, in this case, being the projection performed in a different norm compared with that used to define FPS_y , it is not possible to guarantee that the estimate $\Phi_{LS}(y)$ belongs to the feasible set FSS_y . Thus, in general, the least squares algorithm is not optimal (in any sense) in presence of errors bounded in ℓ_∞ norm. However, for special structures of the information operator F , the algorithm Φ_{LS} can enjoy optimality properties. In particular, in [4] it has been shown that to identify the parameters of a FIR filter, or the impulse response of a linear system, the least squares algorithm is globally optimal also in the case of noise bounded in ℓ_∞ norm, if the input sequence is an impulse or a step, and the norm used in the X space is ℓ_∞ or ℓ_1 .

2.2 Asymptotic properties of algorithms

In Section 1.2.2 the concepts of asymptotic and robust convergence of an estimation algorithm have been introduced. Moreover, it has been shown that the analysis of asymptotic properties of optimal and almost-optimal estimators boils down to convergence analysis of the diameter of information $D(\varepsilon)$. Now, let us describe some results concerning the linear setting considered in the previous section.

Let us examine the case where the dimension n of the element space is finite and fixed. Let us denote by $F^m \in \mathbb{R}^{m \times n}$ the information operator matrix, by FPS_y^m the feasible parameter set and by $D(\varepsilon, m)$ the corresponding diameter of information, to make explicit the dependance on the number m of available measurements. The following lemma is a general characterization of the asymptotic behaviour of the diameter of information [4].

Lemma 2.2 *If there exists m_0 such that $\text{rank}(F^{m_0}) = n$, then*

$$\lim_{\varepsilon \rightarrow 0} \lim_{m \rightarrow \infty} D(\varepsilon, m) = 0. \quad (2.13)$$

Proof. In Section 1.1.2 it has been reported that FPS_y^m is bounded if and only if $\text{rank}(F^m) = n$. Since this holds for every y , it follows that $FPS_0^{m_0}$ is bounded. Moreover, it is straightforward that, adding new measures, the feasible set becomes smaller, i.e.

$$FPS_0^{m+1} \subseteq FPS_0^m \quad \forall m$$

and so FPS_0^m is bounded for all $m \geq m_0$. From Lemma 2.1 it follows that

$$D(\varepsilon, m) \leq D(\varepsilon, m_0) \quad \forall m \geq m_0$$

and moreover

$$\lim_{\varepsilon \rightarrow 0} D(\varepsilon, m) = 0 \quad \forall m \geq m_0$$

from which (2.13) follows. \square

An important consequence of Lemma 2.2 is the possibility to guarantee the robust convergence of all optimal and almost-optimal algorithms.

Theorem 2.7 *Let $\text{rank}(F^{m_0}) = n$, for some m_0 . Then, any globally optimal or almost-optimal algorithm is robustly convergent.*

Proof. Let Φ_o and Φ_{ao} be two algorithms, globally optimal and almost-optimal respectively. From the optimality definition it follows that

$$\begin{aligned} E(\Phi_o, \varepsilon) &= R(\varepsilon) \\ E(\Phi_{ao}, \varepsilon) &\leq k R(\varepsilon). \end{aligned}$$

Since by (1.19) $R(\varepsilon) \leq D(\varepsilon)$, the proof follows immediately from Lemma 2.2.

\square

From the previous theorem, robust convergence of some important classes of estimators follows.

Corollary 2.2 *If the hypothesis of Theorem 2.7 holds, then all algorithms described in the following are robustly convergent:*

- *central algorithms;*
- *projection algorithms;*
- *interpolatory algorithms.*

Note that the asymptotic results previously stated are valid for every norm in the X , Y and Z spaces. Thus, it is possible to conclude that, if the space of the problem Elements has finite dimension, the hypothesis of Lemma 2.2 guarantees the convergence of all estimation algorithms considered so far.

It is worthwhile to emphasize that projection algorithms are convergent if the projection is done in the same Y norm appearing in the UBB hypothesis. On the contrary, there exist examples where the least squares algorithm diverges when $\|\cdot\|_Y = \ell_\infty$ (see [4]).

2.3 Properties in the nonlinear case

Many results reported in the previous paragraphs are related to the linearity of the information and solution operators. Nevertheless, it is possible to extend some properties of central and projection algorithms to the general case.

Theorem 2.8 *A central algorithm Φ_c is Y -locally optimal. The minimum Y -local error turns out to be*

$$E_y(\Phi_c, \varepsilon) = \text{rad}(FSS_y).$$

Note that Theorem 2.8 holds for every X, Y, Z norm.

Let us analyze how to compute a central algorithm when $\|\cdot\|_Y = \ell_\infty$ in the nonlinear case [11].

Theorem 2.9 *Let $\|\cdot\|_Z = \ell_\infty$. Then the central algorithm Φ_c can be computed in the following way:*

$$\Phi_c(y) = \text{cen}(FSS_y) = \frac{\overline{z}_i + \underline{z}_i}{2} \quad i = 1, \dots, p \quad (2.14)$$

where

$$\begin{aligned} \overline{z}_i &= \sup_{x \in FPS_y} S_i(x) & i = 1, \dots, p \\ \underline{z}_i &= \inf_{x \in FPS_y} S_i(x) & i = 1, \dots, p \end{aligned} \quad (2.15)$$

and S_i is the i -th component of the operator S . The Y -local error is given by

$$\text{rad}(FSS_y) = \max_{i=1, \dots, p} \frac{\overline{z}_i - \underline{z}_i}{2}.$$

Thus the computation of the central algorithm is equivalent to the resolution of the $2p$ optimization problems (2.15). In the nonlinear case, these problems are in general non convex. Global optimization tools provide only approximated solutions for problems (2.15), and are not able to evaluate the approximation error. However, if $F(\cdot)$ and $S(\cdot)$ are polynomial functions, there exist special optimization algorithms that can provide better results [14].

Theorem 2.10 *If $F(x)$ and $S(x)$ are polynomial functions, there exist algorithms which converge (with probability 1) to the global extremes of problems (2.15).*

Under the hypotheses of Theorem 2.10, (2.15) are problems of *polynomial optimization*. In general these kinds of problems are non convex and can present local solutions [15]. But if problem variables are strictly positive, we have the so-called *signomial optimization* problems, for whom convergent (in probability) algorithms are provided [16, 17, 14]. If the sign of some problem variables (i.e. some components of x) is unknown, it is possible to reconduct

a polynomial problem to a signomial one by replacing these variables with the difference of two auxiliary variables strictly positive.

Concerning projection algorithms, they also have in the general case two important features: the feasibility of the provided estimation (they are interpolatory algorithms) and the Y -local and global robust almost-optimality. This means that results stated in Theorem 2.4 and 2.10 hold, as well as the other considerations about robustness of projection algorithms. This last property, along with the relative easiness to compute such algorithms, is the reason why projection algorithms are often preferred to the optimal ones.

Conditional estimation

In estimation problems, a typical goal is to find an estimate within a subset of the solution space. This can happen for several reasons; a typical example concerns system identification, where it is convenient to find a solution within a set of models of reduced order. This is particularly useful when the estimated model is used to design a controller.

In other cases, the restriction on the solution set is related to the problem nature, as for instance in set-membership filtering, where the estimate of the state sequence at one time instant is constrained by the previously computed estimates.

In this chapter the so-called *reduced-complexity estimation* (or *conditional estimation*) is introduced in the context of the theory of optimal algorithms (Section 3.1). In Section 3.2 some conditional pointwise estimators along with a description of their general properties are reported. A specific treatment concerning the case of energy bounded noise is reported in Section 3.3. Finally, a detailed characterization about properties of the main conditional algorithms is provided in Section 3.4.

3.1 Problem formulation

Let us consider the generic problem described in Section 1.1, under the linear hypotheses introduced in Section 2.1. Thus, the relation between the unknown element $x \in \mathbb{R}^n$ and the noisy measurements $y \in \mathbb{R}^m$ is

$$y = Fx + e$$

under the UBB noise hypothesis

$$\|e\|_Y \leq \varepsilon.$$

In the following, to simplify the treatment, we suppose that the quantity we want to estimate is x , and so $S = I$ (the parameter set and the solution set coincide). From this assumption, it follows that the set of the feasible estimates compatible with the available information is FPS_y whose definition has been given in (1.7). Moreover, let us assume that there are no a priori hypotheses on the unknown element, and so $K = X$.

In previous chapters, the aim of the estimation problem was to find an algorithm $\Phi(\cdot)$ such that $z = \Phi(y) \in X$ is a good approximation of the unknown element x . On the contrary, now we want to compute an estimate which belongs to a previously defined set. This constraint can be summarized as follows

$$z \in \mathcal{M}$$

where \mathcal{M} is a subset of X ($\mathcal{M} \subset X$). The aim is to find a *reduced-complexity estimator* (or *conditional estimator*) $\Phi : Y \rightarrow \mathcal{M}$ such that $z = \Phi(y) \in \mathcal{M}$ is a good estimate of the unknown element x . We will denote the class of conditional estimators as $\mathcal{A}_{\mathcal{M}}$, to emphasize the dependence on the set \mathcal{M} .

The quality of an estimation algorithm $\Phi \in \mathcal{A}_{\mathcal{M}}$ is still evaluated by the Y -local error

$$E_y(\Phi, \varepsilon) = \sup_{x \in FPS_y} \|x - \Phi(y)\|_X$$

that is the maximum distance of the estimate $\Phi(y) \in \mathcal{M}$ from an element of the feasible set. Of course, the minimum error (local or global) that can be obtained changes for reduced-complexity algorithms. We can define the *conditional radius of information* as

$$R_{\mathcal{M}}(\varepsilon) = \inf_{\Phi \in \mathcal{A}_{\mathcal{M}}} E(\Phi, \varepsilon).$$

From (1.17) it follows immediately that

$$R_{\mathcal{M}}(\varepsilon) \geq R(\varepsilon) \quad \forall \mathcal{M}.$$

It is straightforward that the conditional radius of information depends on the choice of the set \mathcal{M} . In the following we will consider affinely parameterized sets \mathcal{M} , such as

$$\mathcal{M} = \{z \in \mathbb{R}^n : z = z^o + M\alpha, \quad \alpha \in \mathbb{R}^h\} \quad (3.1)$$

with $h < n$. In other words, the admissible sets \mathcal{M} are h -dimensional linear manifolds, contained in the space \mathbb{R}^n .

3.2 Conditional pointwise estimators

In this section, three useful algorithms commonly used for reduced-complexity estimation are introduced. They are defined as follows.

Definition 3.1 *The conditional central algorithm is the operator*

$$\Phi_{cc}(y) = z_{cc}$$

where

$$z_{cc} = \arg \inf_{z \in \mathcal{M}} \sup_{x \in FPS_y} \|z - x\|_X \triangleq cen_{\mathcal{M}}(FPS_y). \quad (3.2)$$

Definition 3.2 *An interpolatory projection algorithm is an operator*

$$\Phi_{ip}(y) = z_{ip}$$

where

$$z_{ip} = \arg \inf_{z \in \mathcal{M}} \|z - x_i\|_X \quad (3.3)$$

and $x_i \in FPS_y$.

Definition 3.3 *The central projection algorithm is the operator*

$$\Phi_{cp}(y) = z_{cp}$$

where

$$z_{cp} = \arg \inf_{z \in \mathcal{M}} \|z - z_y\|_X \quad (3.4)$$

and

$$z_y = cen(FPS_y) = \arg \inf_{z \in \mathbb{R}^n} \sup_{\tilde{z} \in FPS_y} \|z - \tilde{z}\|_X. \quad (3.5)$$

Definition 3.4 *The restricted projection algorithm is the operator*

$$\Phi_{rp}(y) = z_{rp}$$

where

$$z_{rp} = \arg \inf_{z \in \mathcal{M}} \|Fz - y\|_Y. \quad (3.6)$$

Remark 3.1 *The previously defined algorithms are often cited in the literature in various ways. For example in [18], when $\|\cdot\|_Y$ is a Hilbert norm, Φ_{cp} is called conditional least squares algorithm, whereas Φ_{rp} is called reduced least squares algorithm.*

It is useful to provide a geometric interpretation of conditional estimators. The algorithm Φ_{cc} computes the Chebyshev center of FPS_y “conditioned” to the set \mathcal{M} , that is the element in \mathcal{M} where it is centered the minimum radius sphere (in X norm) which contains FPS_y . Conversely, the estimate provided by Φ_{cp} is the projection (in X norm) of the Chebyshev center of FPS_y on the manifold \mathcal{M} , that is the element of \mathcal{M} which has the minimum distance

from $cen(FPS_y)$. Finally, the algorithm Φ_{rp} computes the projection (in Y norm) of the measurement vector y on the set $F(\mathcal{M})$. Note that if $K = \mathcal{M}$, the restricted projection algorithm coincides with the non restricted one (see Definition 1.12).

In reduced-complexity estimation, the conditional central algorithm plays the same role as the central algorithm in non conditional estimation problems.

Theorem 3.1 *The conditional central algorithm Φ_{cc} is Y -locally optimal (and so globally optimal) among the reduced orders estimators*

$$E_y(\Phi_{cc}, \varepsilon) \leq E_y(\Phi, \varepsilon) \quad \forall y, \forall \Phi \in \mathcal{A}_{\mathcal{M}}.$$

Thus the conditional central algorithm minimizes the estimation error among all algorithms whose estimates are constrained to belong to \mathcal{M} .

The central projection algorithm is not optimal; however it is possible to provide bounds on its worst-case estimation error.

Theorem 3.2 *The central projection algorithm is optimal within a factor 3, that is*

$$E_y(\Phi_{cp}, \varepsilon) \leq 3 E_y(\Phi, \varepsilon) \quad \forall y, \forall \Phi \in \mathcal{A}_{\mathcal{M}}.$$

Proof. From (3.4) one has, for all $z \in FPS_y$

$$\begin{aligned} \|z_{cp} - z\|_X &\leq \|z_{cp} - z_y\|_X + \|z_y - z\|_X \leq \\ &\leq \|z_{cc} - z_y\|_X + \|z_y - z\|_X. \end{aligned}$$

Maximizing with respect to $z \in FPS_y$, one obtains

$$E_y(\Phi_{cp}, \varepsilon) \leq \|z_{cc} - z_y\|_X + \sup_{z \in FPS_y} \|z_y - z\|_X. \quad (3.7)$$

On the other hand, using again the triangle inequality, one has

$$\|z_{cc} - z_y\|_X \leq \|z_{cc} - z\|_X + \|z - z_y\|_X \quad \forall y$$

and thus

$$E_y(\Phi_{cp}, \varepsilon) \leq \sup_{z \in FPS_y} \|z_{cc} - z\|_X + 2 \sup_{z \in FPS_y} \|z_y - z\|_X.$$

From (3.5), it follows that

$$E_y(\Phi_{cp}, \varepsilon) \leq 3 E_y(\Phi_{cc}, \varepsilon)$$

and, thanks to Theorem 3.1, the proof is completed. \square

This optimality factor can be improved depending on the structure of the FPS_y . In particular, we have the following result.

Corollary 3.1 *If $z_y \in FPS_y$, the central projection algorithm is optimal within a factor 2.*

Proof. Since $z_y \in FPS_y$, one has

$$\|z_{cc} - z_y\|_X \leq \sup_{z \in FPS_y} \|z_{cc} - z\|_X.$$

The proof follows by (3.7) in the proof of Theorem 3.2. \square

Note that the preceding results are general. They do not depend either on the set \mathcal{M} , or on the norm choice $\|\cdot\|_X$, $\|\cdot\|_Y$. In some specific cases it is possible to provide more detailed information about conditional algorithm performances. A typical example is reported in the following [18].

Theorem 3.3 *Let $\|\cdot\|_X = \ell_\infty$ and $\|\cdot\|_Y = \ell_2$. Let \mathcal{M} be the linear manifold defined in (3.1), such that $M = [I_h \ 0]'$ (that is \mathcal{M} is a hyperplane parallel to h cartesian axes). Then Φ_{cp} is Y -locally optimal among the reduced-complexity algorithms*

$$E_y(\Phi_{cp}, \varepsilon) \leq E_y(\Phi, \varepsilon) \quad \forall y, \forall \Phi \in \mathcal{A}_{\mathcal{M}}.$$

Proof. See [18]. \square

Let us now analyze interpolatory projection algorithms.

Theorem 3.4 *Any interpolatory projection algorithm is optimal within a factor 3, that is*

$$E_y(\Phi_{ip}, \varepsilon) \leq 3 E_y(\Phi, \varepsilon) \quad \forall y, \forall \Phi \in \mathcal{A}_{\mathcal{M}}.$$

Proof. Let $z \in FPS_y$ and $z_m \in \mathcal{M}$. From (3.3), one has

$$\begin{aligned} \|z_{ip} - z\|_X &\leq \|z_{ip} - x_i\|_X + \|x_i - z\|_X \leq \\ &\leq \|z_m - x_i\|_X + D(\varepsilon) \leq \sup_{x \in FPS_y} \|z_m - x\| + 2R(\varepsilon). \end{aligned}$$

The result is obtained by maximizing over $z \in FPS_y$ and minimizing over $z_m \in \mathcal{M}$. \square

In the following, it will be reported a more general case with respect to that considered in Theorem 3.3, where it will be possible to characterize in details the algorithms Φ_{cc} , Φ_{cp} , Φ_{rp} and their estimation errors.

3.3 Conditional estimation with energy bounded noise

Let us consider the reduced estimation problem described in Section 3.1 along with the set \mathcal{M} defined as a linear manifold (3.1). Moreover let us assume that noise is bounded in ℓ_2 norm, that is

$$\|e\|_2 \leq \varepsilon.$$

First of all it is useful to highlight some geometric features of the problem.

- The feasible set is the ellipsoid

$$FPS_y = \{x \in \mathbb{R}^n : x' F' F x - 2y' F x + y'y \leq \varepsilon^2\}.$$

- z_{cc} in (3.2) is the Chebyshev center of the ellipsoid FPS_y , constrained to the h -dimensional hyperplane \mathcal{M} .

- z_y in (3.5) is the center of the ellipsoid FPS_y

$$z_y = (F'F)^{-1}F'y.$$

- z_{cp} in (3.4) is the projection of z_y on \mathcal{M} , in norm $\|\cdot\|_X$. If we assume that $\|\cdot\|_X = \ell_2$, one has

$$z_{cp} = z^o - M(M'M)^{-1}M'(F'F)^{-1}F'(y - Fz^o).$$

- z_{rp} in (3.6) is the counterimage, through F , of the orthogonal projection of y on the linear manifold $F(\mathcal{M})$ in \mathbb{R}^m , that is

$$z_{rp} = z^o + M(M'F'FM)^{-1}M'F'(y - Fz^o).$$

By means of a suitable change of coordinates in the X space, the problem can be stated, without loss of generality, in the following way. Let the feasible set FPS_y be the ellipsoid

$$\mathcal{E} = \{x \in \mathbb{R}^n : x'Qx \leq 1\} \quad (3.8)$$

$$Q = \text{diag}\{q_i\}_{i=1}^n, \quad 0 < q_1 \leq q_2 \leq \dots \leq q_n \quad (3.9)$$

and let the linear manifold \mathcal{M}

$$\mathcal{M} = \{z \in \mathbb{R}^n : z = z^o + M\alpha, \alpha \in \mathbb{R}^h\} \quad (3.10)$$

$$M'M = I_h \quad (3.11)$$

$$M'z^o = 0. \quad (3.12)$$

with $h < n$. Thus we have $z_y = 0$ and, if $\|\cdot\|_X = \ell_2$, the estimates provided by the three previously considered algorithms become respectively

$$z_{cc} = \arg \min_{z \in \mathcal{M}} \max_{x \in \mathcal{E}} \|x - z\|_2 \quad (3.13)$$

$$z_{cp} = z^o \quad (3.14)$$

$$z_{rp} = [I_n - M(M'QM)^{-1}M'Q]z^o. \quad (3.15)$$

Note that in general the computation of the conditional center is a very complex problem. However, under the previous hypotheses (energy bounded disturbances and errors measured in ℓ_2 norm), an efficient algorithm has been provided to solve such a problem [19].

3.3.1 \mathcal{H}_2 set-membership identification

In this section, an example of conditional set-membership identification problem is provided.

Let us consider the problem described in Section 1.5. We assume that the noise sequence is energy bounded in ℓ_2 norm, and that there are no a priori information, that is $K = X$.

The aim is to estimate the truncated impulse response x on the base of N input/output measures.

The feasible set is now given by

$$FPS_y^N = \{x : \|UT^N x - y\|_2 \leq \varepsilon\} \quad (3.16)$$

where U is the Toeplitz matrix in (1.25).

Of course, since $y \in \mathbb{R}^N$ it is possible to estimate only the first N samples of the impulse response, $x^N = T^N x$.

For many reasons, it can be useful to estimate x^N inside a linearly parameterized model class. This means that we want to find an algorithm $\Phi : Y \rightarrow \mathcal{M}$ providing a reduced order model $z \in \mathcal{M}$, where \mathcal{M} is a linear manifold in the infinite dimensional space of impulse responses. In system identification theory, the basis \mathcal{M} is usually a set of impulse responses of linear filters, such as Laguerre filters, Kautz filters and other orthonormal functions [20, 21]. A deeper treatment about such topics is reported in Chapter 4.

When the estimation error is measured in ℓ_2 norm (as in \mathcal{H}_2 identification¹), it can be shown that the basis elements can be impulse responses of FIR filters of length N , obtained by truncating the impulse responses of the basis after N time steps ([22], see also Section 4.4). This means that the choice of the estimate of x^N can be restricted to the set

$$\mathcal{M} = \{x^N \in \mathbb{R}^N : x^N = \bar{x}^N + M\alpha, \alpha \in \mathbb{R}^h\}$$

where \bar{x}^N and the columns of $M \in \mathbb{R}^{N \times h}$ are the basis of the class \mathcal{M} of order $h < N$. It follows that the algorithms described in Definitions 3.1–3.4 provide possible solutions of the \mathcal{H}_2 set-membership identification problem. In particular, by Theorem 3.1, it follows that the conditional central estimation $\Phi_{cc}(y)$ provides the optimal impulse response (i.e. the minimum Y -local error) among the reduced order models \mathcal{M} .

3.4 Properties of conditional algorithms

In Section 3.2 the central projection algorithm and the restricted projection algorithm have been introduced (Definitions 3.3 and 3.4) for the generic problem of restricted complexity estimation. As described in Section 3.3, if the noise is energy bounded it follows that the estimates provided by these algorithms can be easily computed by means of (3.14) and (3.15). Since these algorithms provide a computationally efficient alternative to the optimal algorithm Φ_{cc} , it is useful to investigate their properties, and in particular, their degree of optimality.

Regarding the central projection algorithm, the following result is an immediate consequence of the problem geometry [22].

Proposition 3.1 *If $\|\cdot\|_Y = \ell_2$, the central projection algorithm Φ_{cp} is optimal within a factor 2.*

¹ \mathcal{H}_2 is the set of real functions that are square integrable on the unit circle.

Proof. Being FPS_y an ellipsoid, z_y in (3.5) is its symmetry center. Thus $z_y \in FPS_y$ and the result follows immediately by Corollary 3.1. \square

This bound can be improved under the hypothesis that also the X norm is ℓ_2 as stated in the following theorem.

Theorem 3.5 *Let \mathcal{M} be a full-rank linear manifold as in (3.1) and let $\|\cdot\|_X = \|\cdot\|_Y = \ell_2$. Then the central projection algorithm Φ_{cp} satisfies*

$$E_y(\Phi_{cp}, \varepsilon) \leq \sqrt{\frac{4}{3}} E_y(\Phi, \varepsilon) \quad \forall y, \forall \Phi \in \mathcal{A}_{\mathcal{M}}.$$

Proof. See [23]. \square

Indeed, the previous theorem states that under hypotheses of energy bounded noise and estimation errors measured in energy norm, the central projection algorithm is optimal within a factor $\sqrt{4/3}$. Moreover, it has been shown that there exist estimation problems for which the upper bound is achieved.

Let us examine a generic interpolatory projection algorithm.

Theorem 3.6 *Under the hypotheses of Theorem 3.5, an interpolatory projection algorithm Φ_{ip} is optimal within a factor 2, that is*

$$E_y(\Phi_{ip}, \varepsilon) \leq 2 E_y(\Phi, \varepsilon) \quad \forall y, \forall \Phi \in \mathcal{A}_{\mathcal{M}}.$$

Proof. See [23]. \square

Let us consider the restricted projection algorithm. If the set \mathcal{M} is a linear manifold, the estimate z_{rp} assumes the following geometric meaning.

Proposition 3.2 *Let \mathcal{E} and \mathcal{M} be defined respectively as in (3.8) and (3.10). If $\mathcal{E} \cap \mathcal{M} \neq \emptyset$, then z_{rp} in (3.15) is the symmetry center of $\mathcal{E} \cap \mathcal{M}$.*

Proof. It results that

$$\mathcal{E} \cap \mathcal{M} = \{z \in \mathbb{R}^n : z = z^o + M\alpha, \quad \alpha' M' Q M \alpha + 2z^{o'} Q M \alpha + z^{o'} Q z^o \leq 1\}$$

and thus, if $\mathcal{E} \cap \mathcal{M}$ is not empty, its symmetry center is $z^o + M\alpha_c$, with

$$\alpha_c = -(M'QM)^{-1}M'Qz^o$$

which coincides with z_{rp} in (3.15). \square

The previous result allows to characterize the degree of optimality of the restricted projection algorithm Φ_{rp} , for the cases where there exist reduced-complexity estimates which belong to the feasible set FPS_y .

Proposition 3.3 *If $\|\cdot\|_Y = \ell_2$ and $\mathcal{M} \cap FPS_y \neq \emptyset$, with \mathcal{M} as in (3.1), then the restricted projection algorithm Φ_{rp} is optimal within a factor 2.*

Proof. By Proposition 3.2 the estimate $z_{rp} = \Phi_{rp}(y)$ is the symmetry center of $\mathcal{M} \cap FPS_y$ and so $z_{rp} \in FPS_y$. Therefore Φ_{rp} turns out to be an interpolatory algorithm, and so optimal within a factor 2 by Proposition 1.4. \square

In other words, the previous proposition states that, if the estimation class is compatible with the available information (i.e. there exists at least one element of \mathcal{M} which is feasible), then also the restricted projection algorithm is almost-optimal. However, in general, it is not possible to guarantee an upper bound to the Y -local estimation error of the algorithm Φ_{rp} . This is shown in the following proposition.

Proposition 3.4 *For any $c > 0$, there exists a reduced-complexity estimation problem such that $E_y(\Phi_{rp}, \varepsilon) > c \cdot E_y(\Phi_{cc}, \varepsilon)$.*

Proof. See [23]. \square

Indeed, the error of the restricted projection algorithm can be arbitrarily large.

By previous results, it may appear that the central projection algorithm provides always a better estimate (as Y -local error) than the restricted projection algorithm. In the following example [23] it is shown that this is not always true.

Example 3.1 *Let us consider the ellipsoid \mathcal{E} , defined as in (3.8) with*

$$Q = \begin{bmatrix} 0.05 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 2.5 \end{bmatrix}$$

and the linear manifold \mathcal{M} in (3.10), where

$$z^o = \begin{bmatrix} -0.47 \\ -0.1 \\ 0.94 \end{bmatrix} \quad M = \frac{v}{\|v\|_2}, \quad \text{with } v = \begin{bmatrix} -0.17 \\ 1.269 \\ 0.05 \end{bmatrix}.$$

Computing the conditional central estimate z_{cc} (by means of the procedure reported in [19]), and the projection estimates z_{cp} and z_{rp} as in (3.14) and (3.15), one obtains the following estimation errors

$$\begin{aligned} d_\varepsilon(z_{cc}) &= 25.0691 \\ d_\varepsilon(z_{cp}) &= 25.3368 \\ d_\varepsilon(z_{rp}) &= 25.1121 \end{aligned}$$

and it is possible to conclude that, in this case, the central projection algorithm provides a larger error with respect to that provided by the restricted projection algorithm.

However, there exist special situations in which it is possible to a priori order the Y -local errors of the algorithms $\Phi_{cc}, \Phi_{cp}, \Phi_{rp}$. One of these is described in the following proposition.

Proposition 3.5 *Let $\|\cdot\|_Y = \ell_2$ and let \mathcal{M} be a linear manifold defined as in (3.1), with $h = n - 1$. Then*

i) the estimates $\Phi_{cc}, \Phi_{cp}, \Phi_{rp}$ are lined up on the manifold \mathcal{M} ;

ii) $E_y(\Phi_{cc}, \varepsilon) \leq E_y(\Phi_{cp}, \varepsilon) \leq E_y(\Phi_{rp}, \varepsilon) \quad \forall y$.

Proof. See [24]. □

Indeed, if the linear manifold \mathcal{M} has dimension $n - 1$, the error provided by the central projection algorithm is always less than that provided by the restricted projection one. It has been verified, by means of randomly generated experimental data, that there is a significant difference between performances of the two algorithms. In fact, examples in which the estimate z_{rp} is better than z_{cp} seem to occur quite rarely.

Orthonormal basis functions in conditional set-membership identification

In Section 3.3.1 it has been shown how a linearly parameterized manifold can be selected as a reduced-order model class, in order to perform restricted-complexity identification of a given transfer function (or impulse response sequence). In this chapter, the class of orthonormal basis functions previously introduced in Section 3.3.1 will be analyzed in details, with special emphasis on Laguerre filters.

The chapter is organized as follows. In Section 4.1 a general introduction to orthonormal basis functions is given. A description of the main classes of orthonormal basis functions is addressed in Section 4.2; in particular Laguerre, Kautz and generalized orthonormal basis functions are presented. In Section 4.3, the use of orthonormal basis functions in set-membership system identification is discussed. Particular emphasis on error bounds for different pole choices is addressed in Section 4.4, while simulation examples are reported in Section 4.5.

4.1 Orthonormal basis functions

Decomposing dynamical systems in terms of orthonormal expansions enables approximation of a system by a finite parameterization.

The importance of orthogonal basis functions goes beyond the areas of system identification and adaptive signal processing. Many problems in circuit theory, signal processing, telecommunications, system and control theory, estimation and optimization theory benefit from an efficient representation or parameterization of particular classes of signals/systems. A decomposition of a signal/system in term of independent orthogonal components play an important role in devising estimation/optimization algorithms where the choice for particular orthogonal structures can be made dependent on prior knowledge on the signal/system at hand.

Let us consider a stable system $G(z) \in \mathcal{H}$ described by

$$G(z) = \sum_{k=0}^{\infty} g_k z^{-k} \quad (4.1)$$

with $\{g_k\}_{k=0,1,2,\dots}$ the impulse response sequence and \mathcal{H} is a suitable space.

Let $\{f_k(z)\}_{k=0,1,2,\dots}$ be an orthonormal basis for the set of systems \mathcal{H} .

The orthonormality of the basis is defined as

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} f_k(e^{j\omega}) f_l(e^{-j\omega}) d\omega = \begin{cases} 1 & (k = l) \\ 0 & (k \neq l) \end{cases}. \quad (4.2)$$

Note that $f_k(z) = z^{-k}$ is one of the possible choices for such a basis; in this case the choice corresponds to the use of the so-called finite impulse response (FIR) models.

The following theorem holds.

Theorem 4.1 *Let $G(z)$ and $f_k(z)$ be defined as in (4.1), (4.2). Then there*

exists a unique expansion

$$G(z) = \sum_{k=0}^{\infty} L_k f_k(z) \quad (4.3)$$

with $\{L_k\}_{k=0,1,2,\dots}$ the real-valued expansion coefficients.

Usually a model of the system $G(z)$ is represented by a finite length expansion

$$\hat{G}(z) = \sum_{k=0}^{n-1} \hat{L}_k f_k(z). \quad (4.4)$$

In this case it is clear that the accuracy of the model, in terms of the minimal possible deviation between system and model (in any norm), will be essentially dependent on the choice of basis functions $f_k(z)$.

For this reason the development of appropriate basis functions is a topic that has attracted considerable interest. The issue here is that it is profitable to design basis functions that reflect the dominant dynamics of the process to be modelled.

Besides FIR, typical basis functions are Laguerre functions, Kautz functions [21] and generalized orthonormal basis functions [25, 20]. Laguerre functions involve a scalar design variable a that has to be chosen in a range that matches the dominant (first order) dynamics of the process to be modelled. Kautz functions are used for moderately damped systems, and consist in a second order generalization of the Laguerre functions. Recently a generalized set of orthonormal basis functions has been investigated; they are generated by inner all-pass transfer functions of any prechosen order. This type of basis functions generalizes the Laguerre and Kautz bases, which appear as special cases when choosing first and second order inner functions. Using generalized basis functions that contain system dynamics may be advantageous in identification and approximation problems. If the dynamics of the basis generating system and the dynamics of the system to be modelled are close, the convergence rate of a

series expansion of the system becomes very fast. Moreover the identification of expansion coefficients in a series expansion benefits very much from a fast convergence rate and the number of coefficients to be determined to accurately model the system becomes smaller. This leads to a reduction of both bias and variance in the estimated models.

4.2 Classes of orthonormal basis functions

4.2.1 Laguerre functions

It is possible to define Laguerre functions in continuous time or in discrete time. Let us analyze first the continuous time case.

Lemma 4.1 *Assume the function $G(s)$ to be strictly proper ($G(\infty) = 0$), analytic in $\text{Re}(s) > 0$ and continuous in $\text{Re}(s) \geq 0$. Let $a > 0$, then there exists a sequence $\{g_k\}$ such that*

$$G(s) = \sum_{k=1}^{\infty} g_k \frac{\sqrt{2a}}{s+a} \left(\frac{s-a}{s+a} \right)^{k-1}, \quad \text{Re}(s) \geq 0. \quad (4.5)$$

Proof. See [26]. □

The functions

$$L_i(s, a) = \frac{\sqrt{2a}}{s+a} \left(\frac{s-a}{s+a} \right)^{k-1} \quad (4.6)$$

consisting of a first order low-pass term and a $(k-1)$ th-order all pass factor, are the Laplace transforms of the classical Laguerre functions. These functions are orthogonal in $L_2(0, \infty)$, and form a complete set in $L_2(0, \infty)$ and in $L_1(0, \infty)$.

The parameter a is the so-called *Laguerre pole* and plays a fundamental role in determining the convergence rate of the approximation.

Since in system identification it is customary to have a fixed number of measurements taken at exact time intervals, the discrete Laguerre expansion is

usually adopted.

Lemma 4.2 *Assume the function $G(z)$ to be strictly proper ($G(\infty) = 0$), analytic in $|z| > 1$ and continuous in $|z| \geq 1$. Let $-1 < a < 1$, then there exists a sequence $\{g_k\}$ such that*

$$G(z) = \sum_{k=1}^{\infty} g_k \frac{\sqrt{1-a^2}}{z-a} \left(\frac{1-az}{z-a} \right)^{k-1}, \quad |z| \geq 1. \quad (4.7)$$

The functions

$$L_k(z, a) = \frac{\sqrt{1-a^2}}{z-a} \left(\frac{1-az}{z-a} \right)^{k-1}, \quad k = 1, 2, \dots \quad (4.8)$$

are the z -transform of the classical Laguerre sequences.

An alternative way to define Laguerre filters (in z^{-1}) is

$$L_k(z, a) = \frac{\sqrt{1-a^2}}{1-az^{-1}} \left(\frac{z^{-1}-a}{1-az^{-1}} \right)^k, \quad k = 0, 1, 2, \dots \quad (4.9)$$

In the following, only discrete Laguerre functions will be considered.

4.2.2 Optimal pole selection in Laguerre expansion

It is well known that the rate of approximation of a function by Laguerre filters heavily depends on the pole choice. A typical example consists in the approximation of a function $h(t)$ by a truncated Laguerre series with a fixed number of terms. One has

$$h_n(t) = \sum_{i=0}^n c_i(a) l_i(t, a) \quad (4.10)$$

where $l_i(t, a)$ is the time response of $L_i(z, a)$ and $c_i(a)$ are the optimal coefficients of the linear combination, i.e.

$$c_i(a) = \arg \min_{\substack{c_i \in \mathbb{R} \\ i=0,1,\dots,n}} \|h(t) - h_n(t)\|_2 \quad (4.11)$$

with $h_n(t)$ given by (4.10).

Let us assume that we want to evaluate the minimum deliverable estimation error in ℓ_2 norm, that is

$$E_n(a) = \|h(t) - h_n(t)\|_2 \quad (4.12)$$

where the parameter a is indicated to emphasize the dependence of the approximation error on the pole.

It is possible to rewrite the error in the following way

$$E_n(a) = \langle h(t), h(t) \rangle - \sum_{i=0}^n c_i^2(a). \quad (4.13)$$

Now, the aim is to find the optimal position of the pole a , that is the value of a such that $E_n(a)$ is minimized. It has been shown that at each stationary point of $E_n(a)$ the following condition holds [27, 28, 29].

$$c_n(a) c_{n+1}(a) = 0. \quad (4.14)$$

Clearly, the determination of all stationary points of the squared error $E_n(a)$ requires the solution of the equations $c_i(\cdot) = 0$, for $i = n$ and $i = n + 1$. This is in general a very difficult task. Many procedures have been proposed in the past years to solve this problem, such as for instance in [30, 27, 28, 31, 32, 29, 33]. Another approach to this problem, leading to the optimal position of the pole for a class of systems satisfying certain measurements, has been given in [34, 35].

In the following example it is shown that the approximation of a function depends strongly on the choice of the Laguerre pole.

Example 4.1 Let $H(z) = \frac{z^3 - 1.2z^2 + 0.7z - 1.5}{z^4 + 1.2z^3 + 0.65z^2 + 0.21z - 0.01}$ and let h be its impulse response ($N=20$).

Let \mathcal{M}_p be the model class of Laguerre filters (of order $n = 4$) with pole p and $\tilde{h}(p) = \arg \inf_{\tilde{h} \in \mathcal{M}_p} \|h - \tilde{h}\|_2$ be the best approximation for the pole p .

In Fig. 4.1 the true and the approximated impulse response coefficients are plotted, while in Table 4.1 the numerical values of estimation errors are provided.

It is clear by this example that the approximation strongly depends on the pole choice of Laguerre functions.

4.2.3 Kautz functions

Although the use of Laguerre filters in system estimation/identification is quite common, a known drawback of this kind of functions is that, in general, poorly damped systems are difficult to approximate with a reasonable number of terms. For this reason other kinds of orthonormal basis functions have been introduced.

In this section, the so-called (two parameters) discrete Kautz functions are briefly described. Through these functions it is possible to approximate more efficiently signals with strong oscillatory behaviour [36, 21].

Like for Laguerre filters, for a large number of expansion terms, the choice of the free parameters is not crucial, but for a limited number of expansion terms the choice of the parameters is of great importance to provide a good approximation of a given function.

The discrete Kautz functions can be written in various forms. One possible way to define them is through their z -transform $K_l(b, c)$, i.e.

$$G(z) = \sum_{j=0}^{\infty} L_j K_j(z)$$

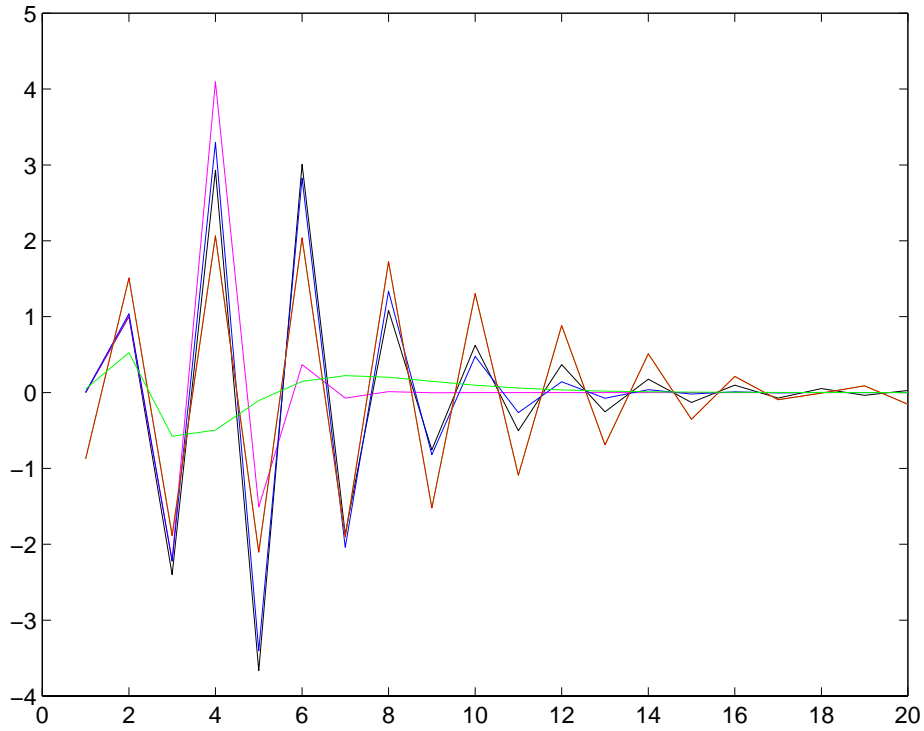


Figure 4.1: Nominal model (black) and Laguerre approximations of 20 impulse response coefficients for different pole choice.

Laguerre pole	$\ h - \tilde{h}(p)\ _2$	Color (Fig. 4.1)
-0.9	2.8116	red
-0.4	0.7598	blue
-0.1	4.3441	magenta
0.4	6.5548	green

Table 4.1: Comparison among approximation errors for different pole choice of the Laguerre filters.

with

$$K_{2n}(z) = \frac{z\sqrt{(1-c^2)(1-b^2)}}{z^2 + b(c-1)z - c} \left(\frac{-cz^2 + b(c-1)z + 1}{z^2 + b(c-1)z - c} \right)^n$$

$$K_{2n+1}(z) = \frac{z(z-b)\sqrt{1-c^2}}{z^2 + b(c-1)z - c} \left(\frac{-cz^2 + b(c-1)z + 1}{z^2 + b(c-1)z - c} \right)^n$$

where

$$-1 < b < 1 \quad , \quad -1 < c < 1 \quad , \quad n = 0, 1, 2, \dots$$

Compared to Laguerre filters, there are now two free parameters b and c , which make it possible to assign a pair of complex conjugate poles to the transfer functions $K_j(z)$.

4.2.4 Generalized orthonormal basis functions

Generalized Orthonormal Basis Functions (GOBF) are extensions of the previously described functions. These functions were introduced in [25, 37, 38], and the main result concerning them is reported in the following theorem.

Theorem 4.2 *Let $G_b(z)$ be a scalar inner function¹ with McMillan degree $n_b > 0$, having a minimal balanced realization (A, B, C, D) . Denote*

$$V_k(z) = z(zI - A)^{-1}BG_b^k(z). \quad (4.15)$$

Then the sequence of scalar rational functions $\{e_i^T V_k(z)\}_{i=1, \dots, n_b; k=0, \dots, \infty}$, where e_i is the i -th euclidean basis vector in \mathbb{R}^{n_b} , forms an orthonormal basis for the Hilbert space \mathcal{H}_2 .

Note that these basis functions exhibit the property that they can approximate system dynamics in a very general way. One can construct an inner function G_b from any set of poles, and thus the resulting basis can incorporate dynamics of any complexity, combining, for example, both fast and slow dynamics in

¹A function $G(z)$ is called *inner* if it is stable and it satisfies $G(z^{-1})G(z) = 1$.

damped and resonant modes. A direct result is that for any specifically chosen $V_k(z)$, any strictly proper transfer function $G(z) \in \mathcal{H}_2$ has a unique series expansion

$$G(z) = z^{-1} \sum_{k=0}^{\infty} L_k V_k(z) \quad (4.16)$$

with $L_k \in \ell_2^{1 \times n_b}[0, \infty)$.

In the following it is shown how specific choices of $G_b(z)$ can generate well known classical basis functions.

- With $G_b(z) = z^{-1}$, having minimal balanced realization $(0, 1, 1, 0)$, the standard FIR basis $V_k(z) = z^{-k}$ results.
- Choosing a first order inner function

$$G_b(z) = \frac{(1 - az)}{(z - a)}$$

with some real valued a , $|a| < 1$, and balanced realization

$$(A, B, C, D) = (a, \sqrt{1 - a^2}, \sqrt{1 - a^2}, -a),$$

the Laguerre basis is obtained.

- Similarly, the Kautz functions originate from the choice of a second order inner function

$$G_b(z) = \frac{-cz^2 + b(c - 1)z + 1}{z^2 + b(c - 1)z - c}$$

with some real valued b and c satisfying $|c|, |b| < 1$. A balanced realization of $G_b(z)$ can be found to be given by (see [38])

$$A = \begin{bmatrix} b & \sqrt{1 - b^2} \\ c\sqrt{1 - b^2} & -bc \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \sqrt{1 - c^2} \end{bmatrix}$$

$$c = \begin{bmatrix} \sqrt{(1 - c^2)(1 - b^2)} & -b\sqrt{1 - c^2} \end{bmatrix}, \quad D = -c.$$

The generalized orthonormal basis for \mathcal{H}_2 also induces a similar basis for the signal space $\ell_2[0, \infty)$ of squared summable sequences, through inverse z -transformation to the signal domain. Denoting

$$V_k(z) = \sum_{l=0}^{\infty} \phi_k(l) z^{-l}$$

it follows that $\{e_i^T \phi_k(l)\}_{i=1, \dots, n_b; k=0, \dots, \infty}$ is an orthonormal basis for the signal space $\ell_2[0, \infty)$. These ℓ_2 basis functions can be also constructed directly from G_b and its balanced realization (A, B, C, D) (see [38]).

4.3 Orthonormal basis functions in conditional set-membership identification

In Chapter 3 the concept of conditional identification has been introduced. It has been shown that in this context it is common to restrict the set of the estimated model inside a linearly parameterized set \mathcal{M} .

A typical choice of the set \mathcal{M} is given by the impulse responses of orthonormal basis functions. Let us consider the framework described in Section 3.3.1 regarding \mathcal{H}_2 conditional identification and let the impulse response of a system be denoted by $h \in \mathcal{H}$. A priori knowledge on the system is expressed as $h \in \mathcal{S}$, where \mathcal{S} is a set contained in \mathcal{H} . It is possible to write the set \mathcal{M} emphasizing the dependence on the pole vector $p \in \mathbb{R}^m$

$$\mathcal{M} = \{h : h = M_p \theta, \theta \in \mathbb{R}^n\} \quad (4.17)$$

where M_p is a linear operator and θ is the n -dimensional parameter vector to be identified, $n < N$. For example, for Laguerre filters, p is the real Laguerre pole ($m = 1$), while for Kautz functions p denotes the pair of complex conjugate poles ($m = 2$).

It is known that, once the pole vector p is fixed, the optimal solution of this

problem is given by the conditional central algorithm, i.e.

$$\Phi_{cc}(y) = M_p \theta_{cc}$$

where

$$\theta_{cc} = \arg \inf_{\theta \in \mathbb{R}^n} \sup_{h \in \mathcal{F}} \|h - M_p \theta\| \quad (4.18)$$

and \mathcal{F} is the feasible set given by (see (3.16))

$$\mathcal{F} = FSS_y = \{h \in \mathcal{S} : \|UT^N h - y\|_2 \leq \varepsilon\}. \quad (4.19)$$

It is straightforward to define the optimal pole (for a given function $h \in \mathcal{H}$ and a model class \mathcal{M}) as

$$p^*(h) = \arg \inf_{p \in \mathcal{P}} \inf_{\theta \in \mathbb{R}^n} \|h - M_p \theta\| \quad (4.20)$$

where \mathcal{P} is the set of admissible pole locations, which usually rely on a priori knowledge on the true system.

In the context of conditional set membership identification, the selection of the optimal pole is performed with respect to all the elements in the feasible set, i.e. via the minimization of the worst-case error. This leads to the following optimization problem

$$p^*(\mathcal{F}) = \arg \inf_{p \in \mathcal{P}} \inf_{\theta \in \mathbb{R}^n} \sup_{h \in \mathcal{F}} \|h - M_p \theta\|. \quad (4.21)$$

This is, in general, a very complicated min-max optimization problem for which the derivation of simple conditions appears to be an awkward task. For this reason suboptimal algorithms described in Section 3.2 are commonly used. In particular, let us analyze the central projection algorithm and an interpolatory projection algorithm.

- *Central projection algorithm.*

Let \mathcal{F} be the FSS_y and let h_c be the Chebyshev center of \mathcal{F} in a given norm, as stated in Definition 1.1, i.e.

$$h_c = \arg \inf_{h \in \mathcal{H}} \sup_{\tilde{h} \in \mathcal{F}} \|h - \tilde{h}\|.$$

Then,

$$h_{cp} = \Phi_{cp}(y, p^*(h_c)) = M_{p^*(h_c)} \theta^*(h_c) \quad (4.22)$$

where

$$\theta^*(h_c) = \arg \inf_{\theta \in \mathbb{R}^n} \|h - M_{p^*(h_c)} \theta\|. \quad (4.23)$$

The resulting estimation error is

$$E[\Phi_{cp}(y; p^*(h_c))] = \sup_{h \in \mathcal{F}} \|h - \Phi_{cp}(y, p^*(h_c))\|. \quad (4.24)$$

- *Interpolatory projection algorithm.*

Let $h_i \in \mathcal{F}$ be an element of the feasible set. Then,

$$h_{ip} = \Phi_{ip}(y, p^*(h_i)) = M_{p^*(h_i)} \theta^*(h_i) \quad (4.25)$$

where

$$\theta^*(h_i) = \arg \inf_{\theta \in \mathbb{R}^n} \|h - M_{p^*(h_i)} \theta\|. \quad (4.26)$$

The resulting estimation error is

$$E[\Phi_{ip}(y; p^*(h_i))] = \sup_{h \in \mathcal{F}} \|h - \Phi_{ip}(y, p^*(h_i))\|. \quad (4.27)$$

Since the model class \mathcal{M} must be selected via optimization with respect to poles p , as in the computation of (4.22) and (4.25), one can first compute h_c or h_i , then exploit condition (4.20) to obtain $p^*(h_c)$ or $p^*(h_i)$, and hence the estimates h_{cp} and h_{ip} .

Note that for these suboptimal algorithms, the optimal pole p^* is computed with respect to an element (h_c and h_i respectively), whereas for the optimal algorithm it should be computed with respect to the feasible set \mathcal{F} , see (4.21).

In Fig. 4.2 it is sketched how the chosen pole (and hence the corresponding subspace M_{p^*}) can change depending on different suboptimal algorithms.

In the next section some results on the suboptimality degree of algorithms (4.22) and (4.25) with respect to the minimum identification error will be provided (see also [39, 40]).

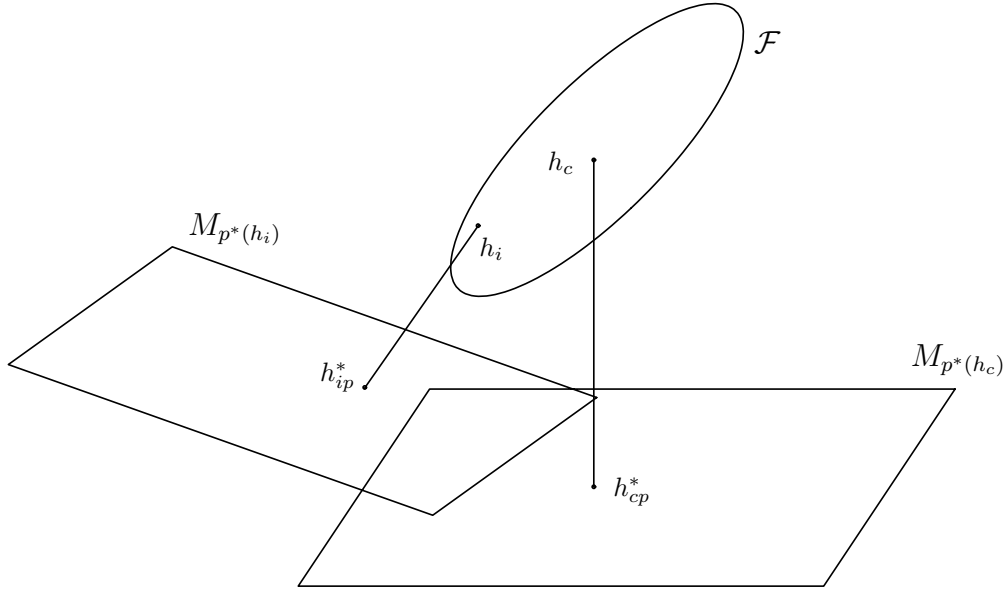


Figure 4.2: Example showing how optimal subspace M_{p^*} can change depending the approximation algorithm chosen.

4.4 Suboptimal pole choice and error bounds

The aim of this section is to derive tight bounds on the identification error provided by the projection algorithms (4.22) and (4.25), i.e. to determine the minimum $\kappa \geq 1$, such that

$$E[\Phi_{cp}(y; p^*(h_c))] \leq \kappa \cdot E[\Phi_{cc}(y; p^*(\mathcal{F}))]$$

for all possible y and \mathcal{F} (and similarly for an interpolatory projection algorithm Φ_{ip}).

Before proceeding, it is useful to recall that in the conditional set membership identification setting, several problems of interest can be restricted to the finite dimensional space \mathbb{R}^N . In fact, let T^N denote the truncation operator in \mathcal{H} , such that $T^N h = h^N$, and R^N be the remainder operator $R^N h = \{h\}_{i=N}^\infty$. Then, under the mild assumption that $R^N \mathcal{S}$ is a balanced set (i.e., if $h \in R^N \mathcal{S}$, then also $-h \in R^N \mathcal{S}$), it can be shown that for any ℓ_p norm, $1 \leq p < \infty$, one has $E[T^N \Phi] \leq E[\Phi]$ for any conditional algorithm Φ , model class \mathcal{M} and

feasible set \mathcal{F} . Therefore, one can consider only truncated basis expansions such as

$$\mathcal{M} = \{h^N \in \mathbb{R}^N : h^N = M_p \theta, \theta \in \mathbb{R}^n, n < N\}. \quad (4.28)$$

Consequently, an identification algorithm turns out to be a mapping from \mathbb{R}^N to an n -dimensional subspace of \mathbb{R}^N (the truncated model class). Moreover, one has

$$E[\Phi, \mathcal{M}] = \left(\sup_{h^N \in T^N \mathcal{F}} \|h^N - \Phi(y)\|_p^p + \sup_{h \in \mathcal{S}} \|R^N h\|_p^p \right)^{1/p} \quad (4.29)$$

(in the following, dependence on \mathcal{M} will be omitted to simplify notation). The rightmost term in (4.29) depends only on \mathcal{S} and can be computed a priori. Hence, in the following, the finite-dimensional feasible set $\mathcal{F}_N = T^N \mathcal{F} \subset \mathbb{R}^N$ will be considered, when computing the estimates $\Phi_{cc}(y; p^*(\mathcal{F}_N))$, $\Phi_{cp}(y; p^*(h_c))$ and $\Phi_{ip}(y; p^*(h_i))$.

The following results are valid when the ℓ_2 identification error is considered, i.e. $\|\cdot\|_{\mathcal{H}} = \|\cdot\|_2$ (to simplify the notation, the ℓ_2 norm will be denoted by $\|\cdot\|$). Let us recall the Chebyshev radius of \mathcal{F}_N as $\text{rad}(\mathcal{F}_N) = \sup_{h \in \mathcal{F}_N} \|h - h_c\|$, where h_c is the Chebyshev center of \mathcal{F}_N in the ℓ_2 norm. Now, the following result can be stated.

Theorem 4.3 *Let $r = \text{rad}(\mathcal{F}_N)$ and $d = \|h_c - h_{cp}\|$. Then,*

$$E[\Phi_{cp}(y; p^*(h_c))] \leq \sqrt{2 - \frac{(r-d)^2}{r^2 + d^2}} \cdot E[\Phi_{cc}(y; p^*(\mathcal{F}_N))]. \quad (4.30)$$

Proof. In order to prove the theorem, the following lemma is needed (for a proof, see [23]).

Lemma 4.3 *Let h_c be the Chebyshev center of \mathcal{F}_N in the ℓ_2 norm and consider the closed halfspace $\mathcal{Q} = \{h \in \mathbb{R}^N : a_q^T h \geq b_q, a_q \in \mathbb{R}^N, b_q \in \mathbb{R}\}$, such that $a_q^T h_c = b_q$. Then, there exists $h_e \in \mathcal{F}_N \cap \mathcal{Q}$ such that $\|h_e - h_c\|_2 = \text{rad}(\mathcal{F}_N)$.*

Now, let us first consider $E[h_{cp}] = E[\phi_{cp}(y; p^*(h_c))]$. One has

$$E[h_{cp}] = \sup_{h \in \mathcal{F}_N} \|h_{cp} - h\| \leq \|h_{cp} - h_c\| + \sup_{h \in \mathcal{F}_N} \|h_c - h\| = d + r \quad (4.31)$$

with d and r defined as in the statement of Theorem 4.3.

Then, let us analyze the minimum error $E[h_{cc}] = E[\phi_{cc}(y; p^*(\mathcal{F}_N))]$. Consider the halfspace $\mathcal{Q}_{cc} = \{h \in \mathbb{R}^N : (h - h_c)'(h_c - h_{cc}) \geq 0\}$. Lemma 4.3 guarantees that there exists $h_e \in \mathcal{F}_N \cap \mathcal{Q}_{cc}$ such that $\|h_e - h_c\| = r$. Therefore, one has

$$\begin{aligned} E[h_{cc}]^2 &\geq \|h_e - h_{cc}\|^2 \\ &= \|h_e - h_c\|^2 + \|h_c - h_{cc}\|^2 + 2 \frac{(h_e - h_c)'(h_c - h_{cc})}{\|h_e - h_c\| \|h_c - h_{cc}\|} \\ &\geq \|h_e - h_c\|^2 + d^2 = r^2 + d^2 \end{aligned} \quad (4.32)$$

where it has been exploited the fact that $\|h_c - h_{cc}\| \geq d$, which follows from the definitions of h_{cp} and d . Then, from (4.31) and (4.32) one has $\frac{E[h_{cp}]}{E[h_{cc}]} \leq \frac{r+d}{\sqrt{r^2+d^2}} = \sqrt{2 - \frac{(r-d)^2}{r^2+d^2}}$ which proves the theorem. \square

It is worth remarking that Theorem 4.3 holds for any N (dimension of the feasible set depending on the data set) and for any n (model order), $n < N$. The maximum value of the bound with respect to all possible feasible sets (i.e. with respect to all $r \geq 0$, $d \geq 0$) is given by the next corollary, which follows immediately from Theorem 4.3, when $r = d$.

Corollary 4.1 *For all $\mathcal{F}_N \subset \mathbb{R}^N$*

$$E[\Phi_{cp}(y; p^*(h_c))] \leq \sqrt{2} \cdot E[\Phi_{cc}(y; p^*(\mathcal{F}_N))] .$$

The next theorem shows that the bound on the identification error is *tight*, i.e. it is possible to find a feasible set \mathcal{F}_N and a family of model classes $\mathcal{M}(p)$, depending on p , such that in (4.30) equality holds.

Theorem 4.4 *Let $\mathcal{M}(p)$ be a family of orthonormal bases as in (4.28), with $M_p \in \mathbb{R}^{N \times n}$. Assume that for some $\bar{h} \in \mathbb{R}^N$, the infimum in (4.20) is achieved*

for two distinct pole vectors p_1^*, p_2^* , and let θ_1^*, θ_2^* be the corresponding optimal parameter vectors, so that

$$\|\bar{h} - M_{p_1^*} \theta_1^*\| = \|\bar{h} - M_{p_2^*} \theta_2^*\| \leq \|\bar{h} - M_p \tilde{\theta}\|, \quad \forall p, \quad \forall \tilde{\theta} \in \mathbb{R}^n. \quad (4.33)$$

Moreover, let $(\bar{h} - M_{p_1^*} \theta_1^*)'(\bar{h} - M_{p_2^*} \theta_2^*) = 0$. Then, there exists a feasible set $\mathcal{F}_N \subset \mathbb{R}^N$ such that

$$E[\Phi_{cp}(y; p^*(h_c))] = \sqrt{2} \cdot E[\Phi_{cc}(y; p^*(\mathcal{F}_N))].$$

Proof. Let $\mathcal{F}_N = \{h \in \mathbb{R}^N : h = \bar{h} + \alpha(\bar{h} - h_{cp}), |\alpha| \leq 1\}$, where $h_{cp} = M_{p_1^*} \theta_1^*$ has been chosen as the central projection (this choice is correct, as $\bar{h} = \text{cen}(\mathcal{F}_N)$ and (4.33) holds). Moreover, let $d = \|\bar{h} - h_{cp}\|$. It is easy to show that

$$\begin{aligned} E[h_{cp}] &= \sup_{h \in \mathcal{F}_N} \|h_{cp} - h\| \\ &= \|h_{cp} - (\bar{h} + \bar{h} - h_{cp})\| \\ &= \|2(h_{cp} - \bar{h})\| = 2d. \end{aligned} \quad (4.34)$$

Now, let $\hat{h} = M_{p_2^*} \theta_2^*$. Then, from the assumptions of the theorem, one has $\|\bar{h} - \hat{h}\| = d$ and $(\bar{h} - \hat{h})'(\bar{h} - h_{cp}) = 0$. Therefore, it follows that

$$\begin{aligned} E[\hat{h}]^2 &= \sup_{h \in \mathcal{F}_N} \|\hat{h} - h\|^2 \\ &= \|(\hat{h} - \bar{h}) - (h_{cp} - \bar{h})\|^2 \\ &= \|\hat{h} - \bar{h}\|^2 + \|h_{cp} - \bar{h}\|^2 = 2d^2. \end{aligned} \quad (4.35)$$

Hence, from (4.34) and (4.35) one has $\frac{E[h_{cp}]}{E[\hat{h}]} = \frac{2d}{\sqrt{2}d} = \sqrt{2}$ which means that $h_{cc} = \hat{h}$ (otherwise Theorem 4.3 would be violated) and the upper bound is achieved. \square

Example 4.2 In Fig. 4.4 it is shown a simple illustrative example ($N = 2$, $n = 1, m = 1$); the feasible set is an ellipsoid while the adopted basis is the Laguerre one. It is possible to note that the optimal pole relative to the central projection is $p^*(h_c) \simeq -1$, while the one corresponding to the conditional center

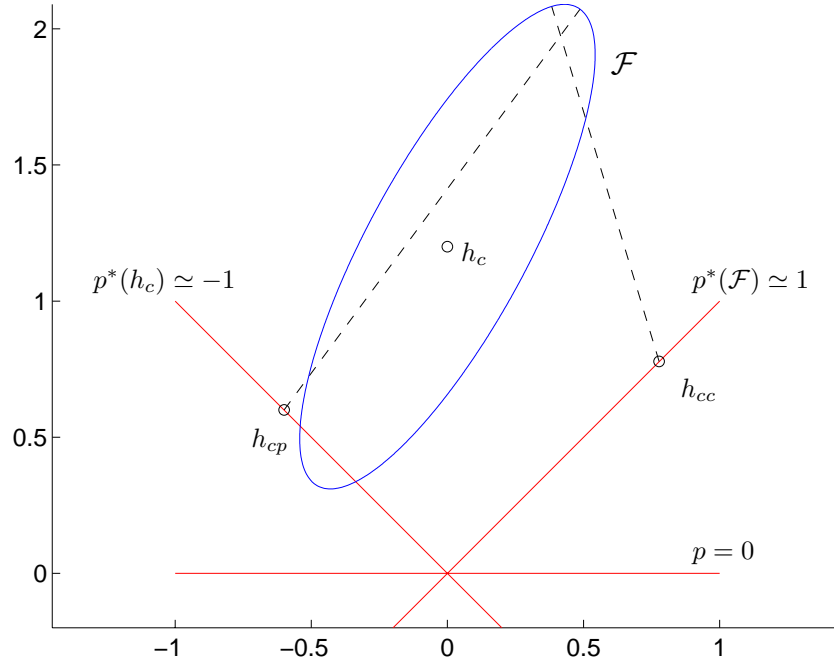


Figure 4.3: Example showing the different optimal subspaces and approximation errors for different estimators.

is $p^*(\mathcal{F}) \simeq 1$. It results that $E[\Phi_{cp}(y; p^*(h_c))] \simeq 1.34 E[\Phi_{cc}(y; p^*(h_{\mathcal{F}}))]$ which is quite large but less than $\sqrt{2}$ as stated in Corollary 4.1.

A bound on the identification error can be provided also for interpolatory projection algorithms. Unfortunately, this turns out to be much larger than that given by Theorem 4.3.

Theorem 4.5 Let $r = \text{rad}(\mathcal{F}_N)$ and $d_i = \|h_i - h_{ip}\|$. Then,

$$E[\Phi_{ip}(y; p^*(h_i))] \leq \min \left\{ 2 + \frac{d_i}{r}, 1 + \frac{2r}{d_i} \right\} \cdot E[\Phi_{cc}(y; p^*(\mathcal{F}_N))]. \quad (4.36)$$

Proof. For any $h \in \mathcal{F}_N$ one has

$$\begin{aligned} \|h_{ip} - h\| &\leq \|h_{ip} - h_c\| + \|h_c - h\| \\ &= \|\Pi_n h_i - h_c\| + \|h_c - h\| \end{aligned}$$

$$\begin{aligned}
&\leq \|\Pi_n h_i - h_i\| + \|h_i - h_c\| + \|h_c - h\| \\
&\leq d_i + r + \|h_c - h\|
\end{aligned} \tag{4.37}$$

where $\Pi_n h_i$ is the projection of h_i onto the linear manifold \mathcal{M} . Then, maximizing both sides of (4.37) over $h \in \mathcal{F}_N$ one gets

$$E[h_{ip}] \leq d_i + 2r. \tag{4.38}$$

On the other hand, one has that

$$d_i = \|\Pi_n h_i - h_i\| = \inf_p \inf_{\theta \in \mathbb{R}^n} \|h_i - M_p \theta\| \leq \inf_p \inf_{\theta \in \mathbb{R}^n} \sup_{\tilde{h} \in \mathcal{F}_N} \|\tilde{h} - M_p \theta\| = E[h_{cc}].$$

Since, by definition, $r = \text{rad}(\mathcal{F}_N) \leq \sup_{\tilde{h} \in \mathcal{F}_N} \|h_{cc} - \tilde{h}\| = E[h_{cc}]$, one has

$$E[h_{cc}] \geq \max\{d_i, r\} \tag{4.39}$$

and the result follows immediately from (4.38) and (4.39). \square

In order to obtain the maximum value of the bound in a worst-case setting, one has to consider the worst interpolatory estimator, i.e. the projection of the worst $h_i \in \mathcal{F}_N$, and all possible feasible sets. This corresponds to maximizing (4.36) with respect to all $r \geq 0$, $d_i \geq 0$, thus obtaining the bound in the next corollary, which follows immediately from Theorem 4.5, for $r = d_i$.

Corollary 4.2 *For all $\mathcal{F}_N \subset \mathbb{R}^N$ and $h_i \in \mathcal{F}_N$,*

$$E[\Phi_{ip}(y; p^*(h_i))] \leq 3 \cdot E[\Phi_{cc}(y; p^*(\mathcal{F}_N))].$$

Also the above bound turns out to be tight, as one can find a feasible set \mathcal{F}_N , an element $h_i \in \mathcal{F}_N$ and a family of model classes $\mathcal{M}(p)$, depending on p , such that the error of the interpolatory projection algorithm is arbitrarily close to three times the error of the optimal algorithm Φ_{cc} .

Remark 4.1 *It is interesting to compare the above results to those reported in Section 3.4 for the case of a fixed model class \mathcal{M} , with poles p assigned a priori.*

It has been shown that $E[\Phi_{cp}] \leq \sqrt{4/3} E[\Phi_{cc}]$ and $E[\Phi_{ip}] \leq 2 E[\Phi_{cc}]$; moreover, there exist feasible sets \mathcal{F}_N and model classes \mathcal{M} for which the equality holds. Obviously, in this context, the ratio between the worst-case identification error of the projection algorithms and the minimum achievable error is larger, due to the fact that suboptimal algorithms select the poles p by optimizing over a single element related to \mathcal{F}_N (namely h_c or h_i), while the minimum error is achieved by choosing $p^(\mathcal{F}_N)$ as in (4.21), where the whole feasible set is considered. Nevertheless, the bounds provided by the results in this section are useful, as they clarify that the maximum possible gap between a “set-oriented” choice of the poles and a choice based only on a single element is not very large. This is especially true for the central projection algorithm, which is in turn much easier to compute than $\Phi_{cc}(y; p^*(\mathcal{F}_N))$, in particular when the feasible set admits a symmetry center.*

4.4.1 Error bounds for ellipsoidal feasible sets

In some identification problems, the feasible set has a special structure that can be exploited in the computation of bounds on the identification error. If the noise is bounded in the ℓ_2 norm, and the a priori set \mathcal{S} provides constraints only on the tail $R^N h$ of the impulse response (the so-called *residual* a priori information, often adopted in the literature, see e.g. [22]), the feasible system set \mathcal{F}_N is an N -dimensional ellipsoid. Theorems 4.3 and 4.4 guarantee that the error provided by the central projection algorithm is not greater than $\sqrt{2}$ times the minimum error. However, for ellipsoidal feasible sets this bound can be further reduced, exploiting the special structure of the set.

Theorem 4.6 *Let $\mathcal{F}_N \subset \mathbb{R}^N$ be an ellipsoid of center h_c , and let L_M and L_m be the lengths of its maximum and minimum semi-axis, respectively. Moreover, define $d = \|h_c - h_{cp}\|$. Then,*

$$\frac{E[\Phi_{cp}(y; p^*(h_c))]}{E[\Phi_{cc}(y; p^*(\mathcal{F}_N))]} \leq \begin{cases} \frac{L_M + d}{L_M \sqrt{1 + \frac{d^2}{L_M^2 - L_m^2}}} & \text{if } d < \frac{L_M^2 - L_m^2}{L_m} \text{ or } L_m = 0, \\ \frac{L_M + d}{L_m + d} & \text{if } d \geq \frac{L_M^2 - L_m^2}{L_m} \text{ and } L_m > 0. \end{cases} \quad (4.40)$$

Moreover this upper bound is tight.

Proof. In order to prove Theorem 4.6, the following lemmas are needed.

Lemma 4.4 Let $\varepsilon = \{x \in \mathbb{R}^N : x'Qx \leq 1\}$ be a non-degenerate axes-oriented ellipsoid, such that $Q = \text{diag}\{q_i\}_{i=1}^N$, with $0 < q_1 \leq q_2 \leq \dots \leq q_N$. Moreover let $\mathcal{B} = \{z \in \mathbb{R}^N : z'z \geq d, d > 0\}$. Define

$$z^* = \arg \inf_{z \in \mathcal{B}} \sup_{x \in \varepsilon} \|z - x\|_2^2. \quad (4.41)$$

Then, $z_1^* = 0$.

Proof. W.l.o.g., it can be assumed $d = 1$. Moreover, it is straightforward to show that the minimum in (4.41) is reached on the boundary of \mathcal{B} , i.e. $\|z^*\| = 1$. From Theorem 2 in [19], one has that for any z such that $z_1 \neq 0$

$$\max_{x \in \varepsilon} \|z - x\| = \|(I_N - \lambda^*Q)^{-1} \lambda^*Q z\|^2, \quad (4.42)$$

where λ^* is the largest real solution of the equation

$$z'(I_N - \lambda Q)^{-2}Qz - 1 = 0. \quad (4.43)$$

Moreover, it is known that $\lambda^* > \frac{1}{q_1}$. Hence, using (4.42) one has that z^* is the solution of

$$\inf_{z: z'z=1} \sum_{i=1}^N \left(\frac{\lambda^* q_i z_i}{1 - \lambda^* q_i} \right)^2. \quad (4.44)$$

Exploiting (4.43) and substituting $z_1^2 = 1 - \sum_{i=2}^N z_i^2$ into (4.44), one has

$$\sum_{i=1}^N \left(\frac{\lambda^* q_i z_i}{1 - \lambda^* q_i} \right)^2 = \left(\frac{\lambda^* q_1}{1 - \lambda^* q_1} \right)^2 + \sum_{i=2}^N \left\{ \left(\frac{\lambda^* q_i}{1 - \lambda^* q_i} \right)^2 - \left(\frac{\lambda^* q_1}{1 - \lambda^* q_1} \right)^2 \right\} z_i^2.$$

Since $q_1 \leq q_i$, for $i \geq 2$, it is easy to show that $\gamma_i \triangleq \left(\frac{\lambda^* q_i}{1-\lambda^* q_i}\right)^2 - \left(\frac{\lambda^* q_1}{1-\lambda^* q_1}\right)^2 < 0$, $i = 2, \dots, N$ and hence the minimum in (4.44) is achieved for some z_2^*, \dots, z_N^* such that $(z_2^*)^2 + \dots + (z_N^*)^2 = 1$. This implies $z_1^* = 0$. \square

A straightforward extension of Lemma 4.4 is obtained by translating the center of the ellipsoid ε and by rotating the ellipsoid semi-axes onto a new reference system spanned by the orthonormal basis $\{v_1, \dots, v_N\}$. Hence, the next result holds.

Lemma 4.5 *Let $\varepsilon = \{x \in \mathbb{R}^N : x = h_c + \sum_{i=1}^N \alpha_i v_i, \frac{\alpha_1^2}{L_M^2} + \frac{1}{L_m^2} \sum_{i=2}^N \alpha_i^2 \leq 1; h_c \in \mathbb{R}^N; \|v_i\| = 1, i = 1, \dots, N; v_i' v_j = 0, i \neq j; L_M > L_m > 0\}$ and $\mathcal{B} = \{z : \|z - h_c\| \geq d, d > 0\}$. Define*

$$z^* = \arg \inf_{z \in \mathcal{B}} \sup_{x \in \varepsilon} \|z - x\|_2^2.$$

Then, $(z^ - h_c) \in \text{span}\{v_2, \dots, v_N\}$ and $\|z^* - h_c\| = d$.*

Now Theorem 4.6 can be proven. Let us first consider $E[h_{cp}] = E[\phi_{cp}(y; p^*(h_c))]$. One has

$$E[h_{cp}] = \sup_{h \in \mathcal{F}_N} \|h_{cp} - h\| \leq \|h_{cp} - h_c\| + \sup_{h \in \mathcal{F}_N} \|h_c - h\| = d + L_M. \quad (4.45)$$

Then, let us analyze the minimum error $E[h_{cc}] = E[\phi_{cc}(y; p^*(\mathcal{F}_N))]$. Let $\overline{\mathcal{F}}_N$ be an ellipsoid with the same center and axes orientation as \mathcal{F}_N , maximum semi-axis of length L_M and all other semi-axes of length L_m . By construction, $\overline{\mathcal{F}}_N \subseteq \mathcal{F}_N$. Moreover, $\overline{\mathcal{F}}_N$ coincides with ε in Lemma 4.5 (with v_1, \dots, v_N being the directions of the semi-axes of \mathcal{F}_N). By definition of h_{cp} in (4.22), one has that the conditional center h_{cc} satisfies $\|h_{cc} - h_c\| \geq \|h_{cp} - h_c\| = d$. Hence $E[h_{cc}] = \sup_{h \in \mathcal{F}_N} \|h_{cc} - h\| \geq \sup_{\bar{h} \in \overline{\mathcal{F}}_N} \|h_{cc} - \bar{h}\| \geq \inf_{z: \|z - h_c\| \geq d} \sup_{\bar{h} \in \overline{\mathcal{F}}_N} \|z - \bar{h}\|$. Then, from Lemma 4.5 one gets

$$E[h_{cc}] \geq \sup_{\bar{h} \in \overline{\mathcal{F}}_N} \|z^* - \bar{h}\| \quad (4.46)$$

for some z^* such that $(z^* - h_c) \in \text{span}\{v_2, \dots, v_N\}$ and $\|z^* - h_c\| = d$. W.l.o.g., assume $(z^* - h_c) \in \text{span}\{v_N\}$ (a rotation of the axes v_2, \dots, v_N can be applied

without affecting $\overline{\mathcal{F}}_N$) and set $z^* = h_c - d v_N$ (the sign of v_N can be chosen arbitrarily).

First, observe that when $L_m = 0$ the ellipsoid $\overline{\mathcal{F}}_N$ collapses onto a segment with extremal points $\overline{h} = h_c \pm L_M v_1$. Then, one has $\|z^* - \overline{h}\|^2 = L_M^2 + d^2$, and hence from (4.45) and (4.46) $\frac{E[h_{cp}]}{E[h_{cc}]} \geq \frac{L_M + d}{\sqrt{L_M^2 + d^2}}$ as stated in the upper part of (4.40).

Now, let us examine the case $L_m > 0$. A generic point on the boundary of $\overline{\mathcal{F}}_N$ can be written as $\overline{h} = h_c + \alpha_1 v_1 + \dots + \alpha_N v_N$, where

$$\frac{\alpha_1^2}{L_M^2} + \frac{1}{L_m^2} \sum_{i=2}^N \alpha_i^2 = 1. \quad (4.47)$$

Then,

$$\begin{aligned} \|z^* - \overline{h}\|^2 &= \|h_c - d v_N - h_c - \alpha_1 v_1 - \dots - \alpha_N v_N\|^2 \\ &= \alpha_1^2 + \alpha_2^2 + \dots + (\alpha_N + d)^2 \\ &= \alpha_1^2 \left[\frac{L_M^2 - L_m^2}{L_M^2} \right] + 2 \alpha_N d + L_m^2 + d^2 \end{aligned}$$

where the last equality has been obtained by using (4.47). Exploiting the above expression, the maximization of $\|z^* - \overline{h}\|$ with respect to $\overline{h} \in \overline{\mathcal{F}}_N$ is a straightforward exercise that leads to

$$\sup_{\overline{h} \in \overline{\mathcal{F}}_N} \|z^* - \overline{h}\| = \begin{cases} L_M \sqrt{1 + \frac{d^2}{L_M^2 - L_m^2}} & \text{if } d < \frac{L_M^2 - L_m^2}{L_m}, \\ L_m + d & \text{if } d \geq \frac{L_M^2 - L_m^2}{L_m}. \end{cases} \quad (4.48)$$

Then, (4.40) is an immediate consequence of (4.45), (4.46) and (4.48). \square

Observe that, for the ellipsoidal feasible set above, $L_M = \text{rad}(\mathcal{F}_N)$. Hence, when $L_m = 0$ one obtains the same bound as in Theorem 4.3. Conversely, when L_m tends to L_M , the ratio $E[\Phi_{cp}]/E[\Phi_{cc}]$ tends to 1, as expected, because for a spherical feasible set the conditional center with respect to any \mathcal{M} coincides with the projection of the center of the sphere onto \mathcal{M} .

4.5 Simulation examples

In this section two examples are reported. In the first example, the suboptimal pole choice performed by the central projection algorithm is compared to the optimal one (4.21), in the case of Laguerre basis functions. The optimal model $h_{cc} = \Phi_{cc}(y; p^*(\mathcal{F}_N))$ is obtained by applying the procedure presented in [19] for computing the conditional central estimate of an ellipsoidal feasible set, for each model class \mathcal{M} with fixed Laguerre pole p , and then minimizing with respect to p via a one-dimensional gridding on the interval $(-1, 1)$.

Example 4.3 *Consider the transfer function*

$$H(z) = \frac{5 + 10.7 z^{-1} + 5.002 z^{-2}}{1 + 2.3 z^{-1} + 2.06 z^{-2} + 0.72 z^{-3}}$$

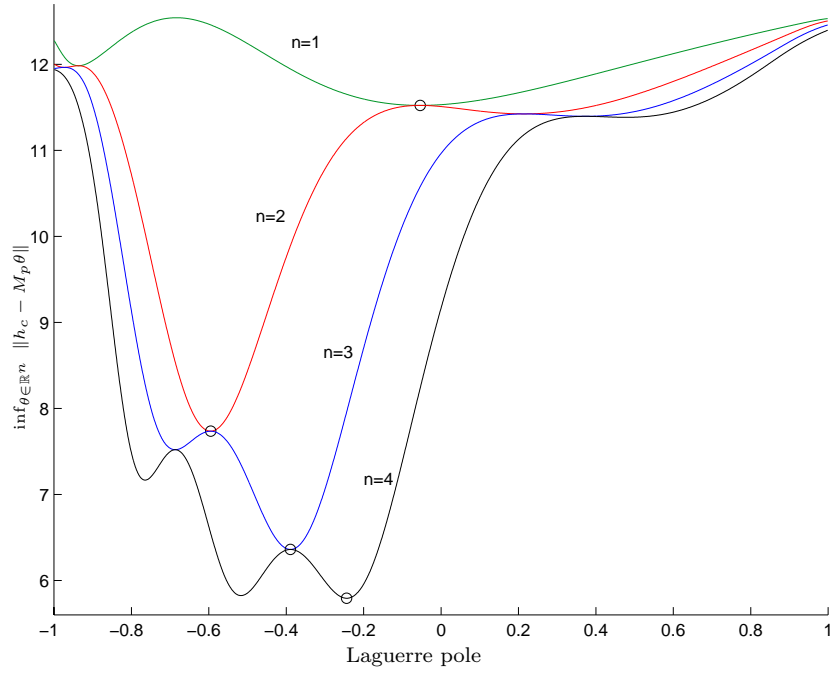
and let $N = 50$ i/o data be available, with the input u_k being a unitary step. Assume that the truncated feasible system set is given by

$$\mathcal{F}_N = \{h^N : \|y - T(u)h^N\| \leq \sqrt{N} \varepsilon\}$$

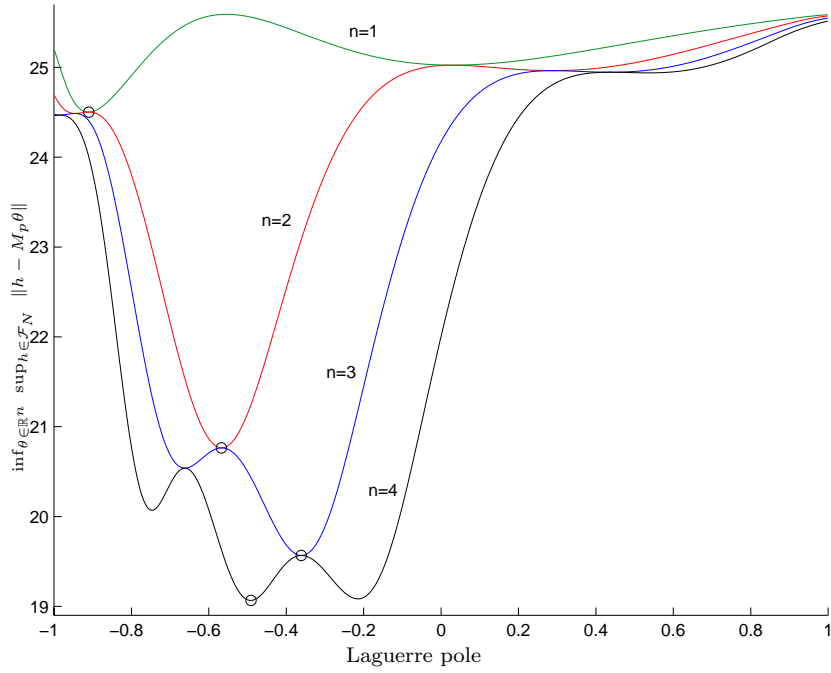
with noise $\{v_k\}_{k=0}^{N-1}$ being a Gaussian random sequence, satisfying $\frac{1}{\sqrt{N}}\|v\| \leq \varepsilon$, and $\varepsilon = 1$.

Let $h_c = T^{-1}(u)y$ and \mathcal{M} be the model class given by Laguerre filters. In Figure 4.4a, the error $\inf_{\theta \in \mathbb{R}^n} \|h_c - M_p \theta\|$ is plotted as a function of the Laguerre pole, for different model orders $n = 1, 2, 3, 4$. This error clearly does not depend on the feasible set, but just on the approximation of its center h_c . The global minimum of each curve corresponds to the pole $p^(h_c)$ for each model order n , which is the one picked by the central projection algorithm Φ_{cp} .*

These values are reported in Table 4.2, together with the associated worst-case identification errors $E[h_{cp}]$; the latter are obviously larger than the errors at the minima in Fig. 4.4a, which are computed only with respect to the center h_c and not to the whole set \mathcal{F}_N .



(a)



(b)

Figure 4.4: (a) Approximation error for the center h_c versus Laguerre pole, for different model orders; (b) Worst-case error for the conditional central algorithm versus Laguerre pole, for different model orders.

n	$p^*(h_c)$	$p^*(\mathcal{F}_N)$	$E[h_{cp}]$	$E[h_{cc}]$	$E[h_{cp}]/E[h_{cc}]$	Bound Thm. 4.6
1	-0.054	-0.910	25.0408	24.5024	1.0220	1.4069
2	-0.595	-0.567	20.8556	20.7634	1.0044	1.3573
3	-0.389	-0.361	19.6296	19.5662	1.0032	1.3223
4	-0.244	-0.491	19.1362	19.0656	1.0037	1.3045

Table 4.2: Selected poles, corresponding worst-case identification errors, actual error ratio and upper bound provided by Theorem 4.6, for different model orders.

Conversely, Figure 4.4b shows the worst-case error (with respect to \mathcal{F}_N) of the conditional Chebyshev center versus the Laguerre pole. The values of the optimal poles and the corresponding error $E[h_{cc}]$ are also reported in Table 4.2, together with the ratio between the errors of suboptimal and optimal algorithm and the upper bound provided by Theorem 4.6.

From the above example, it can be observed that the suboptimal pole $p^*(h_c)$ selected by the central projection algorithm can be quite far from the optimal one $p^*(\mathcal{F}_N)$, due to the presence of local minima whose cost is close to the global one (in Example 4.3, this happens for $n = 1$ and $n = 4$). Nevertheless, the worst-case identification error of the suboptimal algorithm turns out to be pretty close to the minimum achievable error $E[\Phi_{cc}]$, as shown by Table 4.2.

The next example shows that there exist conditional identification problems, with model class \mathcal{M} given by Laguerre expansions, for which the conditions of Theorem 4.4 are satisfied, and hence the upper bound is actually achieved.

Example 4.4 *Let \mathcal{M} be the model class of discrete Laguerre filters, defined as in (4.9).*

Let $h = l_1(-0.7746) + l_1(0.7746)$, and let us set $N = 100$ and $n = 2$. For $\bar{h} = T^N h$ there are two distinct optimal Laguerre poles, given by $p_1^ = -0.5773$ and $p_2^* = 0.5773$. The corresponding optimal parameter vectors (4.23) can be*

computed as $\theta_1^* = M_{p_1}^* \bar{h}$ and $\theta_2^* = M_{p_2}^* \bar{h}$, where $M_{p_i}^* \in \mathbb{R}^{N \times 2}$ is the truncated basis matrix of \mathcal{M} , with pole p_i^* . Let us choose $h_{cp} = M_{p_1}^* \theta_1^*$. Then, it can be verified that $\forall p$ and $\forall \tilde{\theta} \in \mathbb{R}^n$

$$\|\bar{h} - h_{cp}\| = \|\bar{h} - M_{p_2}^* \theta_2^*\| = 0.9129 < \|\bar{h} - M_p \tilde{\theta}\|$$

and $(\bar{h} - h_{cp})' (\bar{h} - M_{p_2}^* \theta_2^*) = 0$. Therefore, due to Theorem 4.4, there exists a feasible set for which $E[h_{cp}] = \sqrt{2} \cdot E[h_{cc}]$.

Indeed, let $\mathcal{F}_N = \{x \in \mathbb{R}^N : \|x - \bar{h}\| \leq 0.9129, h'(M_{p_2}^* \theta_2^* - \bar{h}) = 0\}$.

By applying the procedure in [19] for the computation of the conditional Chebyshev center and minimizing with respect to p , one gets

$$h_{cc} = \Phi_{cc}(y; p^*(\mathcal{F}_N)) = M_{p_2}^* \theta_2^*.$$

Moreover, one has $E[h_{cp}] = 1.8258$ and $E[h_{cc}] = 1.291$.

Optimal input design

In this chapter, the issue of choosing an appropriate input signal for a set-membership identification experiment is described. Different approaches will be analyzed in the set-membership framework, which depend on a priori knowledge, and on the norms used to bound the noise error and to evaluate the modelling error.

The chapter is organized as follows. In Section 5.1 an overview on general problems concerning the experiment design is reported. In Section 5.2 an approach to set-membership optimal input design is described, under the framework of the so-called time and model complexity. Separation between errors deriving from experimental data and model representation is reported. Optimal input design concerning the noise free case is also reported. In Section 5.3 some results concerning the energy bounded noise case are described.

5.1 Experiment design in system identification

Designing an identification experiment involves several choices. First of all, it is essential to have some information about the system to identify, such as input and output signals. In some cases it is not clear which signals are to be considered as inputs and which are to be considered as outputs. This means that one has to decide where sensors should be placed to acquire system outputs, and which signals should be manipulated (inputs) to drive the system during the experiment. It is worth noticing that in some cases there may be signals associated with the process which have to be considered as inputs, although it is not possible to manipulate them. In this case, however, it can be useful to place sensors to measure them (if they are measurable) and to consider them when building a model.

One more question regards the sampling time according to which measurements must be made. This is an important feature since using a sample time too large it is impossible to properly identify high frequency dynamics, while using a too small sample time may produce a useless growth of the computational burden.

Another aspect of input design involves the experiment length, that is what is the amount of information needed to reach a fixed level of estimate uncertainty. It is immediate to realize that longer experiments provide better results than shorter ones, but this happens at the expense of an increased computational burden. Of course, the kind of system to be identified may produce some restrictions about experiment length, due either to technological or economical constraints. In the following, we will consider input design experiments concerning a generic N input-output pairs.

Finally, the designed inputs must be *informative enough*, so that the resulting data set is enough informative to allow a good identification procedure.

However, also with reference to this aspect, input signal selection is not free, depending on the kind of application. Usually, in laboratory applications, there is a certain freedom for input choice, while severe constraints may be met dealing with industry processes.

In the statistical approach, it is known that an input must be *persistently exciting* to allow a good identification of a system. Typical input signals can be white noise, filtered white noise, pseudo-random binary sequences (PRBS), chirp signals, etc. For a deeper treatment about statistical input design see [41].

In the next sections, the problem of optimal input design is reported in a set-membership context.

5.2 Time and model complexity for fast identification

Let us consider discrete time systems with impulse response in $\ell_1[0, \infty)$ and let us denote the space of such systems by \mathcal{L} , equipped with a norm $\|\cdot\|_{\mathcal{L}}$. Consider the set of admissible inputs U bounded in ℓ_∞ norm. A system with impulse response k acts on an input u in the usual convolution form, i.e.

$$y(t) = \sum_{\tau=0}^{\infty} k(\tau)u(t-\tau) + v(t) \quad , \quad t = 0, 1, 2, \dots \quad (5.1)$$

Let us rewrite the previous equation in the following form

$$y(t) = (Ku)(t) + v(t) \quad , \quad t = 0, 1, 2, \dots \quad (5.2)$$

The a priori information consists in a subset \mathcal{S} of \mathcal{L} to which the true system belongs. The noise v lies in a set \mathcal{V} bounded in the ℓ_∞ norm.

The results derived in the following are based on the next assumption.

Assumption 5.1 \mathcal{S} and \mathcal{V} are convex symmetric subsets of \mathcal{L} and $\ell_\infty(-\infty, \infty)$ respectively.

This means that $k \in \mathcal{S}$ implies $-k \in \mathcal{S}$, and $v \in \mathcal{V}$ implies $-v \in \mathcal{V}$.

Given an input u , the objective of fast identification is to estimate the system to a specified accuracy in the \mathcal{L} norm as quickly as possible, from noisy observations of the output y . On the basis of the observations, $y(t_0), y(t_0 + 1), \dots, y(t_0 + T - 1)$, the location of the true kernel k_{true} is narrowed down from the a priori data set \mathcal{S} to a smaller set. Thus, the feasible system set turns out to be

$$S(y) = \{k \in \mathcal{S} : (Ku)(t) - y(t) = v(t) \quad \forall t \in [t_0, t_0 + T), \quad \forall v \in \mathcal{V}\} \quad (5.3)$$

In Chapter 3 and 4 it has been remarked that the Chebyshev center provides the optimal estimate, but it is usually difficult to compute, especially when the feasible set lies in an infinite dimensional space. For this reason, finitely parameterized model sets are usually adopted; in this case, the optimal estimation algorithm is provided by the conditional Chebyshev center (see Section 3.2).

For a chosen n -parameter model set \mathcal{M}_n , in order to study the effect of the input and the model set on the identification process, we consider the worst-case error

$$e^T(u, \mathcal{M}_n) = \sup_{k_{true} \in \mathcal{S}} \sup_{v_{true} \in \mathcal{V}} \inf_{k_{est} \in \mathcal{M}} \sup_{k \in S(y)} \|k - k_{est}\|_{\mathcal{L}}. \quad (5.4)$$

This quantity can be viewed as a sort of global error of the conditional central algorithm. To minimize such a function (i.e. to achieve a small worst-case identification error), the model set and the input have to be designed properly.

5.2.1 Separation of input design and model selection

The aim of this section is to separate input design from model class selection, in order to analyze them separately.

Definition 5.1 *Let us define as inherent error the following function*

$$\delta^T(u) = \sup\{\|k\|_{\mathcal{L}} : k \in \mathcal{S}, (Ku)(t) = v(t), v \in \mathcal{V}, \forall t \in [t_0, t_0 + T]\} \quad (5.5)$$

Note that the inherent error can be viewed as a global error for $y = 0$, and depends only on the input.

Definition 5.2 *Let us define as representation error the following function*

$$\text{dist}(\mathcal{S}, \mathcal{M}_n) = \sup_{k \in \mathcal{S}} \inf_{g \in \mathcal{M}_n} \|k - g\|_{\mathcal{L}} \quad (5.6)$$

Note that this function depends only on the model class \mathcal{M}_n .

The next proposition gives upper and lower bounds of $e^T(u, \mathcal{M}_n)$ in terms of data and representation errors [42].

Proposition 5.1 *Under Assumption 5.1, it follows that*

$$\max\{\delta^T(u), \text{dist}(\mathcal{S}, \mathcal{M}_n)\} \leq e^T(u, \mathcal{M}_n) \leq 3 \max\{\delta^T(u), \text{dist}(\mathcal{S}, \mathcal{M}_n)\} \quad (5.7)$$

Remark 5.1 *Proposition 5.1 states that, once an input u_0 and a model set $\mathcal{M}_{n,0}$ have been chosen to minimize the inherent error and the representation error respectively, then the worst-case uncertainty $e^T(u_0, \mathcal{M}_{n,0})$ is within a factor of three the optimal one.*

Remark 5.2 *The worst-case identification error can be decomposed into two terms, the inherent error and representation error. The inherent error is generated in the information process, due to lack of data and inaccurate measurements, and is irreducible no matter which identification algorithm is used. The representation error is due to inaccurate representation of the a priori uncertainty set. It represents the loss of information during the information processing stage.*

5.2.2 Noise free optimal input design

It is known that between the two sources of the inherent error, that is lack of data and measurement noise, it is the former which puts the more severe constraint on fast identification. The measurement noise can be overcome by increasing the power of the input, a measure feasible on a short time interval. On the other hand, more data can only be obtained by prolonging the observation interval, that is by slowing down the identification process.

To isolate the effect of lack of data, in this section the optimal input design problem in the noise free case is addressed [43]. In the following definition, the output observation interval $[t_0, t_0 + n)$ is viewed as being fixed and the input as being optimized over this interval.

Definition 5.3 *Let us define as identification n -width (or time-width) of \mathcal{S} the following quantity*

$$\theta^n(\mathcal{S}, \mathcal{L}) = \inf_{u \in U} \sup\{\|k\|_{\mathcal{L}} : k \in \mathcal{S}, (Ku)(t) = 0, t \in [t_0, t_0 + n)\} \quad (5.8)$$

It is useful to remark that an optimal input is one for which the infimum (5.8) is attained. Moreover, since the systems in \mathcal{S} are time invariant, t_0 can be fixed at 0 without loss of generality.

In (5.8) it is not excluded the possibility that an optimal input might start prior to the observation interval. Motivation about such an assumption can be found in [42, 43].

The identification n -width characterizes the *time complexity* of the data acquisition process in an identification problem. The inverse of the identification n -width function is the least time needed to reduce the inherent error to a certain predetermined level.

Let us introduce the concept of Gel'fand n -width.

Definition 5.4 *Let \mathcal{L} be a normed linear space and \mathcal{S} a subset of \mathcal{L} . The Gel'fand n -width of \mathcal{S} in \mathcal{L} is given by*

$$d^n(\mathcal{S}, \mathcal{L}) = \inf_{L^n} \sup_{k \in \mathcal{S} \cap L^n} \|k\|_{\mathcal{L}} \quad (5.9)$$

where the infimum is taken over all subspaces L^n of \mathcal{L} of codimension n . A subspace is said to be of codimension n if there exist n independent bounded linear functionals f_1, \dots, f_n such that $L^n = \{k \in \mathcal{L} : f_i(k) = 0, i = 1, \dots, n\}$.

It has been shown that under a mild condition, the identification n -width $\theta^n(\mathcal{S}, \mathcal{L})$ is bounded below by the Gel-fand n -width $d^n(\mathcal{S}, \mathcal{L})$.

Note that the Gel'fand n -width can be seen as the optimized worst-case uncertainty when identification is based on n arbitrary linear measurements, whereas in the case of the n -width θ^n these measurements are restricted to be n consecutive output values. The inverse of the Gel'fand n -width is the least number of measurements needed to reduce the uncertainty to a predetermined value.

It will be shown in the next subsection that in many important cases the identification n -width equals the Gel'fand n -width.

5.2.3 Optimal affine representation

Since the model set optimization over all n -parameter model is a difficult task, in this section we restrict to choose among affine models, i.e. finite dimensional subspaces of \mathcal{L} .

By Definition 5.2, the minimum representation error of \mathcal{S} by an n -dimensional subspace is

$$d_n(\mathcal{S}, \mathcal{L}) = \inf_{\mathcal{M}_n \subset \mathcal{L}} \text{dist}(\mathcal{S}, \mathcal{M}_n) \quad (5.10)$$

This function is known as the *Kolmogorov n -width* of the a priori uncertainty set \mathcal{S} .

Definition 5.5 *The Kolmogorov n -width of \mathcal{S} in \mathcal{L} is given by*

$$d_n(\mathcal{S}, \mathcal{L}) = \inf_{\mathcal{M}_n} \sup_{k \in \mathcal{S}} \inf_{g \in \mathcal{M}_n} \|k - g\|_{\mathcal{L}} \quad (5.11)$$

where the infimum is taken over all n -dimensional subspaces of \mathcal{L} . If the infimum in (5.11) is achieved by some subspace \mathcal{M}_n of dimension at most n , then \mathcal{M}_n is said to be an optimal subspace for $d^n(\mathcal{S}, \mathcal{L})$.

The Kolmogorov n -width characterizes the representation complexity of an identification problem. The inverse function of d_n was called the *metric dimension* function and viewed as an appropriate measure of *metric complexity* of uncertain sets in feedback systems (see [44]). It is the least dimension needed to represent the a priori data set \mathcal{S} within a given tolerance by linear subspaces.

Remark 5.3 *It is worthwhile to remark that each of the three notions of n -width previously reported (Kolmogorov, Gel'fand and identification) describe the complexity of a distinct aspect of the identification problem only. Unfortunately, none of them describes the complexity of an identification problem completely. However, it will be shown that in many special cases they coincide.*

In the next proposition, upper and lower bounds of the optimal worst-case uncertainty in terms of identification n -width and Kolmogorov n -width are derived.

Proposition 5.2 *Under Assumption 5.1, the optimal noise free worst-case uncertainty has the following lower and upper bounds,*

$$\max\{\theta^T(\mathcal{S}, \mathcal{L}), d_n(\mathcal{S}, \mathcal{L})\} \leq \inf_{u \in U} \inf_{\mathcal{M}_n \subset \mathcal{L}} e^T(u, \mathcal{M}_n) \leq 3 \max\{\theta^T(\mathcal{S}, \mathcal{L}), d_n(\mathcal{S}, \mathcal{L})\} \quad (5.12)$$

If $d^T(\mathcal{S}, \mathcal{L}) \leq \theta^T(\mathcal{S}, \mathcal{L})$, then the optimal worst-case uncertainty is bounded below by $\max\{d^T(\mathcal{S}, \mathcal{L}), d_n(\mathcal{S}, \mathcal{L})\}$. It has been proved that for a priori sets

with certain property of monotone decrease, the maximum of the two n -widths is also an upper bound, that is

$$\inf_{u \in U} \inf_{\mathcal{M}_n \in \mathcal{L}} e^T(u, \mathcal{M}_n) = \max\{d^T(\mathcal{S}, \mathcal{L}), d_n(\mathcal{S}, \mathcal{L})\} \quad (5.13)$$

Moreover, for such data sets, the three n -width coincide and therefore either one can be used in (5.13).

In the following, some typical cases satisfying the previously described monotone property are shown.

Let $\mathcal{L} = \ell_1[0, \infty)$, $n > 0$, $C > 0$ and $0 < r < 1$. It can be shown that

1. if

$$\mathcal{S} = \{k \in \mathcal{L} : |k(\tau)| \leq Cr^\tau, \quad r = 0, 1, 2, \dots\} \quad (5.14)$$

then

$$d_n(\mathcal{S}, \mathcal{L}) = d^n(\mathcal{S}, \mathcal{L}) = \theta^n(\mathcal{S}, \mathcal{L}) = \frac{C}{1-r} r^n \quad (5.15)$$

2. if

$$\mathcal{S} = \left\{ k \in \mathcal{L} : \sum_{\tau=0}^{\infty} |k(\tau)| r^{-\tau} \leq C \right\} \quad (5.16)$$

then

$$d_n(\mathcal{S}, \mathcal{L}) = d^n(\mathcal{S}, \mathcal{L}) = \theta^n(\mathcal{S}, \mathcal{L}) = Cr^n \quad (5.17)$$

Other cases can be found in [42, 43]. The following theorem shows that under some hypotheses (satisfied by the previously described a priori sets) the optimal input results to be the impulse.

Theorem 5.1 *Under Assumption 5.1 and other technical conditions (see [42]), the optimal input (in the noise free case) is an impulse applied at the start of the observation interval, and the optimal affine model set is the FIR model $\mathcal{M}_n = \text{span}\{1, z, \dots, z^{n-1}\}$.*

Further details about the topics reported in this section can be found in [42, 43].

5.3 Optimal input for energy bounded noise

In this section the problem of finding the optimal input for an identification experiment in the energy bounded noise case is tackled. This approach differentiates from that described in Section 5.2 essentially for the following aspects:

- errors are supposed to be present (no error free case) and they are bounded in the ℓ_2 norm (energy bounded noise);
- inputs are bounded in ℓ_∞ norm;
- the number of measurements (or the time length of the experiment) is fixed and the system impulse response is negligible for $k > N$. This means that we can tackle the problem in finite dimensional spaces;
- the linear manifold \mathcal{M} is chosen a priori, based on the knowledge on the system.

5.3.1 Problem formulation

This approach recalls the framework introduced in Section 3.3.1. Let the model equation be given by

$$y = Uh + e \tag{5.18}$$

where

- $y \in \mathbb{R}^N$ contains the N measured outputs.
- $U \in \mathbb{R}^{N \times N}$ be the Toeplitz lower triangular matrix obtained by the input vector $u \in \mathbb{R}^N$ as in (1.25).

The input u is constrained to be less than a specified value in ℓ_∞ norm. This is a common constraint essentially due to input saturations. Then $\|u\|_\infty \leq \delta$.

Let \mathcal{S} be a set containing the a priori information about the system, such that $h \in \mathcal{S} \subset \mathbb{R}^N$. Let the error $e \in \mathbb{R}^N$ be energy bounded, that is $\|e\|_2 \leq \varepsilon$.

Let us define the following quantities:

- $\overline{\mathcal{U}} = \{U : U = \text{toeplitz}\{u\}, u \in \mathbb{R}^N, \|u\|_\infty \leq \delta\}$ is the set containing all feasible input matrices.
- $\overline{\mathcal{B}}_\varepsilon = \{e \in \mathbb{R}^N : \|e\|_2 \leq \varepsilon\}$ is the ℓ_2 -ball containing all possible errors.
- $\mathcal{E}(c, Q)$ is a generic ellipsoid of center c and shape Q .
- $\mathcal{M} \equiv \mathcal{M}_n = \{h \in \mathbb{R}^N : h = M\theta, M \in \mathbb{R}^{N \times n}, \theta \in \mathbb{R}^n\} \subset \mathbb{R}^N$ is a generic n -dimensional subspace of \mathbb{R}^N ($n < N$), generated by an orthonormal matrix $M \in \mathbb{R}^{N \times n}$.

By definition of feasible set, it follows that

$$\mathcal{F} = FSS_y = \{h \in \mathcal{S} : \|Uh - y\|_2 \leq \varepsilon\} \quad (5.19)$$

The inequality $\|Uh - y\|_2 \leq \varepsilon$ can be rewritten as

$$(h - U^{-1}y)'(U'U)(h - U^{-1}y) \leq \varepsilon^2 \quad (5.20)$$

Equation (5.20) describes an N -dimensional ellipsoid with

- center $c = U^{-1}y = U^{-1}(Uh + e) = h + U^{-1}e$;
- shape $Q = \frac{U'U}{\varepsilon^2}$.

According to the a priori information \mathcal{S} and $\overline{\mathcal{B}}_\varepsilon$ (concerning h and e respectively), one can introduce the set of admissible ellipsoid centers as

$$\overline{\mathcal{C}} = \mathcal{S} + U^{-1}\overline{\mathcal{B}}_\varepsilon.$$

Then, the feasible set results

$$\mathcal{F} = \mathcal{E}(c, Q) \bigcap \mathcal{S} \quad , \quad c \in \overline{\mathcal{C}} \quad \text{and} \quad Q = \frac{U'U}{\varepsilon^2} \quad (5.21)$$

The problem to be solved consists in finding the optimal input \bar{U}^* such that:

$$\bar{U}^*(\bar{\mathcal{U}}, \varepsilon, \mathcal{S}, \mathcal{M}) = \arg \inf_{U \in \bar{\mathcal{U}}} E^*(U, \varepsilon, \mathcal{S}, \mathcal{M}) \quad (5.22)$$

where

$$E^*(U, \varepsilon, \mathcal{S}, \mathcal{M}) = \sup_{c \in \bar{\mathcal{C}}} \inf_{g \in \mathcal{M}} \sup_{h \in \mathcal{E}(c, Q) \cap \mathcal{S}} \|h - g\|_H \quad (5.23)$$

and $\|\cdot\|_H$ can be $\ell_1, \ell_2, \ell_\infty$.

Now, it will be shown how equations (5.22)-(5.23) can be simplified. The following lemma states that it is not necessary that the center of the ellipsoid lies in $\bar{\mathcal{C}}$ in order to obtain the error E^* .

Lemma 5.1 *An optimal solution of (5.23) is always achieved by some $c \in \mathcal{S}$, that is*

$$\begin{aligned} E^*(U, \varepsilon, \mathcal{S}, \mathcal{M}) &= \sup_{c \in \bar{\mathcal{C}}} \inf_{g \in \mathcal{M}} \sup_{h \in \mathcal{E}(c, Q) \cap \mathcal{S}} \|h - g\|_H \\ &= \sup_{c \in \mathcal{S}} \inf_{g \in \mathcal{M}} \sup_{h \in \mathcal{E}(c, Q) \cap \mathcal{S}} \|h - g\|_H \end{aligned} \quad (5.24)$$

Proof. It is straightforward to note that if $\mathcal{E}_1 = \mathcal{E}(c_1, Q)$, $c_1 \notin \mathcal{S}$, then there exists $\mathcal{E}_2 = \mathcal{E}(c_2, Q)$, $c_2 \in \mathcal{S}$ such that $(\mathcal{E}_1 \cap \mathcal{S}) \subseteq (\mathcal{E}_2 \cap \mathcal{S})$. In fact, let $v \in (\mathcal{E}_1 \cap \mathcal{S})$, one has $\|v - c_1\|_Q \leq 1 \triangleq (v - c_1)'Q^{-1}(v - c_1) \leq 1$. This imply that there exists $\alpha \in [0, 1]$ such that $c_2 = \alpha v + (1 - \alpha)c_1$, $c_2 \in \partial\mathcal{S}$. It follows that: $\|v - c_2\|_Q = \|v - \alpha v - (1 - \alpha)c_1\|_Q = \|(1 - \alpha)(v - c_1)\|_Q = (1 - \alpha)\|v - c_1\|_Q \leq 1$ and the lemma is proved. \square

Lemma 5.2 *Let $\bar{U} = \text{toeplitz}\{\bar{u}\}$ an optimal solution of (5.22) such that $\|\bar{u}\| < \delta$. Then another optimal solution of (5.22) is $\tilde{U} = \text{toeplitz}\{\tilde{u}\}$, where $\tilde{u} = \frac{\bar{u}}{\|\bar{u}\|} \delta$.*

Proof. Let $\|\bar{u}\| = \alpha < \delta$. W.l.o.g. we can suppose $\varepsilon = 1$.

Let $\bar{V}\bar{Q}\bar{V}' = \text{SVD}^1(\bar{U}'\bar{U})$ and $\tilde{V}\tilde{Q}\tilde{V}' = \text{SVD}(\tilde{U}'\tilde{U})$. Since $\tilde{U} = \frac{\delta}{\alpha}\bar{U}$, it

¹SVD=Singular Value Decomposition

follows that $\tilde{V} \tilde{Q} \tilde{V}' = SVD(\frac{\delta^2}{\alpha^2} \bar{U}' \bar{U}) = \frac{\delta^2}{\alpha^2} SVD(\bar{U}' \bar{U})$. This means that $\tilde{V} = \bar{V}$ and $\tilde{Q} = \frac{\delta^2}{\alpha^2} \bar{Q}$. Let $\bar{\mathcal{E}} = \mathcal{E}(c, \bar{U}' \bar{U})$ and $\tilde{\mathcal{E}} = \mathcal{E}(c, \tilde{U}' \tilde{U})$, $c \in \mathbb{R}^N$; it follows that the two ellipsoids have the same axes orientations (since $\tilde{V} = \bar{V}$), but $\tilde{\mathcal{E}}$ is shrunk with respect to $\bar{\mathcal{E}}$, because every semiaxes is reduced by a factor $\frac{\alpha}{\delta}$. So it follows that $\tilde{\mathcal{E}} \subset \bar{\mathcal{E}}$, and since \bar{U} is an optimal solution of (5.22) also \tilde{U} is optimal and the lemma is proofed. \square

Indeed, the previous lemma states that to find a solution of (5.22) it is sufficient to consider only inputs such that $\|u\| = \delta$.

Due to Lemma 5.1 it is possible to rewrite (5.23) as in (5.24). Since the parameter ε affects only the shape of the ellipsoid Q , (while it does not affect the center c), it is possible to rewrite the problem in the following way: let $\eta = \frac{\delta}{\varepsilon}$ and let $\mathbb{U} = \{U : U = \text{toeplitz}\{u\}, u \in \mathbb{R}^N, \|u\| = \eta\}$. It is easy to note that, if $\bar{U} \in \bar{\mathbb{U}}$ ($\|u\| = \delta$) and $U = \bar{U}/\varepsilon$, then $\mathcal{E}(c, \frac{\bar{U}' \bar{U}}{\varepsilon^2}) = \mathcal{E}(c, U' U)$, $\forall c \in \mathbb{R}^N$.

Hence, equations (5.22) and (5.23) can be rewritten as follows.

Let $\mathbb{U} = \{U : U = \text{toeplitz}\{u\}, u \in \mathbb{R}^N, \|u\| = \eta\}$, then:

$$\bar{U}^*(\bar{\mathbb{U}}, \varepsilon, \mathcal{S}, \mathcal{M}) = \varepsilon U^*(\mathbb{U}, \mathcal{S}, \mathcal{M}) \quad (5.25)$$

where

$$U^*(\mathbb{U}, \mathcal{S}, \mathcal{M}) = \arg \inf_{U \in \mathbb{U}} E^*(U, \mathcal{S}, \mathcal{M}) \quad (5.26)$$

and

$$E^*(U, \mathcal{S}, \mathcal{M}) = \sup_{c \in \mathcal{S}} \inf_{g \in \mathcal{M}} \sup_{h \in \mathcal{E}(c, U' U) \cap \mathcal{S}} \|h - g\|_H. \quad (5.27)$$

Remark 5.4 *It follows that if U^* is a solution of (5.26) (and hence $\bar{U}^* = \varepsilon U^*$ is solution of (5.25)), the optimal input is given by*

$$u^* = U^*[\varepsilon, 0, \dots, 0]'$$

5.3.2 Evaluation of worst-case error bounds

Let us now give an assumption that will hold in the following development.

Assumption 5.2 \mathcal{S} is an orthotope centered at the origin.

The set \mathcal{S} can be described as

$$\mathcal{S} = \{h = [h_0, \dots, h_{N-1}]' : |h_0| \leq \gamma_0, |h_1| \leq \gamma_1, \dots, |h_{N-1}| \leq \gamma_{N-1}\}. \quad (5.28)$$

It is easy to show that several kinds of a priori information can be formulated in this way, such as FIRs of order N with exponential decay response as

$$\mathcal{S} = \{h : |h_i| \leq M\rho^i, M > 0, |\rho| < 1, i = 0, \dots, N-1\}. \quad (5.29)$$

Let \mathcal{S} and \mathcal{M} be fixed. Then the following proposition holds.

Proposition 5.3 *Let*

$$\overline{E} = \inf_{g \in \mathcal{M}} \sup_{h \in \mathcal{S}} \|h - g\|_H \quad (5.30)$$

and

$$\underline{E} = \sup_{h \in \mathcal{S}} \inf_{g \in \mathcal{M}} \|h - g\|_H. \quad (5.31)$$

Then,

$$\underline{E} \leq E^*(U, \varepsilon, \mathcal{S}, \mathcal{M}) \leq \overline{E} \quad (5.32)$$

Proof. If $|u_0| = \infty$ the ellipsoid \mathcal{E} collapses into a point (its center), and also the feasible set collapses into a point. Then, this is the condition for which FSS_y is the smallest one (it cannot be empty if a priori assumptions are correct) and one has

$$\begin{aligned} E^*(|u_0| = \infty, \varepsilon, \mathcal{S}, \mathcal{M}) &= \sup_{c \in \mathcal{C}} \inf_{g \in \mathcal{M}} \sup_{h \in (c \cap \mathcal{S})} \|h - g\|_H = \\ &= \sup_{c \in \mathcal{S}} \inf_{g \in \mathcal{M}} \sup_{h=c} \|h - g\|_H = \sup_{c \in \mathcal{S}} \inf_{g \in \mathcal{M}} \|g - c\|_H = \underline{E}. \end{aligned}$$

If $\|u\| \rightarrow 0$ (or in general $\varepsilon \cap \mathcal{S} \equiv \mathcal{S}$):

$$E^*(\|u\| \rightarrow 0, \varepsilon, \mathcal{S}, \mathcal{M}) = \sup_{c \in \bar{\mathcal{C}}} \inf_{g \in \mathcal{M}} \sup_{h \in \mathcal{S}} \|h - g\|_H = \inf_{g \in \mathcal{M}} \sup_{h \in \mathcal{S}} \|h - g\|_H = \bar{E}.$$

□

An illustrative example showing the two bounds for the case $\|\cdot\|_H = \ell_2$ is depicted in Figure 5.1.

Remark 5.5 Note the \underline{E} coincides with the so-called representation error described in Definition 5.2, while \bar{E} is the conditional radius of the set \mathcal{S} on the subspace \mathcal{M} (see Section 3.2).

Remark 5.6 Note that \underline{E} and \bar{E} can be quite far. However, if they coincide, this means that all inputs are equivalent. An example of such a behaviour is reported in Figure 5.2. Moreover, in this example, it is straightforward to note that also all one dimensional subspaces are equivalent.

The following example shows that, under Assumption 5.2, impulse input may not be the optimal one (contrary to what stated in Theorem 5.1 for the minimization of the inherent error in the noise free case).

Example 5.1 Let $N = 2$, $\delta = 1$, $\varepsilon = 1$ and $\|\cdot\|_H = \ell_2$. Let \mathcal{M} be a FIR of order 1 and let $\mathcal{S} = \{h \in \mathbb{R}^2 : |h_0| \leq 1, |h_1| \leq 0.2\}$ be the set containing the a priori information on the impulse response. Let $u_i = [1 \ 0]'$ the impulse input and $u_s = [1 \ 1]'$ the step input. In Figure 5.3 the worst-case errors E_i^* and E_s^* are represented for impulse and step input respectively, and it can be seen that the impulse input is not the best one. Indeed: $E_i^* = \sqrt{(1^2 + 0.2^2)} = 1.0198 = \bar{E}$ whereas $E_s^* = \sqrt{0.8^2 + 0.2^2} = 0.8246$.

In the following propositions, the bounds \bar{E} and \underline{E} are evaluated.

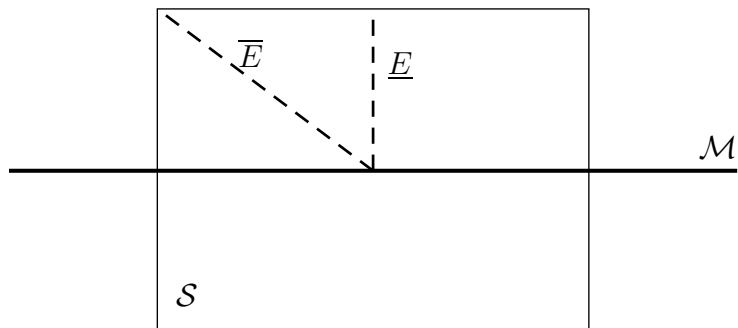


Figure 5.1: Example ($N = 2$, $n = 1$, $\|\cdot\|_H = \ell_2$) showing the upper and lower bounds.

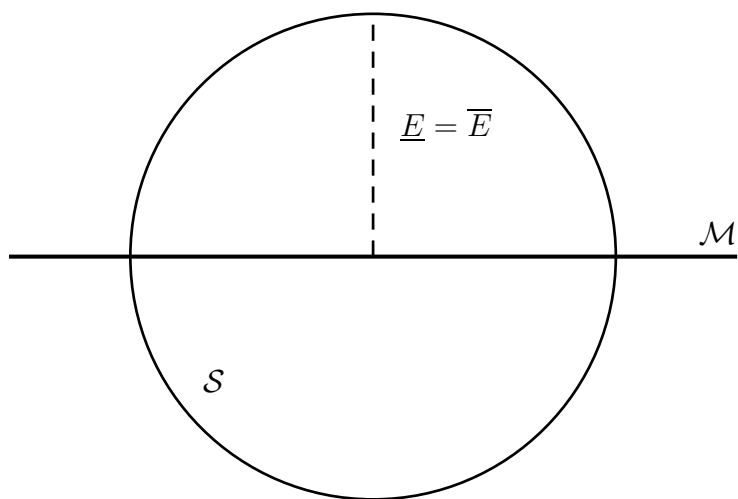


Figure 5.2: Example ($N = 2$, $n = 1$, $\|\cdot\|_H = \ell_2$) in which the upper and lower bounds coincide.

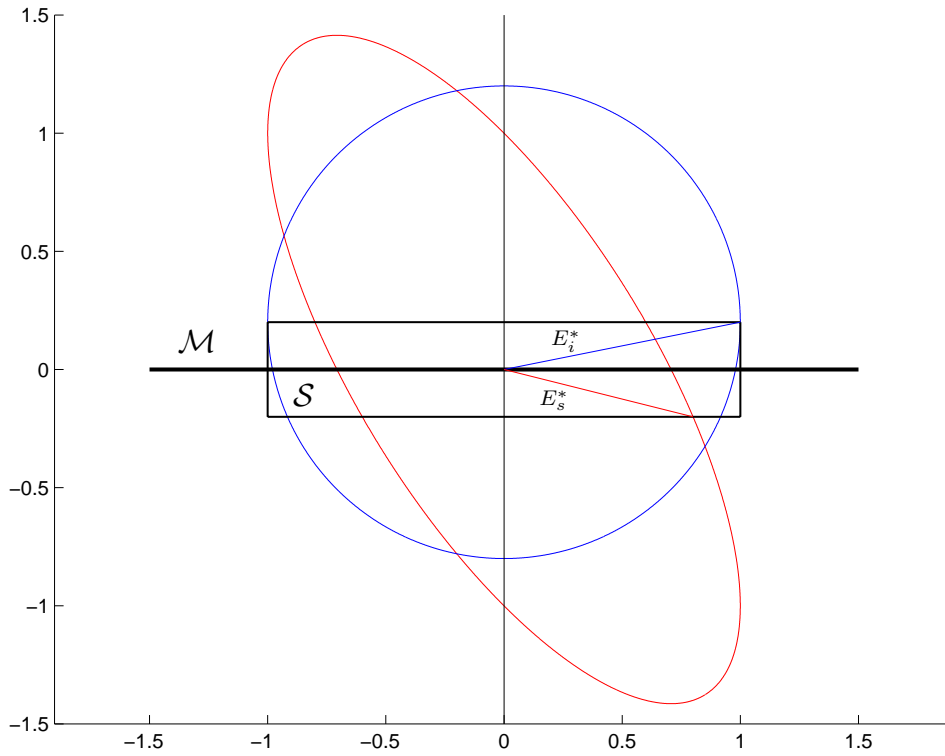


Figure 5.3: Example in which the step input is better than the impulse input.

Proposition 5.4 *Let \mathcal{S} be defined as in (5.28) and let v be any of its vertexes. Then*

$$\overline{E} = \|v\|_H$$

Proof. Since \overline{E} is the conditional radius of \mathcal{S} with respect to \mathcal{M} , and since \mathcal{M} is a subspace which contains the Chebyshev center of \mathcal{S} (origin), it follows that \overline{E} is the Chebyshev radius of \mathcal{S} , and the conditional Chebyshev center coincides with the origin. \square

Proposition 5.5 *Let \mathcal{S} be defined as in (5.28) and let V be the set containing all the vertexes of \mathcal{S} , then*

$$\underline{E} = \sup_{v \in V} \inf_{g \in \mathcal{M}} \|v - g\|_H. \quad (5.33)$$

Moreover, if $\|\cdot\|_H = \ell_2$ one has

$$\underline{E} = \sup_{v \in V} \|(I - M'M)v\|_H \quad (5.34)$$

Proof. Equation (5.33) follows immediately by the fact that \mathcal{S} is a orthotope.

If $\|\cdot\|_H = \ell_2$, by the projection theorem one obtains:

$$\underline{E} = \sup_{h \in \mathcal{S}} \inf_{g \in \mathcal{M}} \|h - g\|_H = \sup_{h \in \mathcal{S}} \|h - MM'h\|_H = \sup_{h \in \mathcal{S}} \|(I - MM')h\|_H$$

Moreover, being \mathcal{S} an orthotope, we have:

$$\sup_{h \in \mathcal{S}} \|(I - MM')h\|_H = \sup_{v \in V} \|(I - MM')v\|_H$$

and the proposition is proofed. \square

The derivation of tighter bounds and the computation of the optimality level for certain classes of input signals is currently under development.

Example of application

In this chapter an example of application of set-membership and statistical identification techniques is reported. Given a physical process (DC motor) it will be shown how to derive suitable models through system identification techniques.

The chapter is organized as follows. In Section 6.1 a description of the physical model to be identified is shown, while in Section 6.2 the identification results obtained by applying some deterministic and statistic identification algorithms are addressed. In particular, one set estimator (minimum outer box) and two point estimators (central and projection algorithms) are used for set-membership identification, whereas the standard least squares algorithm is used in statistical identification.

6.1 Process description

The process to be identified is a DC motor, where motor voltage is assumed as input, while the output is the angular velocity. The process is one of those

connected to the *Automatic Control Telelab* (ACT), a remote laboratory of automatic control developed at the University of Siena which allows the remote control and identification of physical systems. For a deeper treatment about ACT see Part 2. All the results reported in the following have been obtained by identification experiments performed through the Internet.

The process consists of a DC motor, a reduction unit, a tachometer and a visualization system. A picture of the process is reported in Fig. 6.1 while a scheme is illustrated in Fig. 6.2. The DC motor (A) generates the actuating signal for the speed control system and the reducer (B) reduces the angular velocity. Reducer consists in two cogwheel performing a reduction of 50 : 1.

The process is affected by two nonlinearities. The first one is given by a saturation on the input command, which is constrained to belong to the interval $[-5, +5]$ Volts. These values are below the nominal voltage and have been enforced for security reasons. The second nonlinearity consists in a threshold on the output due to the presence of Coulomb friction.

6.2 Identification procedure

In this section the identification procedure used is described. The model class used to identify the system is the ARX class. An ARX model is given by

$$y_k = \sum_{i=1}^{n_a} a_i y_{k-i} + \sum_{i=1}^{n_b} b_i u_{k-i} + e_k \quad (6.1)$$

where u_k is the input, y_k is the output and e_k is the noise. The aim of identification is to estimate the parameters a_i, b_i on the base of the knowledge of the inputs $\{u_1, \dots, u_N\}$ and of the measurements $\{y_1, \dots, y_N\}$.

The parameters will be estimated assuming either that the noise is UBB (set-membership framework) or that it is a white noise with zero mean (statistical identification).

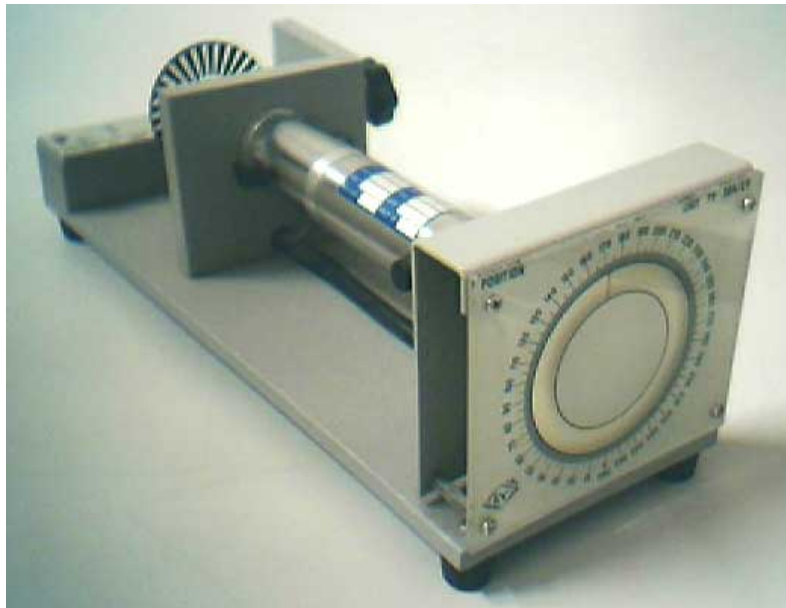


Figure 6.1: Picture of the DC motor.

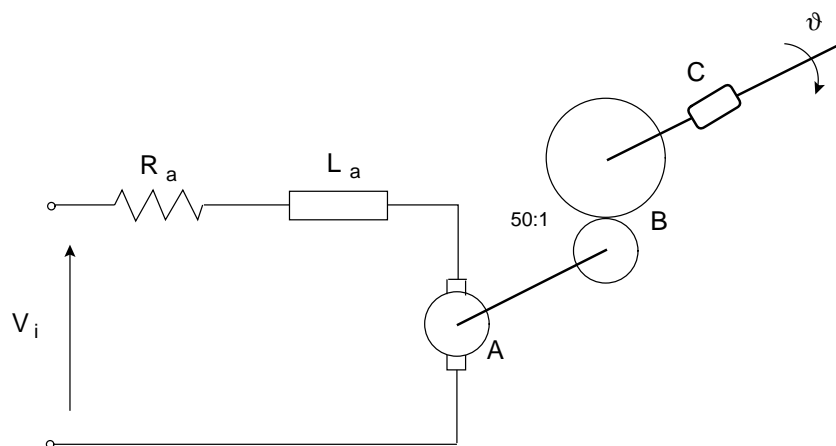


Figure 6.2: Electric circuit of the DC motor.

6.2.1 Set-membership identification

This subsection presents the identification results obtained assuming that the noise is bounded in ℓ_∞ norm, i.e.

$$|e_k| \leq \varepsilon \quad \forall k. \quad (6.2)$$

The problem can be described in the *Information Based Complexity* formalism introduced in Chapter 1 as follows:

- X is the space of unknown parameters (of dimension $n_a + n_b$) whose generic element is

$$x = [a_1, \dots, a_{n_a}, b_1, \dots, b_{n_b}]'.$$

- Y is the measurements space (of dimension $N - n_a$) with elements

$$y = [y_{n_a+1}, \dots, y_N]'$$

and the noise vector $e \in Y$ defined as $e = [e_{n_a+1}, \dots, e_N]'$ is such that

$$\|e\|_\infty \leq \varepsilon.$$

- The solution space coincides with the parameter space, that is $Z = X$ because $S(x) = x$. Moreover, since we assume that no further a priori information is available on unknown parameters, we have $K \equiv X$.

The information operator is given by

$$F(x) = \begin{pmatrix} \phi_{n_a} \\ \phi_{n_a+1} \\ \vdots \\ \phi_{N-1} \end{pmatrix} x$$

where

$$\phi_k = [y_k, \dots, y_{k-n_a+1}, u_k, \dots, u_{k+1-n_b}] \quad (6.3)$$

is the regressor vector. Since $F(x)$ is linear, FPS_y is a convex polytope.

In the following, a set estimator and two point estimators are computed.

Minimum outer box

Computing the minimum box containing the feasible set requires the solution of $2n$ linear programming problems, given by

$$\begin{cases} \sup x_i \\ \inf x_i \end{cases}, \quad i = 1, \dots, N \quad (6.4)$$

subject to

$$\|y - Fx\|_\infty \leq \varepsilon \quad (6.5)$$

that is

$$\begin{bmatrix} F \\ -F \end{bmatrix} x \leq \begin{bmatrix} y + \varepsilon \\ -y + \varepsilon \end{bmatrix}. \quad (6.6)$$

Central algorithm in ℓ_∞ norm

To obtain a pointwise estimate, and in particular the central algorithm in ℓ_∞ norm, it suffices to compute the geometric center of the previously determined box, as stated in Theorem 2.2.

Projection algorithm in ℓ_∞ norm

Another estimate can be computed by the projection algorithm, previously introduced in Definition 1.12. In this case, it is requested to solve one linear programming problem, i.e.

$$x_p = \arg \min_{x \in X} \|y - Fx\|_\infty \quad (6.7)$$

which equals to

$$[x, \varepsilon]' = \arg \min_{x, \varepsilon} \varepsilon \quad \text{subject to} \quad \|y - Fx\|_\infty \leq \varepsilon \quad (6.8)$$

when the constraint is equivalent to:

$$\begin{bmatrix} F & -1 \\ -F & -1 \end{bmatrix} \begin{bmatrix} x \\ \varepsilon \end{bmatrix} \leq \begin{bmatrix} y \\ -y \end{bmatrix}. \quad (6.9)$$

Note that the projection estimate does not depend on ε .

Experimental results

In the following, the results of the previously described identification algorithms applied to the DC motor process are reported. Six different ARX models have been estimated. Equation (6.1) has been rearranged as follows:

$$Ay = Bu + e \quad (6.10)$$

where

$$A = [1 \ a_1 \ a_2 \ \dots \ a_{n_a}] \quad , \quad B = [0 \ b_1 \ b_2 \ \dots \ b_{n_b}]$$

and

$$y = [y_k \ y_{k-1} \ \dots \ y_{k-n_a}]' \quad , \quad u = [u_k \ u_{k-1} \ \dots \ u_{k-n_b}]' \quad , \quad e = e_k$$

where $k = \max\{n_a, n_b\} + 1, \dots, N$.

The experiment has been performed using as input a Pseudo Random Binary Sequence (PRBS) with maximum and minimum value 5 and -5 , respectively. The experiment lasted 30 seconds and the sample time was set to 0.01 second. The time plot regarding input/output data is shown in Figure 6.3.

The first 2000 samples recorded were used for estimation while the following 1000 were used for validation of the obtained model. In particular, as validation index, the FIT on the validation data has been computed (for a deeper

treatment see [41]). The FIT index is given by

$$FIT = \frac{1 - \|\hat{y} - y\|_2}{\|y - \bar{y}\|_2} \cdot 100 \quad (6.11)$$

where \hat{y} is the estimated output while \bar{y} is the output sample mean.

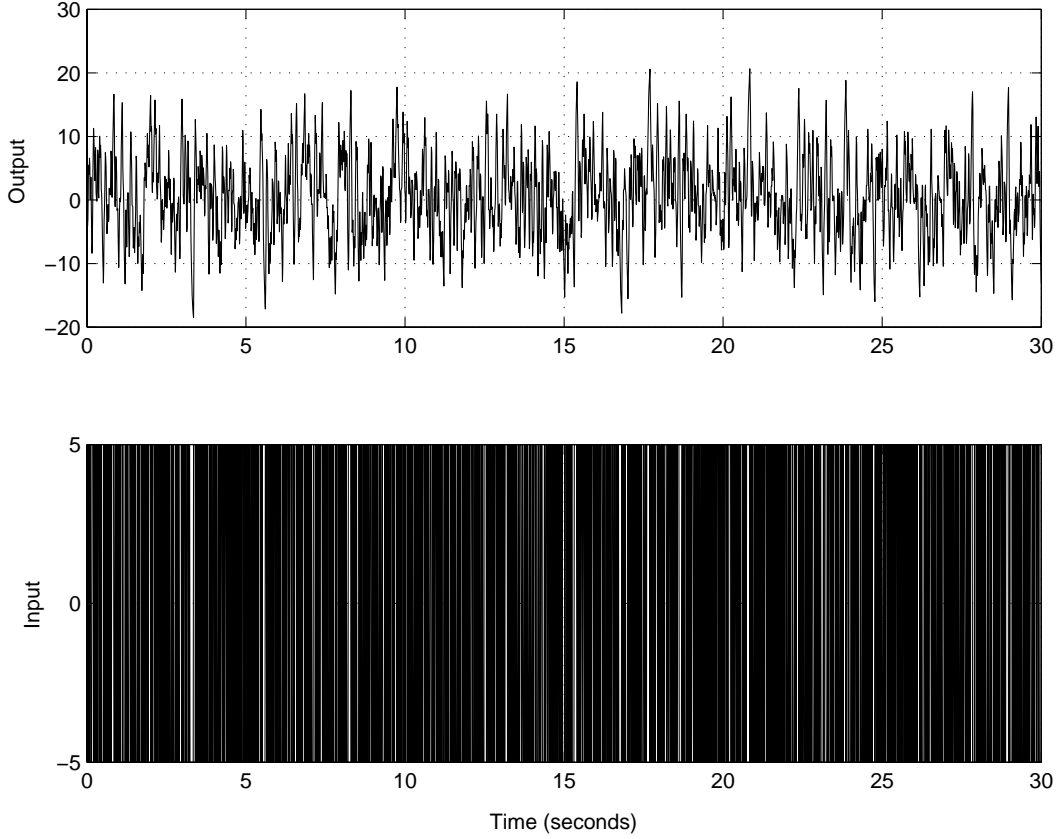


Figure 6.3: Input/output data of the identification experiment.

Table 6.1 shows the parameter vectors obtained by using the central algorithm in ℓ_∞ norm. The noise bound ε has been chosen as the smallest integer between 1 and 10 for which the feasible set turns out to be not empty. Note that higher values of the noise bound ε for some models mean that the model is not appropriate to estimate the system. From this table it is possible to conclude that the model which appear to be the most appropriate is for $n_a = 2$ and $n_b = 2$.

Table 6.2 reports the bounds relative to the model parameters provided by the central algorithm. For example, the box containing the parameters for $n_a = 1$

and $n_b = 1$ is

$$A = [1 \quad 0.39266 \pm 0.13082] \quad \text{and} \quad B = [0 \quad 0.39266 \pm 0.29905].$$

Notice that for $n_a = 2$, $n_b = 2$ the resulting box is very tight.

Finally, in Table 6.3, the parameter vectors obtained applying the projection algorithm are reported. Again in this case, the best model seems to be for $n_a = 2$ and $n_b = 2$.

Time plot regarding the behaviour of estimated models vs validation data are reported in Fig 6.4 and in Fig. 6.5 for the central algorithm and the projection algorithm respectively.

6.2.2 Statistical identification

In this subsection the results obtained by applying statistical identification algorithms are reported. Estimates are computed by the least squares algorithm, which under suitable statistical hypotheses is optimal (see [41]).

The estimate is then given by

$$x_{LS} = \arg \min_{x \in X} \|y - Fx\|_2. \quad (6.12)$$

Notice that this coincides with the ℓ_2 projection algorithm in the set-membership identification setting.

In Table 6.4 the optimal parameters vectors and the FIT index are reported. Note that also in this case the best model is $n_a = 2$ and $n_b = 2$ as previously achieved by set-membership estimators. Figure 6.6 shows the time plots obtained for various model orders.

n_a	n_b	ε	A	B	Fit
1	1	5	[1 -0.80501]	[0 0.39266]	43.867 %
1	2	2	[1 -0.76413]	[0 0.39233 0.424]	82.263 %
2	1	3	[1 -1.3505 0.48092]	[0 0.39038]	76.567 %
2	2	1	[1 -1.0447 0.24862]	[0 0.40592 0.31999]	87.8 %
1	3	2	[1 -0.73955]	[0 0.40019 0.4516 0.13488]	86.03 %
3	1	3	[1 -1.4443 0.86735 -0.27975]	[0 0.42202]	71.429 %

Table 6.1: Identification results concerning the central algorithm.

n_a	n_b	Error bounds on a_i	Error bounds on b_i
1	1	[0.13082]	[0.29905]
1	2	[0.069911]	[0.10384 0.11233]
2	1	[0.46552 0.38543]	[0.15696]
2	2	[0.09634 0.092721]	[0.04023 0.053808]
1	3	[0.11701]	[0.20282 0.22117 0.22784]
3	1	[0.63177 0.97816 0.48079]	[0.23269]

Table 6.2: Bounds on the parameters computed by the central algorithm.

n_a	n_b	A	B	Fit
1	1	[1 -0.76365]	[0 0.38011]	38.841 %
1	2	[1 -0.7594]	[0 0.38652 0.40863]	80.866 %
2	1	[1 -1.234 0.42702]	[0 0.38795]	62.185 %
2	2	[1 -1.0843 0.28293]	[0 0.40737 0.3071]	87.014 %
1	3	[1 -0.7432]	[0 0.38947 0.44094 0.12055]	86.157 %
3	1	[1 -1.7261 1.0115 -0.20096]	[0 0.43074]	51.725 %

Table 6.3: Identification results concerning the projection algorithm.

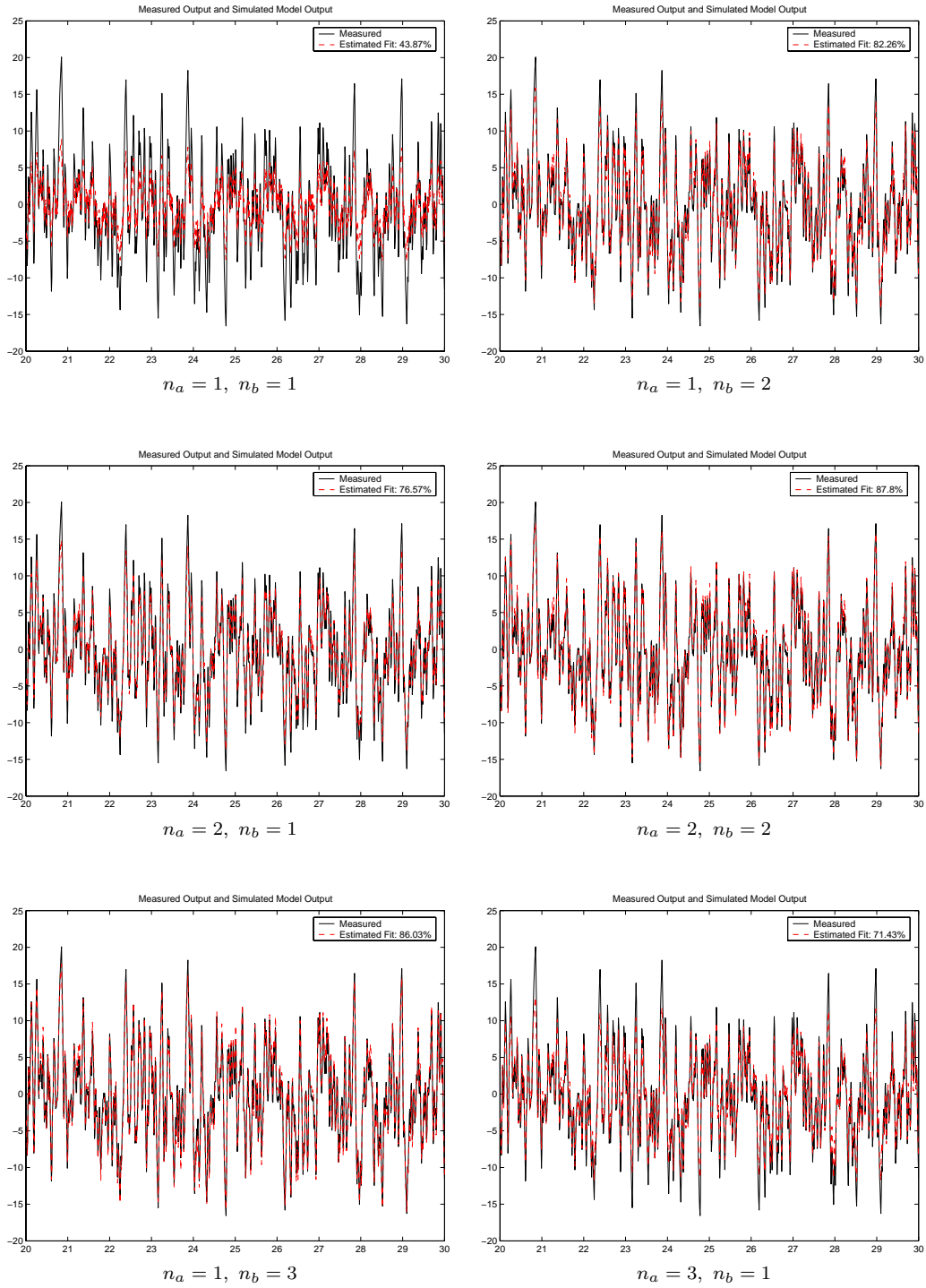


Figure 6.4: Time plot of real and estimated models validation data for different model orders (central estimate).

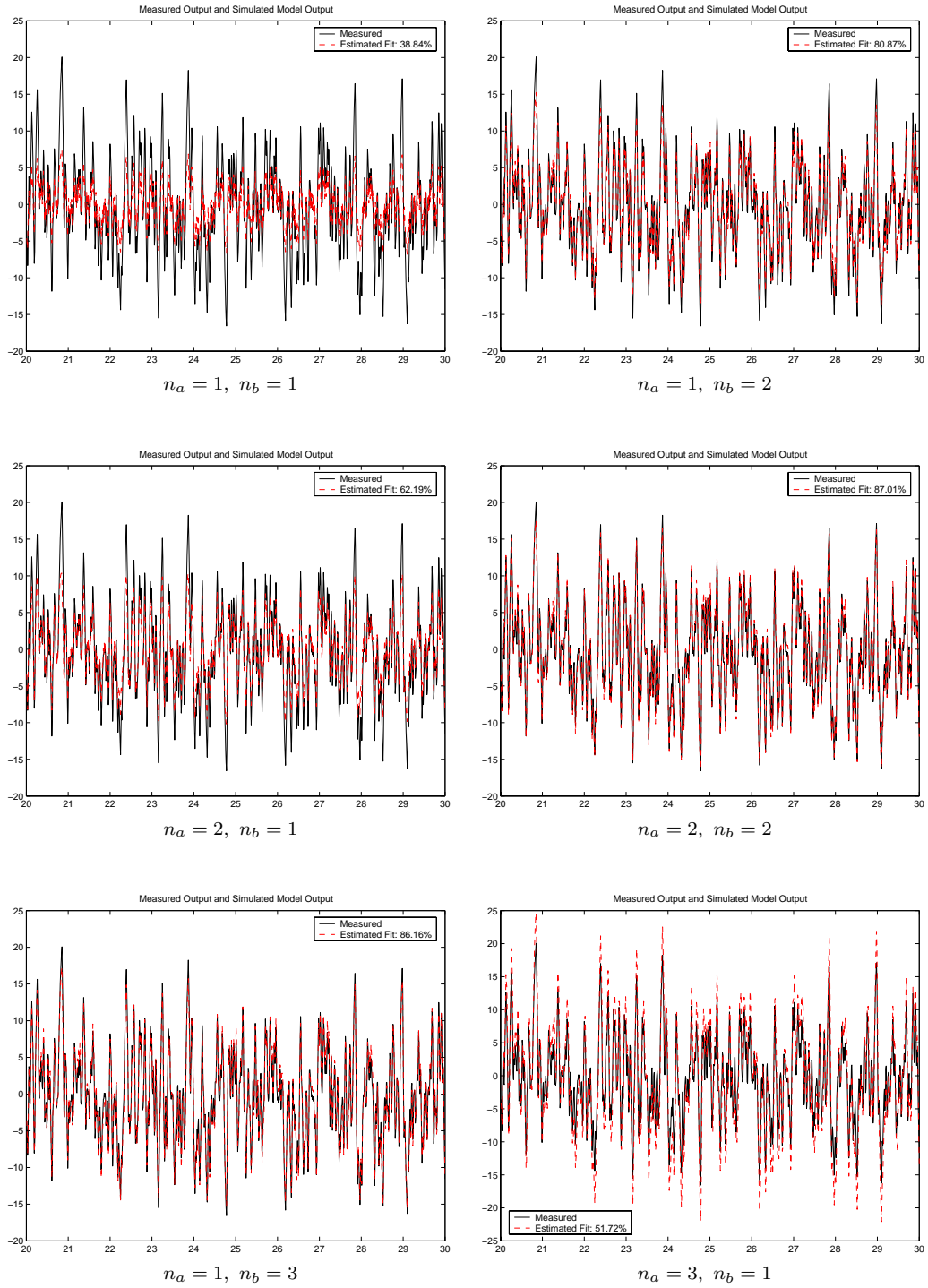


Figure 6.5: Time plot of real and estimated models for validation data for different model orders (projection estimate).

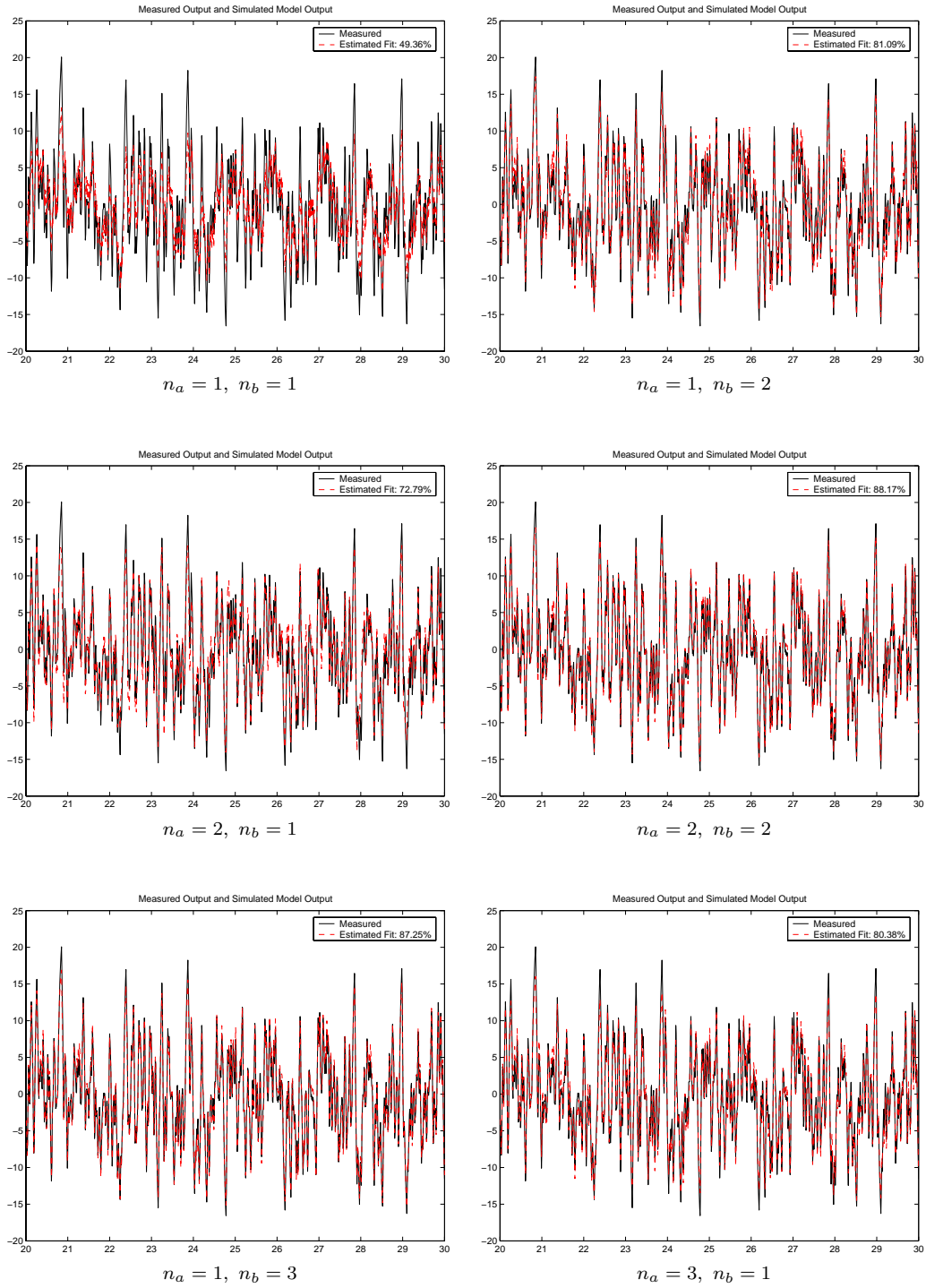


Figure 6.6: Time plot of real and estimated models for validation data for different model orders (least squares estimate).

n_a	n_b	A	B	Fit
1	1	[1 -0.89271]	[0 0.42607]	49.357 %
1	2	[1 -0.78973]	[0 0.4098 0.4153]	81.087 %
2	1	[1 -1.408 0.58263]	[0 0.41881]	72.787 %
2	2	[1 -1.0304 0.24309]	[0 0.41081 0.31192]	88.173 %
1	3	[1 -0.72946]	[0 0.41165 0.43558 0.12046]	87.251 %
3	1	[1 -1.6088 1.0576 -0.33604]	[0 0.41896]	80.385 %

Table 6.4: Identification results concerning the least squares algorithm.

Part II

ACT: a Remote Laboratory of Automatic Control and Identification

The *Automatic Control Telelab*

In this chapter, a remote laboratory of automatic control developed at the University of Siena is described. By means of remote laboratories it is possible to perform experiments on remote processes through the Internet (or other networks). These kinds of labs provide several benefits compared to common labs, and they will be described in the following.

The realized remote lab allows to perform both identification and control of a number of physical processes. Beyond other features described in the sections of this chapter, one distinguishing feature is that it allows to adopt much of the set-membership identification machinery introduced in the first part of this thesis in the process identification stage.

The chapter is organized as follows. Section 7.1 describes the state of the art of virtual and remote laboratories. In Section 7.2 an overview of the Automatic Control Telelab (ACT) is reported, while in Section 7.3 a complete working session is described. Sections 7.4 and 7.5 deal with the student competition and the remote identification respectively. Finally, an overview on the software architecture is reported in Section 7.6.

7.1 Remote laboratories; state of the art

Recent years have witnessed dramatic changes in graduate education because of Internet and Web technologies. Web-based teaching, distance learning, electronic books, and interactive learning environments will play increasingly significant roles in teaching and learning processes in the near future.

Tele-laboratories are expressions of a more general distance education which is attracting wide attention in the academic and government communities. Reference [45] analyzes the state of the art and outlines the future perspectives of on-line distance education in the United States. In that paper, the point is made that the technology to meet the need of distance education is already available, and it will further improve in the near future. Moreover, software engineers consider on-line education as an important emerging market and, therefore, a business opportunity. An overview of Web-based educational systems is given in [46] where an accurate description of the actual problems concerning the design of Web-based educational systems and their possible solutions is also provided.

In [47] Poindexter and Heck present a survey on the usage of Web-based educational systems in the control area. In this paper, authors overview the level of integration between Internet and control courses. Special attention is devoted to the automatic control laboratories accessible through the Web. Tele-laboratories are divided into two classes: *virtual labs* and *remote labs*. The first ones are systems which can run simulations remotely with possible animations of the controlled system. One of the first instances of virtual labs has been developed with CGI scripts and Java applets at the University of Edinburgh, Scotland [48]. The student typically connects from a remote client to the virtual lab server, chooses the experiment, changes some parameters and runs the simulation. Then he or she looks at the simulation results through some graphical interface or downloaded data stored in a given format.

The Matlab environment has been used as the core of the virtual lab in [49] and in [50]. A plug-in is required to launch Matlab on the remote machine from within a Web browser. The advantage of Matlab over Java is that the first is a standard computational tool for control system applications. Therefore, it is considered a standard tool for students of control courses so that they do not need to learn new languages to run a virtual lab session. In fact, several toolboxes are available in the Matlab environment for control applications, e.g. optimization, μ -control, and system identification. Standardized Computer Aided Control System Design (CACSD) tools, mainly based on the Matlab/Simulink environment, have been thoroughly discussed in [51]. In [50] two more plug-ins are needed to run a virtual lab session: one is necessary to display Matlab figures in the browser windows, and one to use the Virtual Reality Modelling Language (VRML) to render the simulation results. In [49], students can run two simulations: a magnetically levitated vehicle and an automated highway. In [50], ball and beam, tanks' level control, and gyro pendulum simulations are available.

Remote labs are laboratories where students can interact with actual experiments via Internet. Usually, remote operators through a Web interface can change several control parameters, run the experiment, see the results and download data. This situation is for instance, the case of [52], where a remote lab for testing analog circuits is described; in [53], a remote chemical control process is implemented, and in [54] several laboratory experiments are made available.

The complexity of designing the hardware and software architecture of the remote laboratory dramatically increases when one of the features required is designing the controller within a remote Web session. One such instance is the case of the remote lab developed at the College of Engineering of Oregon State University [55, 56]. Here, students can remotely control a robot arm by changing some parameters but, more interestingly, transmitting the control

program which changes the dynamics of the closed-loop system. The student can edit his/her own controller and paste it in an applet window. Then this code is uploaded to the server machine of the remote lab, compiled and executed on the robot arm, becoming an actual remote lab experiment. The user interface consists of a graphical network application referred to as SBBT (Second Best to Being There). The design has three main features: collaboration with peers, active presence, and complete control of the remote experiment. Another approach has been investigated in [57] where the user can run an experiment using a controller which resides not only on the server but also on the client, compiling and executing it on the user machine. In this case, some issues about network reliability and delays have been addressed. The audio and video feedback is very important to increase the effect of telepresence, as shown in [58] where authors implemented an interesting pan, tilt, and zoom control of cameras grabbing the experiments of the remote lab.

In general a remote laboratory can use a well-known software environment, such as LabVIEW [59] or Matlab/Simulink [60, 61], but it can also use a special purpose one, as in [62], where a flow process is analyzed.

In [63] a comparison between virtual labs and remote labs is presented. The authors examine a common experiment (ball and beam) from these two points of view and conclude that virtual labs are good to assimilate theory, but they cannot replace real processes since a model is only an approximation which cannot reproduce all the aspects of the process, such as unexpected non-linearities. To avoid these issues, remote laboratories which allow a student to interact directly with real processes should be used; this opportunity takes a fundamental role especially for engineering students.

From a pedagogical point of view, remote labs, allowing for designing the whole control law, are more stimulating. Typically, the price to pay to obtain the controller design feature is that students must learn and use new control languages which are designed for the remote lab and cannot take advantage of

control functions developed in other contexts.

In next sections, the realization of an automatic control remote lab, the Automatic Control Telelab (ACT) developed at the University of Siena, is presented.

7.2 Features of the Automatic Control Telelab

The extension of the teaching capabilities through the Internet is at the base of the Automatic Control Telelab project [64, 65, 66], whose home page is reported in Fig. 7.1.



Figure 7.1: Automatic Control Telelab's home page.

The aim of the project was to allow students to put in practice their theoretical knowledge of control theory in an easy way and without restrictions due to laboratory opening time and experiments availability. The ACT is accessible 24 hours a day from any computer connected to the Internet. No special software or plug-in is required. The ACT is accessible by means of any common browser like Netscape Navigator or Microsoft Internet Explorer. If the user wants to

design his/her own controller, the Matlab/Simulink software is required. The accessibility through the Internet makes the ACT a safe and effective tool for students with disabilities.

Like most of the University labs, the ACT increases the teaching performance of control theory classes. Students can operate as if they were in a conventional laboratory through a full set of operations. They can take decisions on what experiment to run and can learn from mistakes they make. Since students perform their lab assignments through the ACT, the efficiency and the student convenience is dramatically increased along with utilization of equipments.

The ACT remote lab is continuously upgraded with new software versions and experiments. At the present stage, four processes are available for on-line experiments: a DC motor, a tank for level control, a magnetic levitation system and a two degrees-of-freedom helicopter (Fig. 7.2). The DC motor is used to control the axis angular position or the rotation speed. The level control process has been included because, in spite of its simplicity, it shows nonlinear dynamics, whereas the magnetic levitation process, being nonlinear and unstable, shows very interesting properties to be analyzed in control theory education. Finally, the two degrees-of-freedom helicopter, being a nonlinear unstable MIMO system, can be used in graduate control system courses.

In what follows, some of the most interesting features of the ACT are presented.

Easy-to-use interface. Simplicity of use is essential to realize an interface that can be used by everyone [47, 67], so the user can focus his/her efforts on interacting with the experiment without spending time to understand how to proceed through the web site. The ACT is based on intuitive and simple HTML pages and Java applets, that are fully supported by the latest versions of browsers. Help pages are also provided for detailed information. It is not required to install any browser plug-in and software locally. The use of Matlab and Simulink is only required on the remote computer if the user wants to











<p><i>Position Control</i></p> <p>Control of the angle of a DC motor.</p>  <p>System Description Control Experiment</p>  <p>Process Ready</p>	<p><i>Speed Control</i></p> <p>Control of the speed of a DC motor.</p>  <p>System Description Control Experiment Identification Experiment</p>  <p>Process Ready</p>
<p><i>Level Control</i></p> <p>Control of the water level in a tank.</p>  <p>System Description Control Experiment</p>  <p>Process Ready</p>	<p><i>Magnetic Levitation</i></p> <p>Control of a magnetic levitation system.</p>  <p>System Description Control Experiment Student Competition</p>  <p>Process Ready</p>
<p><i>Helicopter Simulator</i></p> <p>Control of a 2 DOF Helicopter.</p>  <p>System Description Control Experiment</p>  <p>Process Ready</p>	

Figure 7.2: The Automatic Control Telelab's on-line experiments.

design his/her own controller.

Simulink based interface for controller design. This is an important distinguished feature of the ACT. It consists in designing a user-defined controller, which will drive the real process, by using Simulink and all its powerful toolboxes. Matlab and Simulink are standard tools in the control community, and they are adopted in many basic and advanced courses. So, being the controller simply a Simulink model, it will not be necessary for the user to learn a new language to implement a controller. A basic knowledge of Matlab/Simulink environment is only required. It is the authors' belief that a different choice for the ACT would have discouraged users from making the effort necessary to learn a new syntax to design the controller. Through the use of the Simulink graphical interface and its toolboxes, it is possible to choose among a large set of functions to build the controller. Thus it becomes very easy to design any type of controllers to be tested, no matter if it is designed in continuous or discrete or if it is linear or nonlinear. At the end of the experiment, the user can download a file in the Matlab workspace format (.mat), where all data of the experiment have been stored for off-line analysis.

Predefined and user-defined controller types. Each experiment of the remote lab can be controlled in two ways: using a predefined or a user-defined controller. In the first case, the student will choose a control law in a given list, and then assign the value of typical parameters. For example, a student can select a PID controller to run the experiment and choose the values of proportional, integral and derivative coefficients.

Rather than using a predefined controller, the user can design his/her own controller to drive the experiment, by means of the Simulink graphical interface, and send it to the ACT server. A Simulink user-defined template is available to help the remote user in this phase.

Predefined and user-defined reference types. The remote user can also choose

what references will steer the process. He can select some references among a given list or create new ones building a Simulink subsystem.

Controller parameter change. While an experiment is running, the ACT provides a mechanism that allows the user to change some typical controller parameters on-line (e.g. the coefficients of the PID controller). Of course, working over the Internet, parameters will be updated after the time needed by packets to reach the ACT server. These time lags can depend on the distance, the type of Internet connection and the network congestion. However, these delays cannot be dangerous for the control of the process since the control law resides in a pc directly connected to the process, as explained in Section 7.6. The only consequence of these time lags consists in a delay between the user parameter change request and the execution of the command.

Tunable parameters can also be included in the user-defined controller by naming the parameter variable according to a special and simple syntax, as described in Section 7.3.1.

Reference change. It is also possible to change the system reference while an experiment is running. In other words, the user does not have to start a new experiment to verify the response of the system to different input signals.

Lab presence. For effective distance learning, it is important for the user to have the sense of presence in the laboratory. To obtain this, a live video and on-line data plots are provided, thus it is possible for students to view the real process while the experiment is in progress. The lab presence is the feature which distinguishes remote labs from virtual labs which only provide software simulations of physical processes.

Resource management and system safety. Like every remote lab, the experiment hardware is controllable by one user at a time. To prevent process monopolization, only a fixed amount of time is assigned to each experiment session. After that time, the user is automatically disconnected and the pro-

cess will be ready for the next experiment. From the web page showing the list of available experiments (Fig. 7.2) it is possible to know which processes are ready as well as the maximum delay time regarding the busy experiments.

As regard the system safety, hardware and software saturations of actuators are used to prevent from users dangerous operations. A check on the maximum input reference extension is performed also. Moreover, a software instability detection system has been provided to reveal when a system becomes unstable, stopping it.

Simplicity to add new processes. The software and hardware architecture of the ACT have been designed with the goal of simplifying the procedure of connecting new processes to the remote laboratory. Note that, as far as the software is concerned, only a Simulink model and a text file must be created to add a new process to the ACT.

7.3 A Session Description

In this section, a typical working session is described. From the home page of the ACT it is possible to access to general information pages, as for instance the user guide of the laboratory and the list of the available experiments. After choosing the experiment to run, the *Control Type Interface* shows up as in Fig. 7.3. Through this interface, the user fill in a form containing personal data (used to provide statistics about ACT users), and then he chooses the controller to be integrated in the control loop of the remote laboratory process. Although some predefined control laws are available, the most stimulating experience for the user consists in synthesizing his/her own controller and reference input signal.

Level Control

Control of the water level in a tank.

Personal Data

Name	Coutry	Email
Marco Casini	Italy	casini@ing.unisi.it

Predefined Controllers

Download Model	P.I.D. Controller	Run Experiment
Download Model	P.I.D. Controller with Anti Windup	Run Experiment
Download Model	Feedback Linearization Controller	Run Experiment

User-defined Controller

?	Controller Model	<input type="text"/> Sfoglia...
?	Controller Data	<input type="text"/> Sfoglia...
?	Sample Time (msec)	<input type="text"/> Range=[100,1000]
?	Download Template	Send Controller
?	Download Simulator	

Figure 7.3: The *Control Type Interface*.

7.3.1 User-defined controller

To simplify the controller design, a template model can be downloaded by the user from the control type interface. This template is a Simulink model which contains two subsystems, one for the controller (“ACT_Controller”) and one for the reference input (“ACT_Reference”), see Fig. 7.4.

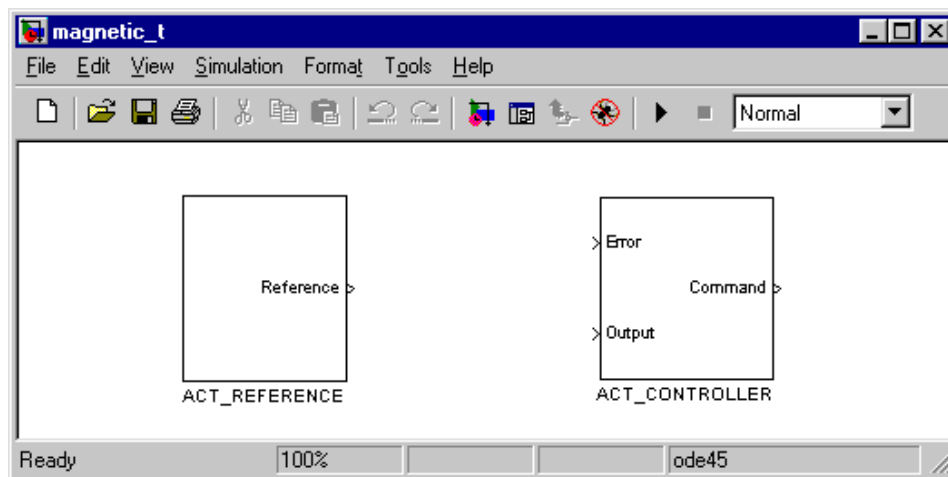


Figure 7.4: The Simulink template model for reference and controller design.

Only a very basic knowledge of the Simulink environment is required to design the controller. The control error, output and command signals are available in the “ACT_Controller” subsystem as shown in Fig. 7.5. The task which is left to the user is that of joining them by means of suitable blocks which define the controller structure. Such blocks can be dropped by any Simulink toolbox available. Moreover, it is also possible to set some “constant” and “gain” as variable parameters which can be modified on-line while the experiment is in progress. This interesting feature is obtained by simply using the prefix “ACT_TP_” (ACT Tuning Parameter) to name these variables as described in the bottom window of the Simulink template in Fig. 7.5.

The “ACT_Reference” subsystem of the template file (Fig. 7.6) is used to build new references which can enter the system during the experiment. A

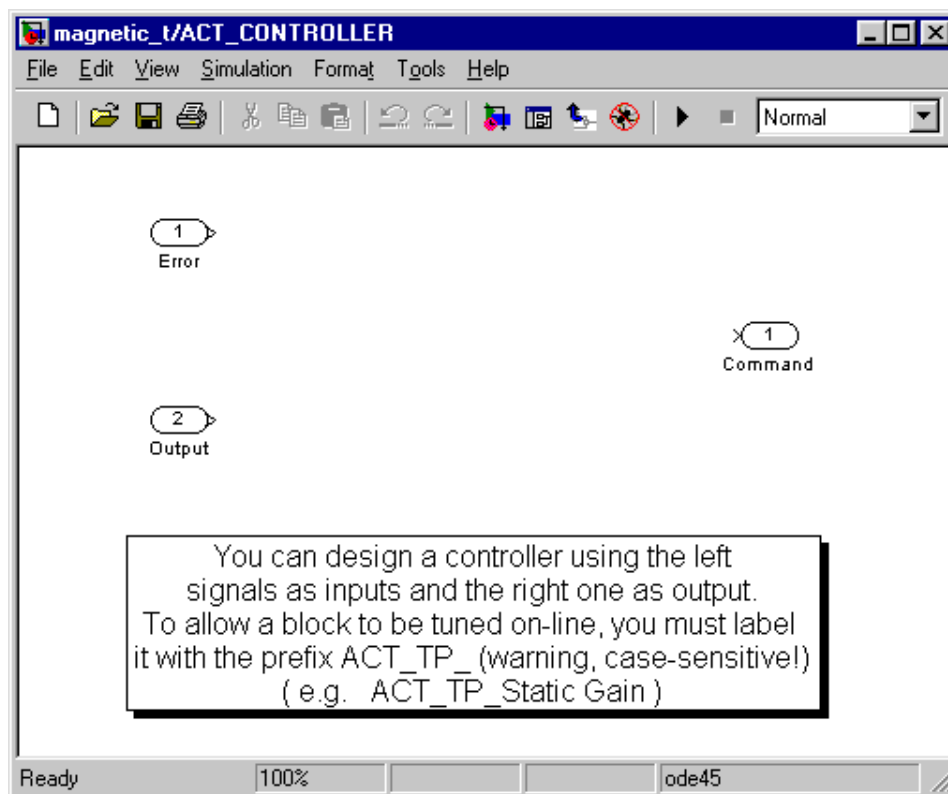


Figure 7.5: The ACT_Controller Simulink subsystem.

set of references is available by default, such as constant and ramp signals or sinusoidal and square waves. The user can remove some of these blocks or add new ones, see Fig. 7.7. To help the user in this task, other reference blocks are provided inside the “Other References” subsystem (Fig. 7.6). However, for advanced users, it is also possible to design special reference input signals in the Simulink environment.

7.3.2 The tank level and magnetic levitation examples

In this section the tank level and magnetic levitation processes shown in Fig. 7.2 are described.

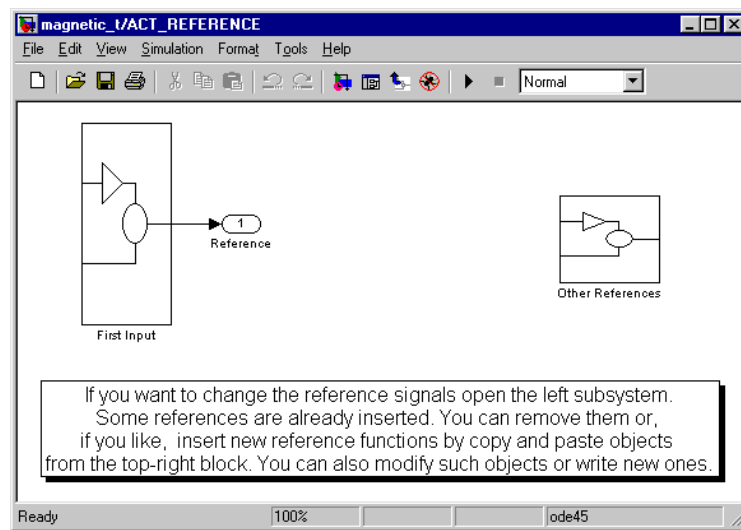


Figure 7.6: The ACT_reference subsystem.

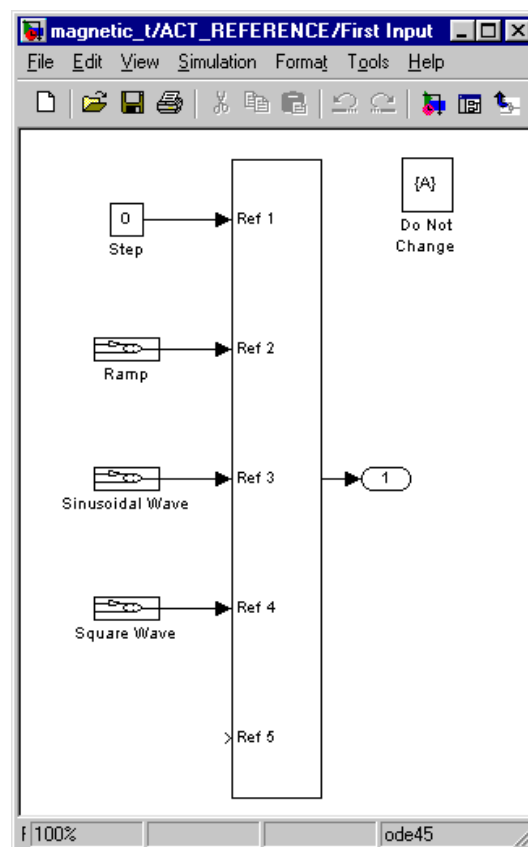


Figure 7.7: A detail of the *first input* block where the user can add any new reference input signal.

The mathematical model of the tank is

$$\dot{h}(t) = -0.008 \sqrt{h(t)} + 100 q(t) \quad (7.1)$$

with:

$$q(t) = \begin{cases} 0 & \text{if } V(t) \leq 3.7 \\ 1.36 \times 10^{-5} (V(t) - 3.7) & \text{if } V(t) > 3.7 \end{cases} \quad (7.2)$$

where h is the water level inside the tank, measured by a pressure transducer on the bottom of the tank, q is the input flow and V is the voltage applied to the pump (command). Due to a threshold on the actuator (pump) the input flow is zero when the voltage applied is less than 3.7 Volts. Moreover an input saturation of 8 Volts is present. Dynamics of the tank process is nonlinear. A possible type of controller is based on the so called *feedback linearization*, whose goal is to cancel the nonlinear part of (7.1) through a suitable action on the command. Moreover, applying to the pump a constant voltage of 3.7 Volts, the problems due to the threshold can be avoided. The Simulink model implementing the feedback linearization controller is shown in Fig. 7.8. It has been obtained from the “ACT controller” template in Fig. 7.5 by simply linking the error, output and command nodes through suitable Simulink block functions. Two parameters have been set to be tuned on-line. The *Proportional Coefficient* is the proportional gain on the system error, while the *Linearization Coefficient* is used to cancel (or at least reduce) the effect of the nonlinearity. Since the model described in (7.1) is an approximation of the true plant, on-line tuning of these parameters is mandatory to get better performances.

The levitation process consists of a magnetic suspension, a ball, whose height must be controlled and an electro magnetic coil. The height of the ball is sensed by an optical sensor. The minimum and the maximum distance of the ball from the coil is respectively 3 cm and 7 cm. The power amplifier supplies the coil with current that is proportional to the command voltage V_u of the actuator. A protection circuit sets the current to zero when it goes over 3 Ampere. Let z be the height of the ball of mass m . System dynamics is

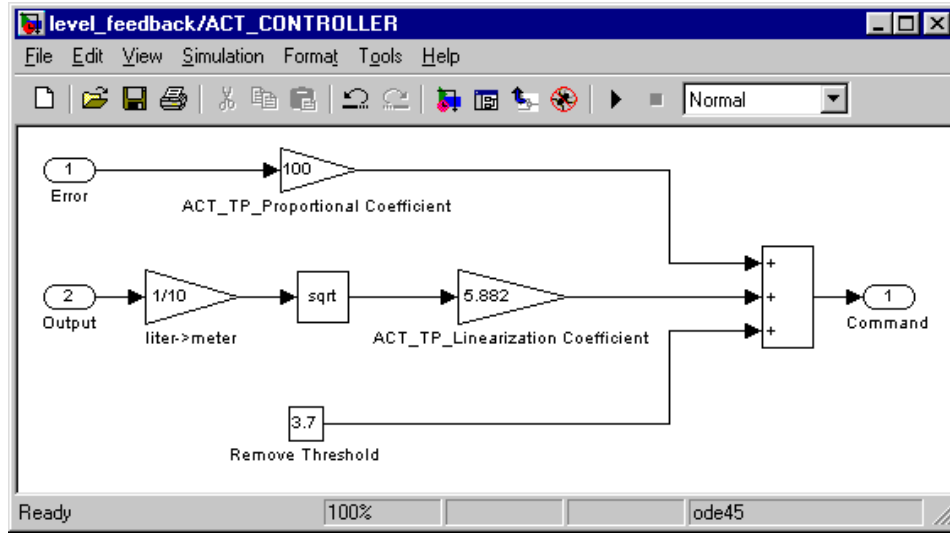


Figure 7.8: The Simulink model representing a controller for the tank process with tuning parameters based on *feedback linearization*.

simply derived as

$$\ddot{z}(t) = g - \frac{F_m}{m}$$

being $F_m = k_{ma} \frac{V_u^2}{z^2}$ the magnetic force and k_{ma} a system constant. A predefined controller is the P.I.D. controller with a pre-filter on the reference whose Simulink model is reported in Fig. 7.9. The coefficients of the proportional, integral and derivative actions can be tuned while the experiment is running.

7.3.3 Running the experiments

Once the user-defined controller has been built, one must upload the controller model to the ACT server through the *send controller* button (this operation is not needed for predefined controllers). If the Simulink model does not contain syntax errors, the *Experiment Interface* shows up, as in Fig. 7.10, whereby it is possible to run the remote experiment through the *start* button. When the experiment is in progress, the user can look at the signals of interest in a window displaying the control input, the reference input and the output along with their numerical values as shown in Fig. 7.10 about the tank level

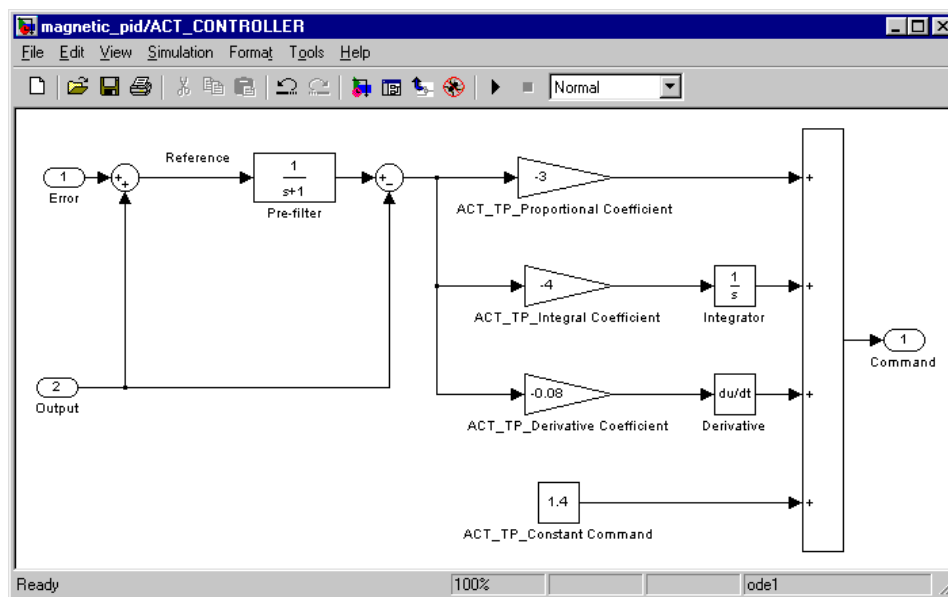


Figure 7.9: The P.I.D. controller for the magnetic levitation process with tuning coefficients.

experiment and in Fig. 7.11 concerning the magnetic levitation experiment.

Note that in the tank level process the input reference is a sinusoidal input while in the magnetic levitation process the step input is considered. Both reference signals have on-line tuning parameters.

Moreover, a live video window is provided to view what is really occurring in the remote lab. Unlike virtual laboratories, based only on software simulations, the presence of a video window is an important feature because the user can look at the real process, having a most sense of presence in the laboratory.

During the experiment it is possible to change references on-line, as well as the controller parameters.

When the user stops the experiment, it is possible to download a file in Matlab format (*.mat*) where all the signal dynamics have been stored. This file can be used to perform off-line analysis (such as the evaluation of the maximum overshoot and the settling time) as shown in the time plot in Fig. 7.12 regarding

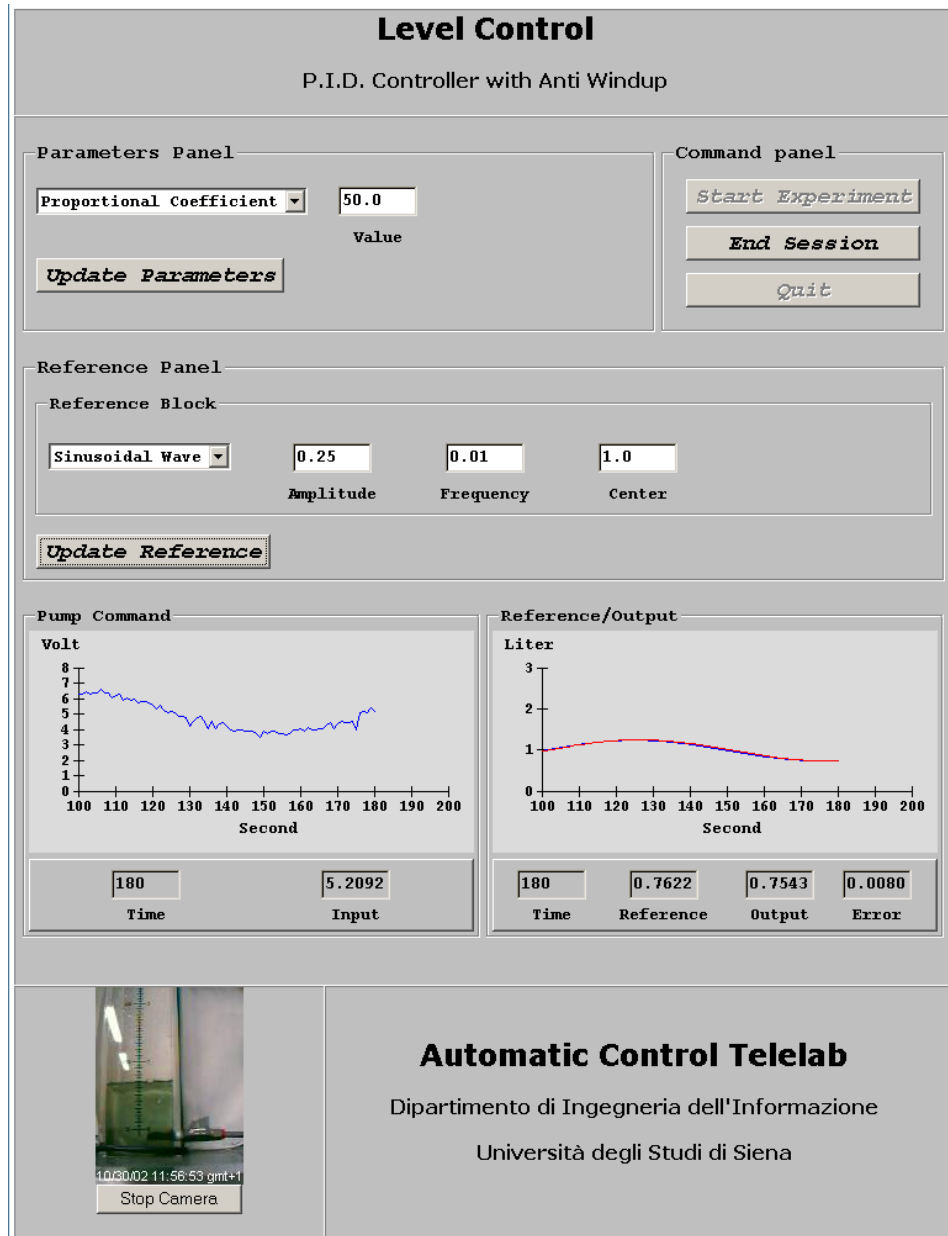


Figure 7.10: The *Experiment Interface* which allows the real execution of the remote experiment on the tank process.

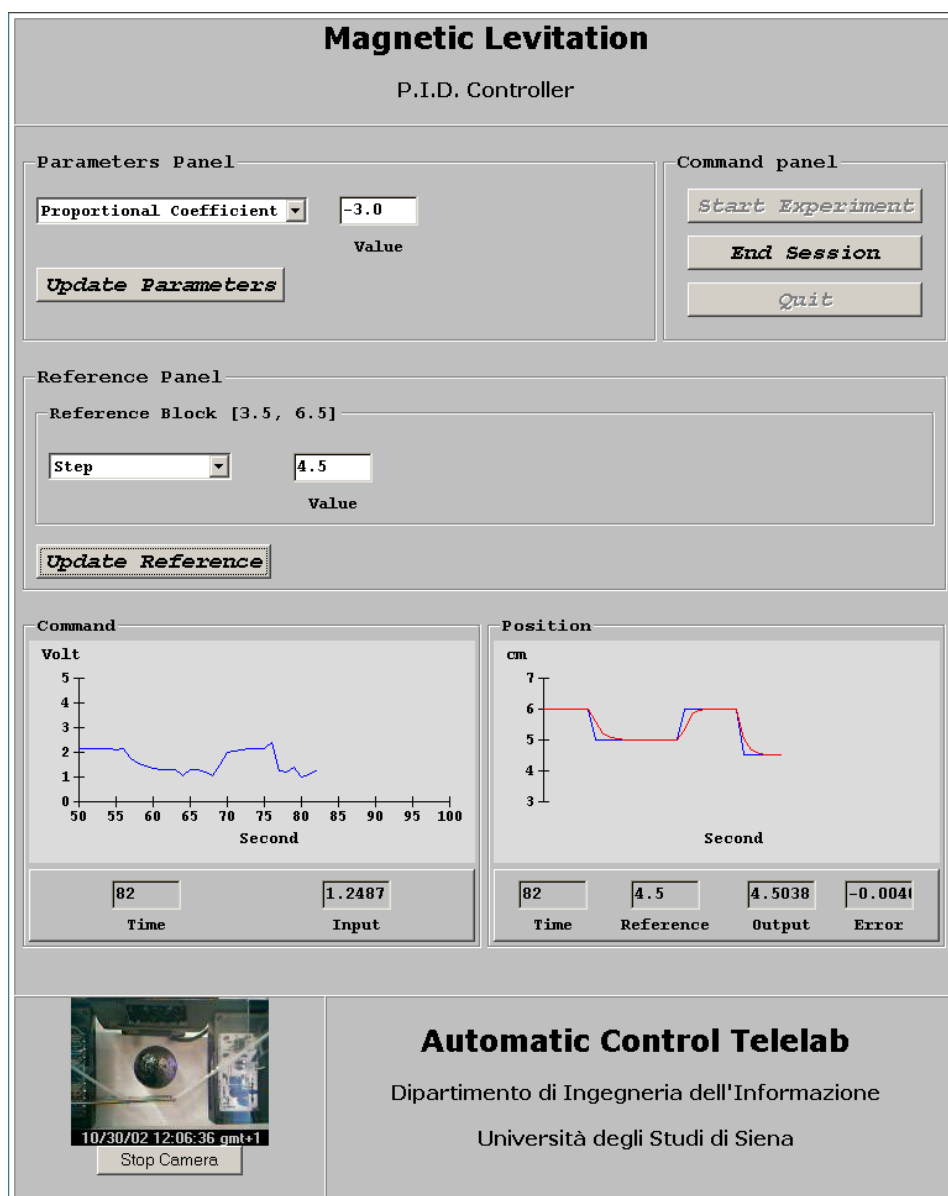


Figure 7.11: The *Experiment Interface* which allows the real execution of the remote experiment on the magnetic process.

the magnetic levitation process.

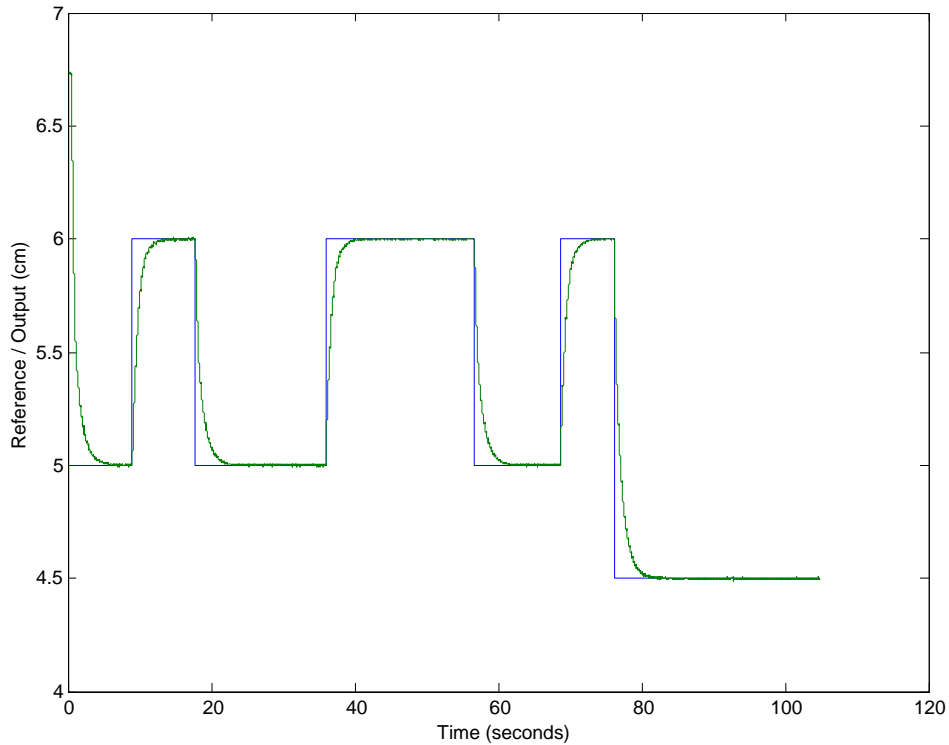


Figure 7.12: The time plot regarding an experiment on the magnetic levitation system.

To avoid that a user could run an experiment for a long time (preventing the access by other users) a time-out is implemented. When this time-out expires, the user is automatically disconnected.

7.4 Student competition overview

A typical remote laboratory allows users to run remote experiments using predefined or user-defined controllers. Students can run an experiment and see the dynamic response, but in general no information on controller performances is provided and it is not possible to know how controllers designed by other people behave on the same process. This is one of the reasons that motivated the building of a student competition mechanism for the ACT. Through this

tool a student knows about performance requirements his/her own controller must satisfy. Moreover a final ranking of the best controllers as well the time plots of the relative experiments are provided.

In the following some features of the ACT competition structure are described.

Remote exercises: in addition to standard control synthesis exercises, this tool allows a student to design a controller, which must satisfy some performance requirements, and to test it on remote real processes. At the end of the experiment, the performance indexes are automatically computed and shown to the user; if such indexes fulfil the requirements, the exercise is completed. An overall index is then computed (usually as a weighted sum of the previous indexes) and the controller is included in the ranking list.

Controller comparison: since this tool allows everybody to view the ranking concerning a competition, it is possible to know what kind of controller achieved better results. In addition to the type, for every controller it is possible to see some data such as a description and the time plot of the experiment. Moreover, for controllers which do not satisfy the requirements, it is also possible to download the Simulink models of that controllers. During the end-competition lesson, students who have designed the best controllers are invited to discuss their projects, while the lecturer shows why some control architectures work better than others.

Many competitions on the same process: it is possible to provide more than one competition benchmark on the same process, thus increasing the number of remote exercises available for students. Due to the software design of the competition structure of the ACT, new benchmarks can be added very efficiently.

It is the authors' opinion that competition can be considered as a new useful

tool for distance learning and, at the same time, a tool which increases the potentiality of remote laboratories.

7.4.1 A competition session description

In this section, a competition session is described. In particular an example of competition regarding the process of magnetic levitation (see Fig. 7.2) is reported [68].

First of all, a student or a group of students who want to compete need to register by filling the form shown in Fig. 7.13 and obtaining a username and a password.

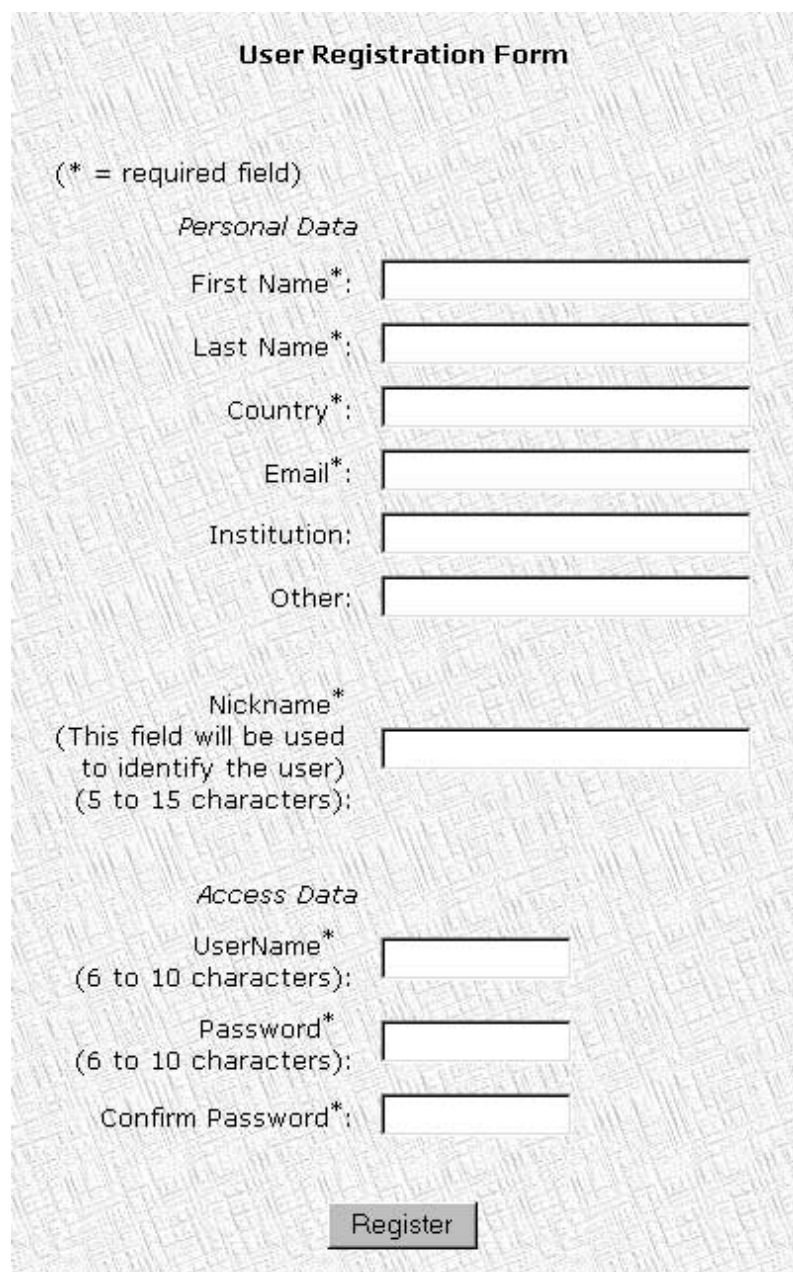
The user can analyze the mathematical model of the process (provided as a *pdf* file) as well as the required performance specifications. In this example it is required that, for a step reference, the settling time (5%) must be less than 1 second and the overshoot must be less than 40%. A more detailed description shows also the working point of the nonlinear benchmark.

The mathematical model of this process, sketched in Fig. 7.14, is summarized as follows

$$\begin{cases} M \ddot{z} &= M g - F_m \\ F_m &= k_m \frac{i^2}{z^2} \\ i &= k_a V_u \end{cases} \quad (7.3)$$

where z is the absolute distance of the center of the ball from the coil, M is the mass of the ball, F_m is the magnetic force, i is the current in the coil, and V_u is the input voltage of the coil ($0 \leq V_u \leq 5$); k_m and k_a are the magnetic constant and the input conductance respectively. The actual values of these coefficients are reported in Table 7.1.

Equation (7.3) can be rewritten with $x_1 = z$, $x_2 = \dot{z}$, $u = V_u$ (input command)



User Registration Form

(* = required field)

Personal Data

First Name*:

Last Name*:

Country*:

Email*:

Institution:

Other:

Nickname*
(This field will be used
to identify the user)
(5 to 15 characters):

Access Data

UserName*
(6 to 10 characters):

Password*
(6 to 10 characters):

Confirm Password*:

Figure 7.13: User registration form.

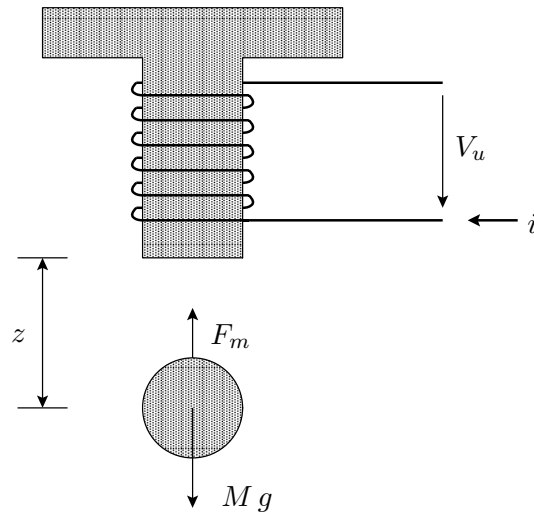


Figure 7.14: Sketch of the process.

M	Mass of the ball	$20 \cdot 10^{-3} \text{ Kg}$
k_m	Magnetic constant	$2.058 \cdot 10^{-4} \text{ N(m/A)}^2$
k_a	Input conductance	$0.5488 \text{ 1}/\Omega$
g	Gravity acceleration	9.80665 m/s^2
k_y	Unit conversion	100 cm/m

Table 7.1: Magnetic levitation system parameters.

and $y = k_y z$ (output in centimeters).

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= g - \frac{k_m k_a^2}{M} \frac{u^2}{x_1^2} = g - k_t \frac{u^2}{x_1^2} \\ y &= k_y x_1 \end{cases}$$

By substituting the actual values of parameters in the above equations, one obtains:

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= 9.80665 - 0.0031 \frac{u^2}{x_1^2} \\ y &= 100 x_1 \end{cases}$$

Since the competition is based on an experiment around the state ($x_{10} =$

$0.05m$, $x_{20} = 0$), students can choose to linearize dynamics

$$\begin{cases} \Delta \dot{x} = A \Delta x + B \Delta u \\ \Delta y = C \Delta x + D \Delta u \end{cases}$$

It follows immediately that $u_0 = \sqrt{\frac{g x_{10}^2}{k_t}} = 2.811$, thus linearized matrices are:

$$A = \begin{bmatrix} 0 & 1 \\ \frac{2 k_t u_0^2}{x_{10}^3} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 139.4389 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ -\frac{2 k_t u_0}{x_{10}^2} \end{bmatrix} = \begin{bmatrix} 0 \\ -6.9719 \end{bmatrix}$$

$$C = [k_t \ 0] = [100 \ 0] \quad , \quad D = [0]$$

Now a linear controller, such as a PID or a lead-lag compensator, can be synthesized. Of course, advanced students can design controllers with nonlinear techniques.

In order to design the controller, students must run the Simulink environment on their own local computers, then download a template file (*template.mdl*) and connect the signals describing the output, the error and the command to design the desired controller as previously described in Section 7.3.1.

A special interface (Fig. 7.15) allows a student to describe the structure of his/her own controller (i.e. P.I.D. Controller) and to set the sample time of the experiment; if the controller is continuous time, the sample time is intended as the integration step of the Simulink solver. Moreover, the user has to specify the file containing the controller and, if needed, the Matlab workspace file (*.mat*) containing essential data for that controller. Those files will be uploaded to the server, compiled and, if no error occurs, executed on the real remote process.

A second graphical interface (Fig. 7.16) allows a user to start the experiment and to observe its behaviour through plots and the live video window.

At the end of the experiment, the performance indexes are computed and are displayed to the user. It is now possible to download a Matlab workspace file containing the full dynamics of the experiment and to view the time plots (Fig. 7.17). The ranking of the user controller is given as in Fig. 7.18.

Since several controllers can achieve the requested performance, an overall index is evaluated to build the ranking. This index is obtained by weighting each performance index. If a controller does not satisfy the requirements, the overall index is not computed.

It is possible, for every user, to view a controller report (Fig. 7.19) where information on ranking and other data, such as the controller description, the nickname of the user and his/her nationality and institution, are displayed.

7.4.2 Teaching experiences

In spring 2002 undergraduate control system classes at the University of Siena used the student competition system.

First of all, the lecturer illustrated the physical model of the magnetic levitation system, emphasizing its unstable and nonlinear dynamics, and suggesting the students to linearize dynamics to design the controller.

Since ACT is accessible at any time, students had no problems to analyze the process and test their own controllers during the days before the second competition class, where the lecturer answered students about their questions and difficulties, and helped them to solve some typical problems. For example, he suggested them to use a pre-filter on the reference to obtain smoother command signals, and, in general, better performances.

Magnetic Levitation

Control of a magnetic levitation system.

Competition on Overshoot and Settling Time

?	Controller Type	P.I.D. Controller
?	Controller Description	P.I.D. Controller based on the linearized model around $y=5$.
?	Controller Model	C:\act\control_pid.mdl Stoglia...
?	Controller Data	Stoglia...
?	Sample Time (msec)	1

Run Experiment

Figure 7.15: The interface describing the controller features.

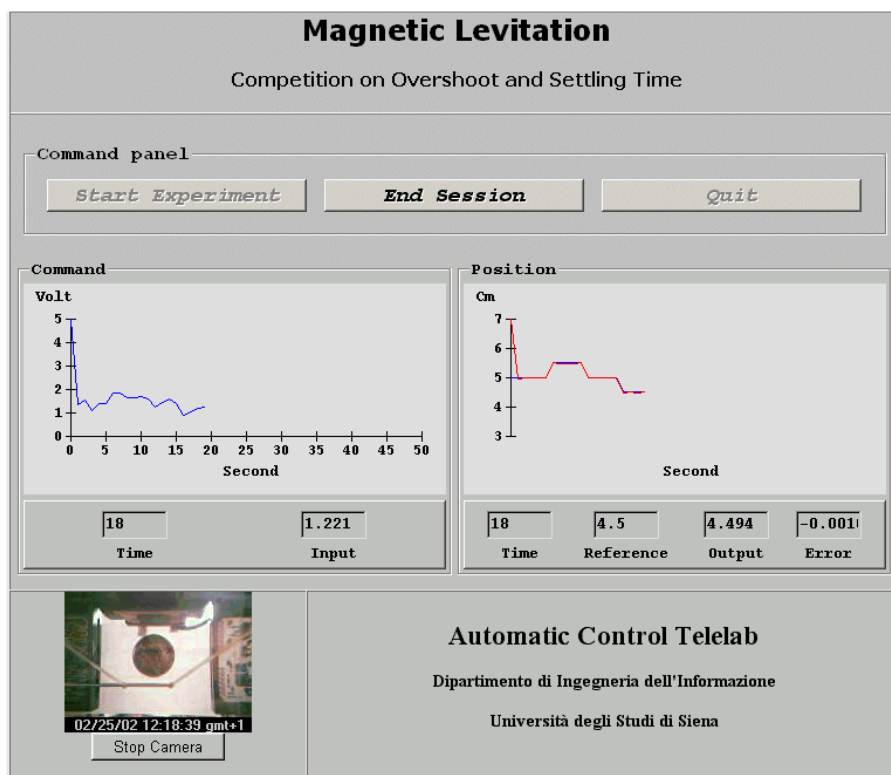


Figure 7.16: The interface showing the running experiment.

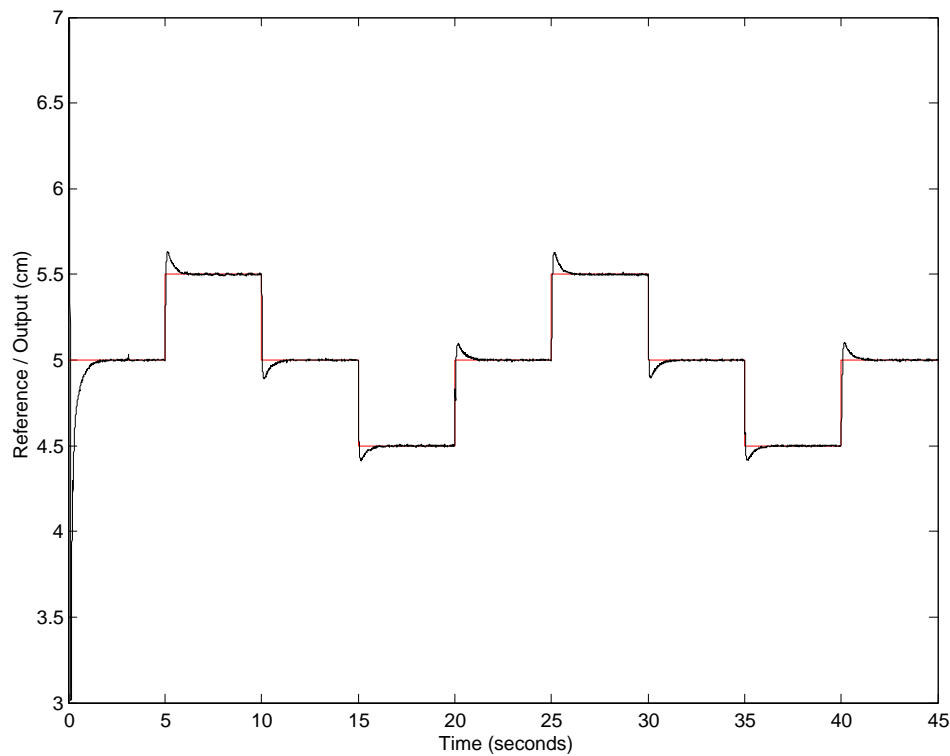


Figure 7.17: Time plots of the experiment.

At the end of the competition almost all the students were able to design a satisfactory controller, and their feedback was really positive.

After an evaluation process, some conclusions about positive and negative aspects of this experience were drawn:

positive aspects: students seemed to be very interested and excited, and used this tool to put in practice many theoretical notions. Moreover, everyone tried to do his/her best to obtain a good position in the ranking. However, the real motivation for this kind of competition, is not to individuate a winner, but to give students a new tool which can help them to better understand some practical control design issues as well as to increase their interest about control systems and technology.

negative aspects: after a first phase when students were really involved in learning new tools for designing a good controller, many students spent


Magnetic Levitation Control of a magnetic levitation system. 					
Competition on Overshoot and Settling Time Overall = Settling Time + (5 * Overshoot)					
Rank	Controller	Overall	Settling Time	Overshoot	
1	Nonlinear Controller	1.127	0.392	0.147	Show
2	PID Controller	1.253	0.548	0.141	Show
3	P.I.D. Controller	1.321	0.601	0.144	Show
4	Lead-Leg Compensator	1.412	0.657	0.151	Show
5	PID Controller	1.789	0.874	0.183	Show

Figure 7.18: Rank position of the controller.

Rank:	3
Nickname:	nick17
Country:	Italy
Institution:	University of Siena
Controller Type:	P.I.D. Controller
Controller Description:	P.I.D. Controller based on linearized model around $y=5$.
Sample Time:	1 ms
Experiment Date - Time:	2002-02-25 - 12:18:00
Overall Index:	1.321
Settling Time	0.601
Overshoot	0.144
Show Experiment	

Figure 7.19: Controller report.

plenty of time to tune controller parameters just to obtain the best controller in the ranking, without any additional educational improvement.

7.5 Remote system identification

In addition to the remote control of processes, ACT allows users to perform system identification experiments. By means of this feature it is possible for a user to choose the input signals to use during the experiment and find a mathematical model of it through a special graphical interface.

It is possible to perform statistic as well as deterministic identification. Regarding statistic identification, the user can select among three models: *ARX*, *ARMAX* and *Output Error*, while concerning set-membership identification one can identify *ARX* models. For a thorough treatment on features of these models see [41].

7.5.1 An identification session description

At the present stage it is possible to perform identification experiments for the DC motor process. The aim of this task is to find a model of suitable order which approximates the real plant. In this case, the input of the process is the DC motor voltage, while the output is the axis velocity.

From the experiment page (Fig. 7.2), the user can choose the *Identification Experiment* option so that the *Identification Control Type Interface* appears as shown in Fig. 7.20.

From this interface it is possible to select the input to apply choosing among a set of predefined inputs (e.g. white noise, sinusoidal and square waves, etc.) or to design a user-defined one by means of a Simulink model just like described in Section 7.3.1 about the user-defined controller.

Identification of a DC Motor

Identification of the speed of a DC motor.

Personal Data

Name	Country	Email
<input type="text" value="Marco Casini"/>	<input type="text" value="Italy"/>	<input type="text" value="casini@ing.unisi.it"/>

Predefined Inputs

Download Model	Standard Inputs (selectable) Sample Time: 10 ms	Run Experiment
--------------------------------	---	--------------------------------

User-defined Inputs

?	Inputs Model	<input type="text"/> <input type="button" value="Sfoglia..."/>
?	Inputs Data	<input type="text"/> <input type="button" value="Sfoglia..."/>
?	Sample Time (msec)	<input type="text"/> Range=[10,100]
?	Download Template	Send Model

Figure 7.20: The *Identification Control Type Interface*.

Once chosen the input signals to use, the *Identification Experiment Interface* will appear as shown in Fig. 7.21.

It is now possible to start the experiment, and to change the applied input signals on-line. Different from the case when control experiments are performed, the user is not allowed to change control parameters in the identification stage.

At the end of the experiment the *Identification Panel* will appear (Fig. 7.22).

Through this interface, it is possible to perform several operations in order to obtain an identified model. In the left side of this interface the time plot as well as the periodogram of the input/output signals are visualized. In the bottom side it is possible to download the Matlab workspace file containing the experiment signals, in order to perform off-line analysis. The right side contains the actual identification interface, which allows the user to choose the model structure of the identified model and the order of the related polynomial. Moreover, it is possible to choose to remove mean and trend as well as to choose how many measurements are used to perform the identification and how many for validation purposes. Once all options are set, it is possible to start the identification experiment by pressing the *Start Identification* button.

At this point, the *Identification Result Interface* will appear as shown in Fig. 7.23 concerning the statistic identification of an ARX model. In the left side of this interface it is possible to see the fitting plot (Fig. 7.24), that is the real validation output compared with the model output, and a frequency response of the estimated model.

In the right side, the coefficients of the identified model are reported as well as three benchmark functions (Loss function, FPE and FIT). The user can now decide to stop his/her identification experiment or to try a new model on the same data.

For a complete identification example, see Section 6 in Part 1 of this thesis.

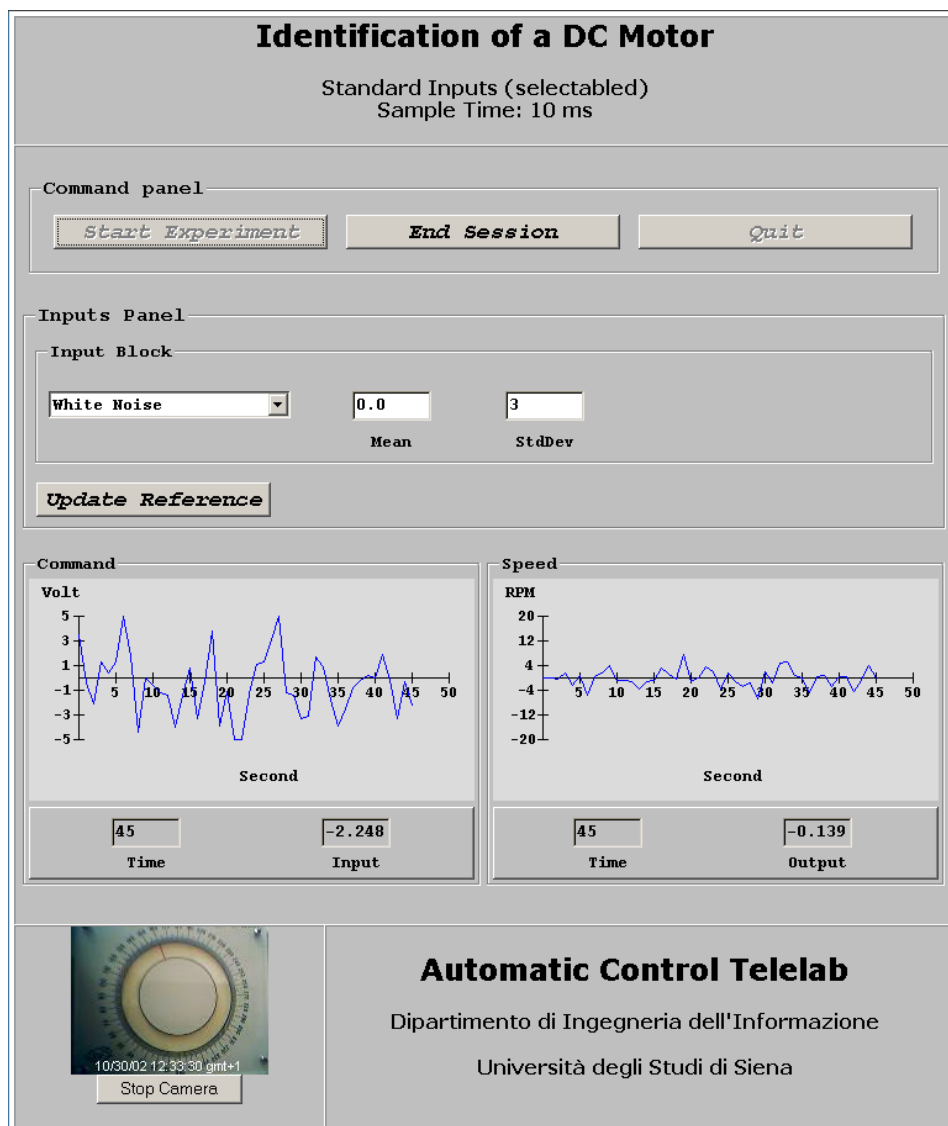


Figure 7.21: The *Identification Experiment Interface* which allows the real execution of remote identification experiments on the DC motor process.

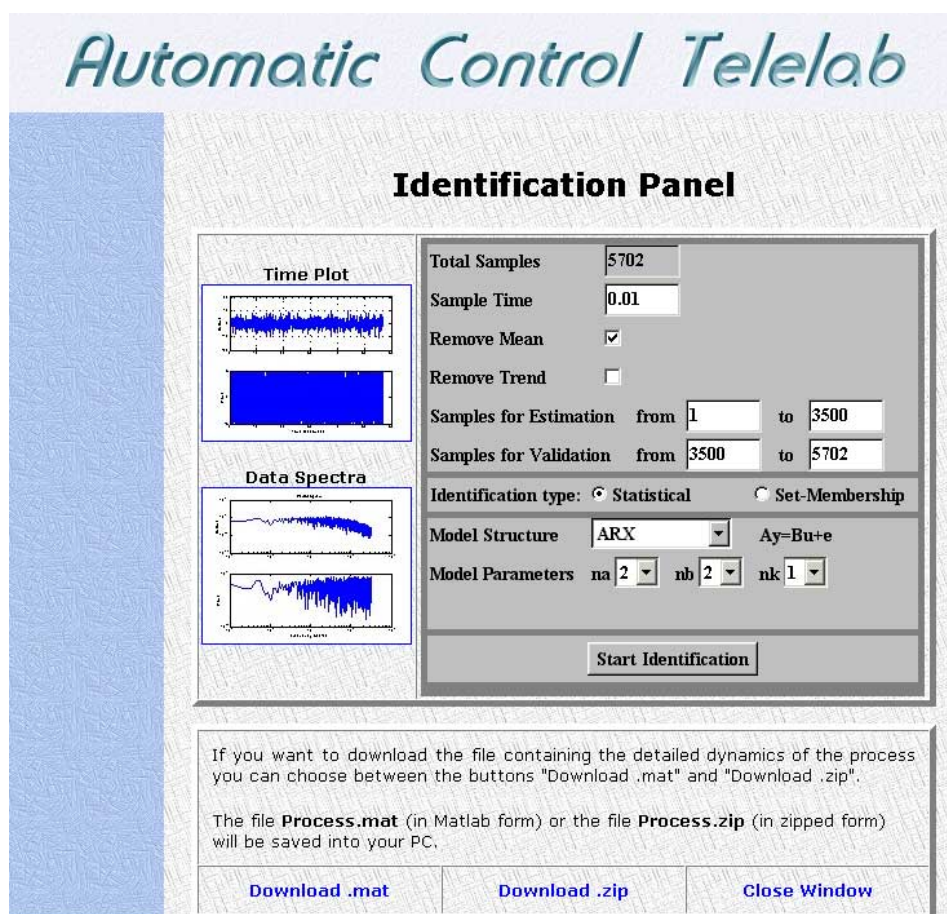


Figure 7.22: The *Identification Panel* which allows a user to choose several identification parameters, such as the model type and order.

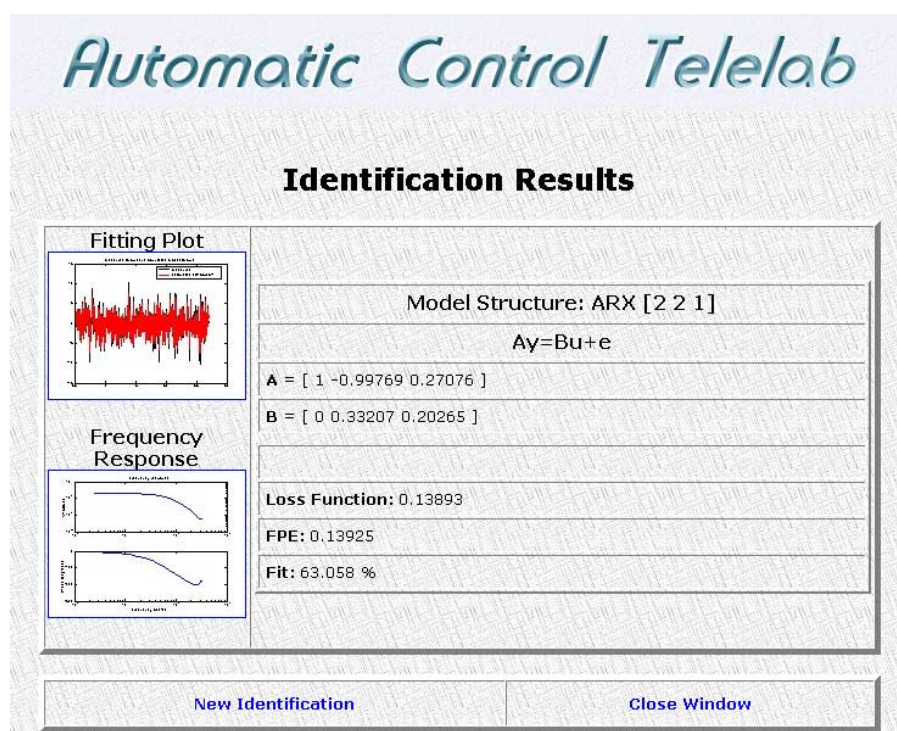


Figure 7.23: The *Identification Result Interface* which shows the results of the identification procedure.

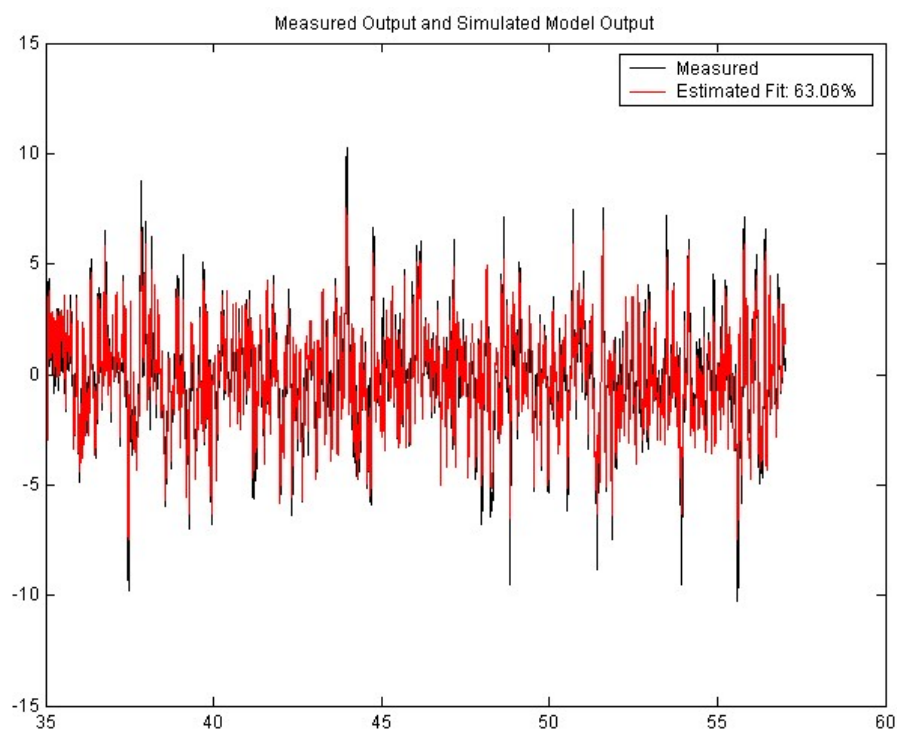


Figure 7.24: The fitting plot of the identification experiment.

7.6 The ACT Architecture

The software architecture consists of two parts: one concerns the control of the physical process (server side) and the other relates to the user interface (client side). Clients will connect to a general web server which contains all general purpose information. Once chosen an experiment, this server will redirect the connection to the pc directly connected to the process, in order to perform the experiment. A sketch of this framework is reported in Fig. 7.25.

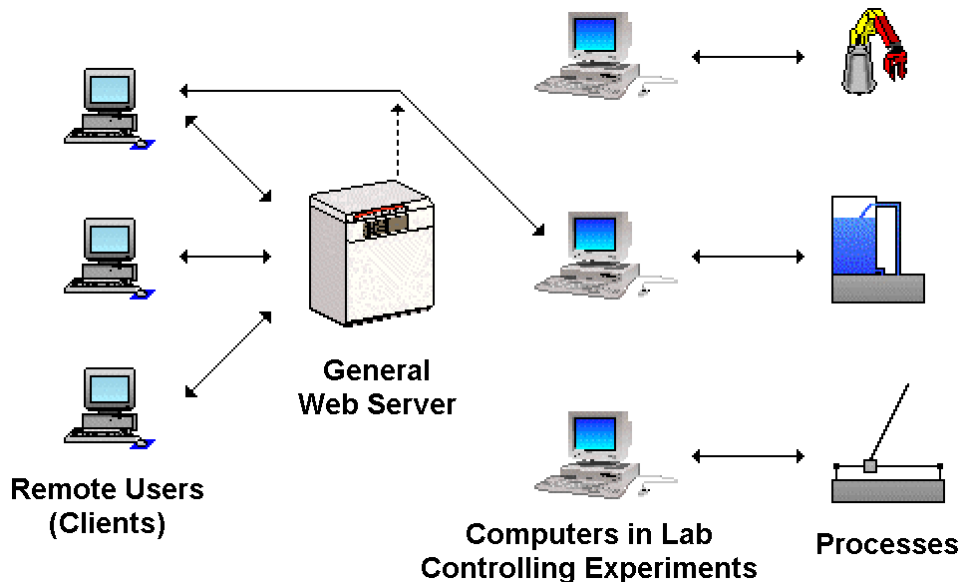


Figure 7.25: Client-server general scheme.

The ACT server runs on the Microsoft Windows 2000 platform and is based on the Matlab/Simulink environment. Such an environment allows the user to design his/her own controller through a Simulink model. The steps necessary to get the executable file from a controller model are shown in Fig. 7.26.

The first phase consists in merging the user defined controller (*control.mdl*) with a Simulink model representing the process (*source.mdl*). Once the output model (*process.mdl*) has been obtained, the Matlab Real Time Workshop (RTW) routine performs the conversion of the Simulink model in a C source

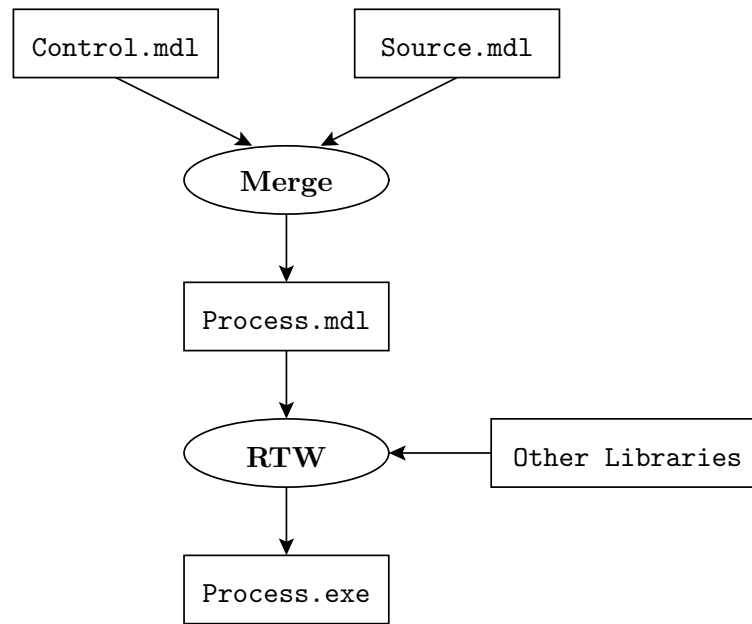


Figure 7.26: Integration of the user-defined controller.

file which is compiled to get an executable code. The compilation task is integrated with some libraries which allow the executable file to perform special functions such as communications with the user and the real-time control of the process.

The client side is essentially based on HTML pages and Java applets to guarantee the maximum portability across various platforms. The home page and other descriptive pages are simply static HTML pages. The *Control Type Interface* in Fig. 7.3, which changes with the chosen experiment, has been implemented as a dynamic page through the use of PHP language. The integration of the user-defined controller in executable code, is handled by another PHP script. All the data about experiments, user access and controllers are store in a MySQL database. In addition to these, all data about student competitions are store in this MySQL database as well.

The overall software architecture has been summarized in the block diagram of Fig. 7.27. Once the executable file *process.exe* has been compiled, the *Experiment Interface* window pops up onto the client machine. This interface is

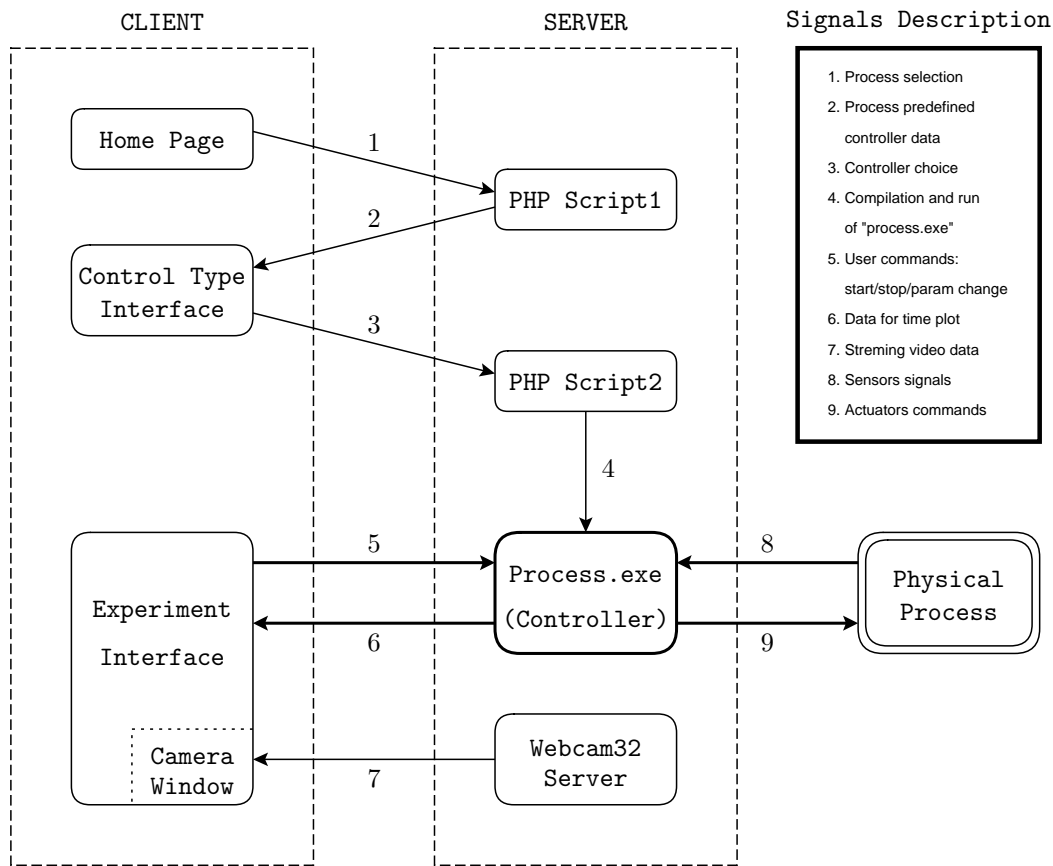


Figure 7.27: The Automatic Control Telelab software architecture.

a Java applet which allows the user to communicate with *process.exe* through a TCP connection. Through this connection it is possible to change the references and the controller parameters on-line, and to send the experimental data over the Internet. A webcam is used to send streaming video to the remote user. The used video software package is the *Webcam32* [69].

As illustrated in Fig. 7.27, the controller (*process.exe*) resides on the computer connected to the process, allowing the safe execution of the experiment despite network delays.

Finally, it is worthwhile to stress again that the software architecture of the ACT has been designed with the goal of simplifying the upgrading procedures and the connection of new processes to the remote laboratory.

Bibliography

- [1] H.S. Witsenhausen. Sets of possible states of linear systems given perturbed observations. *IEEE Transactions on Automatic Control*, 13:556–558, 1968.
- [2] F.C. Schweppe. Recursive state estimation: unknown but bounded errors and system inputs. *IEEE Transactions on Automatic Control*, 13:22–28, 1968.
- [3] J.F. Traub, G.W. Wasilkowski, and H. Woźniakowski. *Information-based Complexity*. Academic Press, San Diego, 1988.
- [4] M. Milanese. Properties of least squares estimates in set membership identification. *Automatica*, 31(2):327–332, 1995.
- [5] M. Milanese and A. Vicino. Information-based complexity and non parametric worst-case system identification. *Journal of Complexity*, 9:427–446, 1993.
- [6] A. Vicino and G. Zappa. Sequential approximation of feasible parameter sets for identification with set membership uncertainty. *IEEE Transactions on Automatic Control*, 41:774–785, 1996.
- [7] L. Chisci, A. Garulli, A. Vicino, and G. Zappa. Block recursive parallelotopic bounding in set membership identification. *Automatica*, 34(1):15–22, 1998.

- [8] P.M. Mäkilä, J.R. Partington, and T.K. Gustafsson. Worst-case control relevant identification. *Automatica*, 31(12):1799–1819, 1994.
- [9] M. Milanese and A. Vicino. Optimal estimation theory for dynamic systems with set membership uncertainty: an overview. *Automatica*, 27(6):997–1009, 1991.
- [10] A.G. Marchuk and K.Y. Oshipenko. Best approximation of functions specified with an error at a finite number of points (in russian). *Mat. Zametki*, 17:359–368, 1975. English Transl.: *Math. Notes* 17 (1975) pp. 207–212.
- [11] M. Milanese and R. Tempo. Optimal algorithms theory for robust estimation and prediction. *IEEE Transactions on Automatic Control*, 30(8):730–738, 1985.
- [12] M. Milanese and G. Belforte. Estimation theory and uncertainty intervals evaluation in presence of unknown but bounded errors. Linear families of models and estimators. *IEEE Transactions on Automatic Control*, 27:408–412, 1982.
- [13] B.Z. Kacewicz, M. Milanese, R. Tempo, and A. Vicino. Optimality of central and projection algorithms for bounded uncertainties. *Systems and Control Letters*, 8:161–171, 1986.
- [14] M. Milanese and A. Vicino. Estimation theory for nonlinear models and set membership uncertainty. *Automatica*, 27:403–408, 1991.
- [15] J.G. Ecker. Geometric programming: methods, computations and applications. *SIAM Rev.*, 1:339–362, 1980.
- [16] J.E. Falk. Global solutions of signomial programs. Technical Report T-274, George Washington University, Washington DC, 1973.
- [17] M. Milanese and A. Vicino. Robust estimation and exact uncertainty intervals evaluation for nonlinear models. In *S. Tzafetas, A. Eisingberg*

- and L. Carotenuto, editors, *System Modelling and Simulations*, Elsevier Science, Amsterdam, 1988.
- [18] B.Z. Kacewicz, M. Milanese, and A. Vicino. Conditional optimal algorithms and estimation of reduced order model. *Journal of Complexity*, 4:73–85, 1988.
- [19] A. Garulli, A. Vicino, and G. Zappa. Conditional central algorithms for worst case set-membership identification and filtering. *IEEE Transactions on Automatic Control*, 45(1):14–23, 2000.
- [20] P.M.J. Van den Hof, P.S.C. Heuberger, and J. Bokor. System identification with generalized orthonormal basis functions. *Automatica*, 31(12):1821–1834, 1995.
- [21] B. Wahlberg and P.M. Mäkilä. On approximation of stable linear dynamical systems using laguerre and kautz functions. *Automatica*, 32(5):693–708, 1996.
- [22] L. Giarrè, B.Z. Kacewicz, and M. Milanese. Model quality evaluation in set membership identification. *Automatica*, 33(6):1133–1139, 1997.
- [23] A. Garulli, B. Kacewicz, A. Vicino, and G. Zappa. Error bounds for conditional algorithms in restricted complexity set membership identification. *IEEE Transactions on Automatic Control*, 45(1):160–164, 2000.
- [24] A. Garulli, A. Vicino, and G. Zappa. Properties of conditional algorithms in restricted complexity set-membership estimation. Technical report, Dipartimento di Ingegneria dell’Informazione, Università di Siena, Italy, 1997.
- [25] P.S.C. Heuberger and O.H. Bosgra. Approximate system identification using system based orthonormal functions. In *Proc. of 29th IEEE Conference on Decision and Control*, pages 1086–1092, Honolulu, HI, 1990.

- [26] B. Wahlberg. System identification using Laguerre models. *IEEE Transactions on Automatic Control*, 36(5):551–562, 1991.
- [27] G.J. Glowes. Choice of the time-scaling factor for linear system approximations using orthonormal Laguerre functions. *IEEE Transactions on Automatic Control*, 10:487–489, 1965.
- [28] J.J. King and T. O’Canainn. Optimum pole positions for Laguerre function models. *Electron. Lett.*, 5:601–602, 1969.
- [29] L. Wang and W.R. Cluett. Optimal choice of time-scaling factor for linear system approximations using Laguerre models. *IEEE Transactions on Automatic Control*, 39:1463–1467, 1994.
- [30] C. Bruni. Analysis of approximation of linear and time-invariant systems pulse response by means of Laguerre finite term expansion. *IEEE Transactions on Automatic Control*, 9:580–581, 1964.
- [31] M. Schetzen. Asymptotic optimum Laguerre series. *IEEE Transactions on Circuit Theory*, 18:493–500, 1971.
- [32] M.A. Masnadi-Shirazi and N. Ahmed. Optimal Laguerre networks for a class of discrete-time systems. *IEEE Transactions on Signal Processing*, 39:2104–2108, 1991.
- [33] T. Oliveira e Silva. On the determination of the optimal pole position of Laguerre filters. *IEEE Transactions on Signal Processing*, 43(9):2079–2087, 1995.
- [34] T.W. Parks. Choice of time scale in Laguerre approximations using signal measurements. *IEEE Transactions on Automatic Control*, 16:511–513, 1971.
- [35] Y. Fu and G. Dumont. An optimum time scale for discrete Laguerre network. *IEEE Transactions on Automatic Control*, 38(6):934–938, 1993.

- [36] B. Wahlberg. System identification using Kautz models. *IEEE Transactions on Automatic Control*, 39(6):1276–1282, 1994.
- [37] P.S.C. Heuberger. *On approximate system identification with system based orthonormal functions*. PhD thesis, Delft University of Technology, The Netherlands, 1991.
- [38] P.S.C. Heuberger, P.M.J. Van den Hof, and O.H. Bosgra. A generalized orthonormal basis for linear dynamical systems. *IEEE Transactions on Automatic Control*, 40:451–465, 1995.
- [39] M. Casini, A. Garulli, and A. Vicino. On worst-case approximation of feasible system sets via orthonormal basis functions. In *Proc. of 40th IEEE Conference on Decision and Control*, pages 2695–2700, Orlando, 2001.
- [40] M. Casini, A. Garulli, and A. Vicino. On worst-case approximation of feasible system sets via orthonormal basis functions. *IEEE Transactions on Automatic Control*, 48(1):96–101, 2003.
- [41] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 1999.
- [42] L. Lin and L.Y. Wang. Time complexity and model complexity of fast identification. In *Proc. of 32nd IEEE Conference on Decision and Control*, pages 2099–2104, San Antonio, TX, 1993.
- [43] G. Zames, L. Lin, and L.Y. Wang. Fast identification n -widths and uncertainty principles for lti and slowly varying systems. *IEEE Transactions on Automatic Control*, 39(9):1827–1838, 1994.
- [44] G. Zames. On the metric complexity of casual linear systems: ϵ -entropy and ϵ -dimension for continuous time. *IEEE Transactions on Automatic Control*, 24(2):222–230, 1979.

- [45] D.A. Harris. Online distance education in the united states. *IEEE Communications Magazine*, pages 87–91, March 1999.
- [46] A. Ausserhofer. Web-based teaching and learning: a panacea ? *IEEE Communications Magazine*, pages 92–96, March 1999.
- [47] S.E. Poindexter and B.S. Heck. Using the web in your courses: What can you do? what should you do? *IEEE Control System*, pages 83–92, Febraury 1999.
- [48] C.M Merrick and J.W. Ponton. The ECOSSE control hypercourse. *Computers in Chemical Engineering*, 20, Supplement:S1353–S1358, 1996. [Online]. Available: www.chemeng.ed.ac.uk/ecosse/control/sample/index.html.
- [49] K.M Lee, W. Daley, and T. McKlin. An interactive learning tool for dynamic systems and control. In *Proc. of International Mechanical Engineering Congress & Exposition*, CA, November 1998.
- [50] C. Schmid. The virtual lab VCLAB for education on the web. In *Proc. of American Control Conference*, pages 1314–1318, Philadelphia, PA, June 1998. [Online]. Available: www.esr.ruhr-uni-bochum.de/VCLab/.
- [51] W.E. Dixon, D.M. Dawson, B.T. Costic, and M.S. de Queiroz. Towards the standardization of a Matlab-based control systems laboratory experience for undergraduate students. In *Proc. of American Control Conference*, pages 1161–1166, Arlington, VA, June 2001.
- [52] C.D. Knight and S.P. DeWeerth. World wide web-based automatic testing of analog circuits. In *Proc. of 1996 Midwest Symposium Circuits and Systems*, pages 295–298, August 1996.
- [53] M. Shaheen, K. Loparo, and M. Buchner. Remote laboratory experimentation. In *Proc. of American Control Conference*, pages 1314–1318, Philadelphia, PA, June 1998.

- [54] J. Henry. *Engineering lab on line*. University of Tennessee at Chattanooga, 1998. [Online]. Available: chem.engr.utc.edu.
- [55] B. Aktan, C.A. Bohus, A. Crowl, and M.H. Shor. Distance learning applied to control engineering laboratories. *IEEE Transactions on Education*, 39(3):320–326, August 1996.
- [56] A. Bhandari and M. Shor. Access to an instructional control laboratory experiment through the world wide web. In *Proc. of American Control Conference*, pages 1319–1325, Philadelphia, PA, June 1998.
- [57] J. W. Overstreet and A. Tzes. An internet-based real-time control engineering laboratory. *IEEE Control Systems Magazine*, 19(5):19–34, October 1999.
- [58] J. Zhang, J. Chen, C.C. Ko, B.M. Chen, and S.S. Ge. A web-based laboratory on control of a two-degree-of-freedom helicopter. In *Proc. of Conference on Decision and Control*, pages 2821–2826, Orlando, FL, December 2001.
- [59] V. Ramakrishnan, Y. Zhuang, S.Y. Hu, J.P. Chen, C.C. Ko, Ben M. Chen, and K.C. Tan. Development of a web-based control experiment for a coupled tank apparatus. In *Proc. of American Control Conference*, pages 4409–4413, Chicago, IL, June 2000. [Online]. Available: vlab.ee.nus.edu.sg/vlab/control/.
- [60] T.F. Junge and C. Schmid. Web-based remote experimentation using a laboratory-scale optical tracker. In *Proc. of American Control Conference*, pages 2951–2954, Chicago, IL, June 2000.
- [61] J. Apkarian and A. Dawes. Interactive control education with virtual presence on the web. In *Proc. of American Control Conference*, pages 3985–3990, Chicago, IL, June 2000. [Online]. Available: www.controlab.com.

- [62] G. Choy, D.R. Parker, J.N. d'Amour, and J.L. Spencer. Remote experimentation: a web-operable two phase flow experiment. In *Proc. of American Control Conference*, pages 2939–2943, Chicago, IL, June 2000.
- [63] M. Exel, S. Gentil, F. Michau, and D. Rey. Simulation workshop and remote laboratory: two web-based training approaches for control. In *Proc. of American Control Conference*, pages 3468–3472, Chicago, IL, June 2000.
- [64] M. Casini. *Designing a Tele Laboratory for control of dynamic systems through Internet*. Master thesis (in italian), Università degli Studi di Siena, June 1999.
- [65] M. Casini, D. Prattichizzo, and A. Vicino. The Automatic Control Telelab: a remote laboratory of automatic control. In *Proc. of 40th IEEE Conference on Decision and Control*, pages 3242–3247, Orlando, Florida, December 2001.
- [66] M. Casini, D. Prattichizzo, and A. Vicino. The automatic control telelab: a user-friendly interface for distance learning. *IEEE Transactions on Education*, 2003.
- [67] S. Dormido. Control learning: present and future. In *15th IFAC World Congress b'02*, Barcelona, July 2002.
- [68] M. Casini, D. Prattichizzo, and A. Vicino. E-learning by remote laboratories: a new tool for control education. In *Preprints of 6th IFAC Symposium on Advances in Control Education*, Oulu, Finland, June 2003.
- [69] N. Kolban. Webcam32 - the ultimate webcam software. [Online]. Available: surveyorcorp.com/webcam32.