Laboratory of Robust Identification and Control

Lecture notes

A. Ayanmanesh Motlaghmofrad

A.Y. 2024/2025

Contents

Chapter 1

Introduction

Disturbances and uncertainties

The key word *Robust* suggests that we are taking uncertainties into account. The model of the plant is as following:

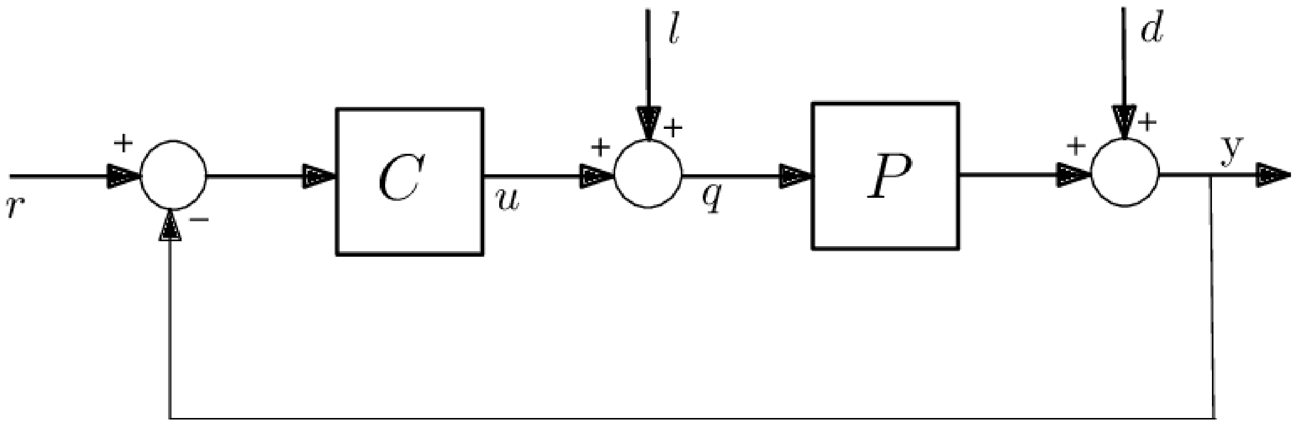


Figure 1.1: A general control plant with additive disturbances l and d

The general nonlinear state-space representation of the system is:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = g(x(t), u(t)) \end{cases} \tag{1.1}$$

Where:

- $x(t)$ is the state vector,
- $u(t)$ is the input vector,
- $y(t)$ is the output vector,
- $f(\cdot)$ is a nonlinear function describing the system dynamics,
- $g(\cdot)$ is a nonlinear function describing the output equation.

The first step of any control problem is typically **derivation of mathematical model of the plant.** This step is the most crucial step, because if we drive the model by applying first principles of physics, we are likely to adopt approximated models, adopting simplifying assumptions, e.g. rigid body assumption etc. Further, the value of the physical parameters involved in the equations, such as friction coefficient, are not exactly known. Such approximations introduce errors and uncertainty in the mathematical description of the plant to be studied and controlled.

This fact is critical, since standard approaches to controller design are model-based; that is, the controller design has a strong dependency on the mathematical models used to describe the plant to be studied and controlled.

Neglecting some physical details $\equiv$ ***Neglecting some state variables***

For example, for modelling a robotic arm, generally, rigidity is assumed for the joints. Nevertheless, in fast movements this assumption does not hold anymore, and the model does not predict the real performance of the robot, neglecting some state variables. Further, in some applications, we are not even aware of the phenomenon or phenomena that is being neglected.

Counter act for disturbances and uncertainties

- **First counter act** These uncertainties and disturbances directly affect the controller design in the time domain, since the feedback gain and observer are directly calculated by solving algebraic equations including physical parameters with uncertainties. Nontheless, In frequency domain, the design of the controller is less affected by these uncertainties.

  This does not mean that *Transfer Function* is not affected by uncertainties of the parameters, because not considering some phenomona leads to the transfer function having less poles or zeros and because the uncertainty of the parameters affect the coefficients of the complex variables, being s or z. In the frequency domain design, we design the controller based on the frequency response of the system, considering cutting frequencies that reject high-frequency and low-frequency disturbances. In addition, by considering phase-margin and gain-margin, some margin for disturbances and uncertainties are taken into accoutn.

  Question to be answered: the sensitivity function is high-pass filter, meaning that the high frequency phenomenon, on the other-hand T is a low-pass filter, if the plant is designed for having a fast rising time, high-frequency phenomena also passes T.

- **Second counter act**: Optimization problems in state-space where introduced to tackle this problem.

- **Third counter act**: Optimization problem, in state space, is combined with the concept of robustness, in frequency space, which is called H_∞.

What is going to be discussed in this course

In the first part of this course, we are going to learn how to learn from the mapping from the input to the output of the system, extracting a **mathematical model** for the plant. Further, it elaborates on this data to drive also a **discription for uncertainty**. These, together, can be used for **robust controller design** in a model-based approach.

In the second step, the aim is to design a controller directly from the data, without the intervention of the model.

Professor's Quote

In conclusion, even if the physical model is very precise, at some point, in order to measure the parameters used in the physical models, it is required to do some experimental measurements, subjected to noise, introducing uncertainties to our model.

If these uncertainties are not taken into account, the controller will not have a good performance when implemented physically, and it is going to work only in simulations.

There are many approaches to tackle this problem. Here, in the first part of the course, we will focus on **System Identification**, which is another modelling paradigm. In this paradigm, we learn the mathematical model of the system to be controlled by using experimentally collected data. Not only do we introduce what *System identification* is and what are the possible approached, but maintly to focus on ***Set-membership Identification Technique*** that allow us to learn the model of the system, and drive information about how the uncertainty is affecting out model; this is used to design a controller in a robust manner. Robust controller designed can be directly applied to the real physical plants.

In the second part of the course, we shift our paradigm again. In this part, assuming that the collected data represents the behavior of the mapping between the input and output, we try to design the controller directly from the data, called **Direct Data-Driver Controller Design**.

System identification deals with the problem of building mathematical model of dynamical systems from sets of experimentally collected input, output data.

There are three different approaches to mathematically model dynamical systems:

1. **White-box modelling**:
   Models, in this approach, are obtained by applying **first principles for physics**. All

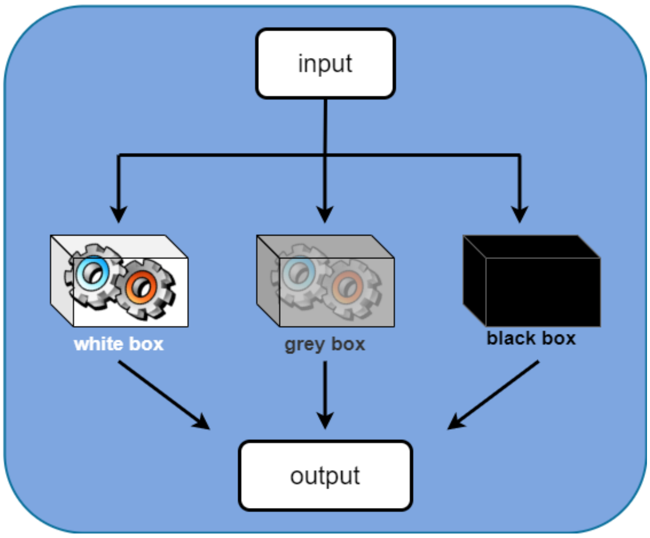


Figure 1.2: Classifications of System Identification Problems

the physical phenomena and also all the physical parameters involved in the equiation are assumed to be exactly know. This approach was applied in the course of *Automatic Control* for modelling systems.

2. **Grey-box modelling**: Here, **models** are base on the equations obtained by applying first principles of **physics**, but **the parameters** entering the equations are not completely known, so they need **to be estimated** from experimental data.

3. **Black-box modelling**: In this case, the structure of the equations is selected by the user on the basis of some **"general" a-priori information**, e.g. linearity, on the system physics, or at any rate by the system properties.

Professor's Quote

In this method, the designer of the system model, has some degree of freedom in selecteing the structure of the model, provided that the model embeds the "general" a-priori information.

If we do not have any a-priori information, an artificial-neural-network may be selected as the structure of the model.

the parameters involved in the equations of the black-box models are **to be estimated**, or computed, by using experimentally collected data. In general, **the parameters of a black-box model do not have a clear physical meaning**.

In general, **white-box models are not very useful in practice**, at least for control applications, because they are based on the assumption that the physics involved in the system under study is well-known.

The comparison between the grey-box and Black-box models:

Similarity In both cases:

- physical insight is exploited to drive/select the structure of the equations.
- experimentally collected data is used to estimate/compute the parameters involved in the equations.

Difference

- In grey-box modelling, the structure of the equations is not selected by the user, since it is forced by the first principles of physics. This suggests that, in general, the equations of a grey-box model depends, in a possibly complex, non-linear ways, on the physical parameters to-be-estimated.

  for instance of a grey-box model: $\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$ where:

  $$A = \begin{bmatrix} \frac{m}{k^2} & \sqrt{\beta} \\ \frac{\alpha^2 . k^3}{\gamma} & 1 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad C = \begin{bmatrix} c_1 & c_2 \end{bmatrix}, \quad D = \begin{bmatrix} d \end{bmatrix}$$

  In this case, using the physical principles for modelling seems like a good idea; the system is simple, and its physics is well-known. The problem is that the parameters are to-be-estimated. This problem leads to an optimization problem. Now, A difficulty may arise, because if the equation we are going to write so that they relates input and output data obtained experimentally depends on parameters in a complex and non-linear fashion, the mathematical problem of driving what are the correct value of the system parameter satisfying the relation between the intput and output is going to be a complex problem. **This is the main limitation of grey-box model.**

- In the black-box version of the same problem, just some general information such as liniearity and time-invariantion is exploited. Now, we found the four matricies *A, B, C, D* in a form that is much simpller, just 4 parameters for A, and so on, for instance. In this case, the system output depends on the input in a way simpler manner.

  $$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad C = \begin{bmatrix} c_1 & c_2 \end{bmatrix}, \quad D = \begin{bmatrix} d \end{bmatrix}$$

Professor's Quote

Therefore, in black-box model, we have more freedom to select the struture of the equations in a way that is computationally more convenient, by embedding/exploiting some general properties drived from our physical insight. The idea is to consider only the most important a-priori information and/or the information that we trust the most!

General procedure for building a grey/black-box modelling:

Let's compare grey-box and black-box models from the point of view of the parameter estimation problem. While building the mathematical model of the system, in general, we follow a procedure similar to the following one, it be a grey-box or black-box structure:

1. to exploint availabe a-priori information on the system under study to select **the structure of the mathematical equation** describing the input-output mapping.

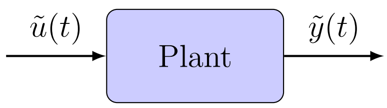
   Professor's Quote

   The model is always in the input/output form. Signals that we can apply to the system is called input, and signals that we can measure, is called output. If we are in the case where we can measure all **state variables** involved in the system, we can have a full description of the system. However, in a general case, the system involves input, output, and state variables, but we are able to measure only a subset of physical variables in the system - being the value to be monitored or the control output.
   Now, these parameters have physical meaning in grey-box case, or are merely mathematical parameters in black-bock case.

   In the end of this step, we have a mathematical model of the following form:

$$y(t) = f\left(u(t), \theta\right) \tag{1.2}$$

2. To collect input-output data representing the behavior of the system under study by performing an experiment. $\tilde{u}$ and $\tilde{y}$ are noise-corrupted data, since, in general, noise can corrupt the output measurements as well as measurement of the input signal applied to the system.



3. To formulate a suitable mathematical problem to estimate/compute the values of the vector of parameters θ in such a way that our mathematical model is going to describe the behavior of real system as well as possible. for example:

$$\hat{\theta} = \arg\min_{\theta \in S} f(\theta) = \arg\min_{\theta} \|\tilde{y} - f(\tilde{u}, \theta)\|_2$$

   J(.) here is a **cost function**. In this case, it is the euclidian norm of $\tilde{y}$ and $f(\tilde{u}, \theta)$.

At the third stage, the difference between grey-box and black-box models comes into play, since:

- In grey-box model: $f(\tilde{u},\theta)$ will, in general depends in a complex, non-linear manner from θ, **leading to multi-minima convex cost function which is really hard to be minimize**

  While

- In black-box model: $f(\tilde{u},\theta)$ will be selected by the user in order to depend linearly from θ,if possible, or in the simplest possible way, **leading to a single-minima convex funciton to be minimized**, which is much easier in comparison.

For example, consider the following multi-minima cost-function:

$$J(\theta_1,\theta_2) = \sin(\theta_1)\cdot\cos(\theta_2) + \frac{\theta_1^2+\theta_2^2}{10}$$

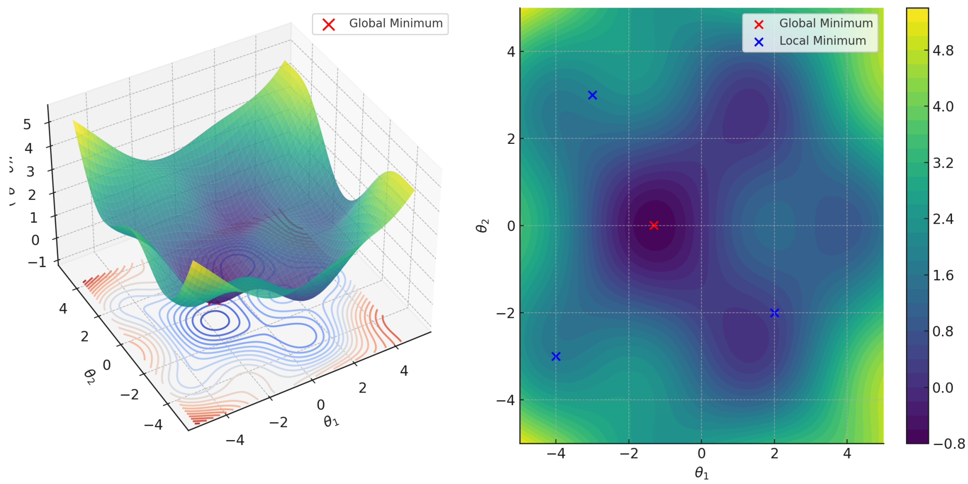


Figure 1.3: The 3D plot and countor plot of the afforementioned cost-function, which is a multi-minima cost-function

In this example, we look for the global minimum of the cost-function. Finding this point, however, is not an easy task, due to the fact that there is no guarantee that our optimization algorithm will not be trapped in one of the local minima, which may correspond to a "very bad" estimation of θ. Even if we find the global minimum by chance, there is no guarantee that output is indeed the global minima.

On the other hand, When a black-box model is considered, a parametrization is considered so that the function $f(u,\theta)$ is a convex function of θ, thereby having a single global minimum. In this case, no matter what optimization algorithm is used, reaching the global minimum is guaranteed.This is evident in the following cost-function:

$J(\theta_1, \theta_2) = \theta_1^2 + \theta_2^2$

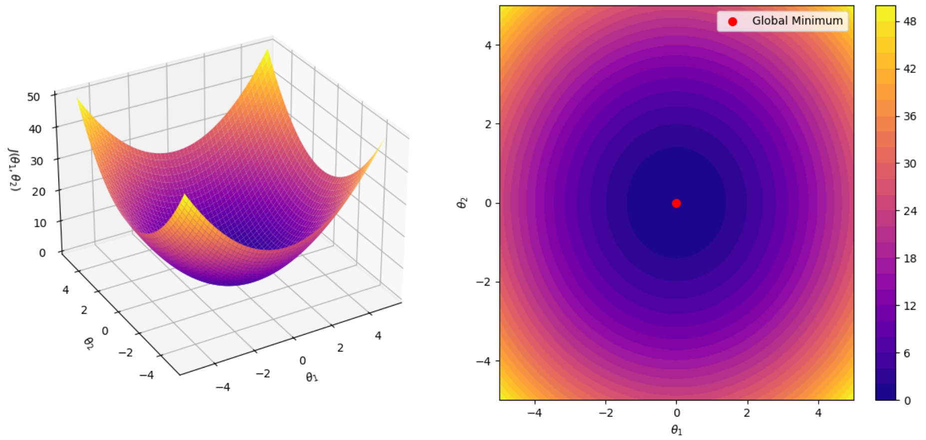


Figure 1.4: The 3D plot and countor plot of the afforementioned cost-function, which is a convex cost-function

Q and A

Question: we use grey-box model when we have an insight about the system and parameters. Hence, when we run the optimization problem, we can initialize the optimization problem with more suitable initial condition. Further, we can neglect some of the estimation, knowing that the estimated value does not correspond to the physical quantity the parameter is expected to have.

Professor's answer: This is true for simple systems. Nonetheless, in such compolex systems as chemical processes or optical lazer systems, it's hard to have an insight before hand about some parameters. In addition, since we are dealling with multi-dimensional problems, it might be the case that we confine the expected value for some parameters but still, for some parameters, we need to deal with the issue mentioned about the multi-minima cost functions. However, it might be the case that the estimation of those physical parameters are of interest, which is not the concern of this course. Here, we do system identification for control purposees.

To clarify the matter further, consider an ***LTI system*** such that the **transfer function** obtained from $H(s) = C(sI - A)^{-1}B + D$ where matrices A, B, C, D are teh state-space matrices obtained by applying the first principles of physics.

$$H(s) = \frac{\frac{P_1}{P_2}s + \frac{P_3}{\sqrt{P_4}}}{s^2 + \frac{P_1.P_2}{P_3^3} + 1}$$

Where P_1, P_2, P_3, P_4 are the physical parameters. Now, in terms of modelling the input-output behavior of the system, we don't miss anything by considering the following transfer function.

$$H(s) = \frac{\theta_1 s + \theta_2}{s^2 + \theta_3 s + \theta_4}$$

In this black-box model, we just take into consideration the most important "general" information that the system is an LTI system - hence being able to be modelled as a transfer function - and of order two. Therefore, we can use a transfer function model for describing the system behavior. State-space models can be obtained from transfer functions by applying basic results on ***realization theory***, which is not going to be discussed in this course.

"Phylosophical Remark"

Pay attention that **we cannot build our model without any assumption about the system**. No matter how much input-output sample is available, we need to have an assuption about the system to be able to descibe it.

In conclusion:
black-box models are thet best choice for the following purposes:

- simulalting the input-output behavior of the system
- modelling for the purpose of designing a controller

Grey-box models, in general, are the best choice when it is desired to:

- estimate the values of some physical parameters

If the physical systeme is well-known, **white-box modelling** is the choice.

Chapter 2

System Identification

Generalities

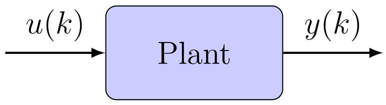
Since experimental measurement procedure typically provides samples of input-output sequences, we start by considering identification of ***discrete-time models***. Procedure for identifying ***continous-time models*** will be dicussed as well.
Here, we are dealling with **dynamical systems**. A system is called dynamic when its output at a given time is dependent on the all the inputs that have been inserted to the system up to that time, or equivalently we know the value of the input at the previous time-instance in and the initial value of the a set of variables called, ***state variables***.

The choice of state variables is not unique, but the number of them depend on the order of the system. It can be proved that **a possible choice for state variables is previous values of the output**.

It is assumed that, without loss of generality, any system, be linear or non-linear, can be modelled by means of the following ***regression form***:

$$y(k) = f(y(k-1), y(k-2), \cdots, y(k-n), u(k), u(k-1), \cdots, u(k-m), \theta_1, \theta_2, \cdots, \theta_{n+m+1}) \quad (2.1)$$

- n is the system order, **related to the number of state variables**
- $m < n$, always for the physical, or at any rate, causal systems.
- $u(k), y(k), k = 1, 2, ..., H$ are the **noise-free** samples of the input and output sequences, respectively.



General black-box EIV set-up

Error-In-variables, or EIV, problems refer to the most general case where both input and output collected samples are affected by noise, as represented in the figure 2.1.
Two a-priori assumption is required at this point:

- a-priori assumption on the model: $f \in F$ where F is a given class of functions
- a-priori assumption on the noise: e.g. statistically distributed, or boundedness, etc

Then, we need to collect input-output data.

> **Without any assumption** about the noise corrupting the measurements, we cannot study the data at hand. Further, without any assumption about the structure, many structure can be find that map input samples to the output, that does not correspond to the behavior of the system.

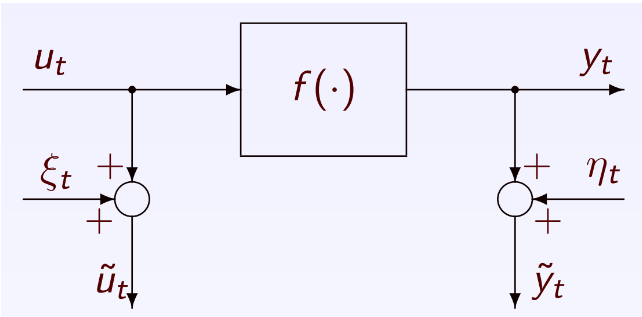


Figure 2.1: a general shceme of error-in-variable problem

A naive noiseless example

Consider the following second order LTI discrete-time system (agent):

$$\begin{aligned} y(k) &= f(y(k-1), y(k-2), u(k), u(k-1), u(k-2)) \\ &= -\theta_1 y(k-1) - \theta_2 y(k-2) + \theta_3 u(k) + \theta_4 u(k-1) + \theta_5 u(k-2) \end{aligned} \tag{2.2}$$

By introducing the backward shift operator q^r: $q^r(s(t)) = s(kr)$, we can rewrite the equation as:

$$y(k) = -\theta_1 q^{-1} y(k) - \theta_2 q^{-2} y(k) + \theta_3 u(k) + \theta_4 q^{-1} u(k) + \theta_5 q^{-2} u(k) \tag{2.3}$$

Solving the equation in the $y(k)$ we obtain:

$$y(k) = \frac{\theta_3 + \theta_4 q^{-1} + \theta_5 q^{-2}}{1 + \theta_1 q^{-1} + \theta_2 q^{-2}} u(k) \tag{2.4}$$

By applying properties of the Z-transform it is possible to show that the system transfer function G(z) can be obtained by simply replacing q^{-1} with z^{-1}:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{\theta_3 z^2 + \theta_4 z + \theta_5}{z^2 + \theta_1 z + \theta_2} u(k) \tag{2.5}$$

Given H experimentally collected input-output samples matrix 2.7, which is also called ***regressor-matrix***, leads to the following system of linear equations:

$$y = A\theta \tag{2.6}$$

where $y = [y(3)y(4)...y(H)]^T, \theta = [\theta_1\theta_2...\theta_5]^T)$, and

$$A = \begin{bmatrix} -y(2) & -y(3) & u(3) & u(2) & u(1) \\ -y(3) & -y(2) & u(4) & u(3) & u(2) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -y(H-1) & -y(H-2) & u(H) & u(H-1) & u(H-2) \end{bmatrix} \tag{2.7}$$

In this ideal, noise-free example, $H =$ the number of parameters + the order of the systerm, in this case, $H = 7$ is enough to solve our 5-equations-and-5-unknown problem. The solution is going to be in the following form :

$$\theta = A^{-1}y \quad \xrightarrow{\text{Existence and Uniqueness Condition}} \quad \begin{cases} A \in \mathbb{R}^{n\times n} \\ |A| \neq 0 \end{cases} \tag{2.8}$$

One possible solution: apply to the system a random input sequence u.

This formulation works for all the description that the output of the system is **linear with respect to the parameters**.

Professor's Quote

In artificial-neural-network, the structure of f is different. Otherwise, the goal of any Machine Learning or System Identification problem is to find the mapping between the inputs and outputs.
Regarding the importance of a-priori info about the system. The previous problem can also be solved for the following structure, that does no correspond to the behavior of a 2-order LTI system. $y(k) = \theta_1 u(k) + \theta_2 u(k)^2 + \cdots + \theta_5 u(k)^5$

Example

It is desired to estimate the value of a resistor. A-priori info is that a resistor follows Ohm's law. As input signal, we insert current to our resistor and we measure the voltage across our resistor.

$y(k) = \theta u(k)$ s.th. y(.),u(.), $\theta \in \mathbb{R}$

Here, θ stands for the value of the resistor to be estimated. For this static system, which means the output of the system depends only on the values of the input at that time instance, we need a pair of noise-free input-output sample to estimate the value of the resistor.

$\theta = \dfrac{y(1)}{u(1)}$

Noise effect

If the measurements are corrupted by noise, the idea is to perform more measurements to average out the effect of the noise; $H >> 2n + 1$ where n is the system order.

In this case, matrix$A$ is not square anymore, which means it is not invertible. In this case, left pseudo-inverse, $(A^T A)^{-1} A^T$ of this matrix should be used.

$$\tilde{y} = A\theta \Rightarrow A^T \tilde{y} = A^T A\theta \Rightarrow \theta = (A^T A)^{-1} A^T \tilde{y}$$

This solution corresponids to the solution of the ***Least Square*** problem.

Least Square approach

Least Square approach is solving the following optimization problem in order to find the parameters that minimizes a norm-2, or euclidian norm, cost-function.

$$\hat{\theta_{LS}} = \arg\min_{\theta} J(\theta) = \arg\min_{\theta} \|\tilde{y} - A\theta\|_2 \tag{2.9}$$

In the context of system identification (but not only) LS as what we refer to as the Least square estimate of the system parameter vector. Computation burden of the Least square algorithms is quite low also for large H. The Least square estimate can be (also) computed recursively (i.e. online).

Consistency property of Least Square method

If the following assumptions are satisfied:

1. The effect of the uncertainties corrupting the collected data can be taken into account by introducing an additive term **e**, or equation error, as follows:

$$\tilde{y}(k) = y(k) + e(k)$$

2. $e(k)$, $k = 1, 2, ..., H$ are **independent and identically distributed** (iid) random variables; typically it is assumed that e can be modeled as a white, zero-mean Gaussian noise.

Least-Square method enjoys the following interesting property:

$$\lim_{H\to\infty} E[\theta_{LS}] = \theta \tag{2.10}$$

Pay attention that, in our assuption, there is no argument about the variance of the noise. This is the reason why LS method is so appealing in practice. Nevertheless, if these two assumtions does not hold, this method is not going to provide suitable results.

This property of the Least Square solution holds only when the afforementioned consistency conditions hold.
Every time, this method is to be applied to an engineering problem, these two assumptions should be satisfied so that the concistency property holds.

As to the first assumption, Let's check whether such an assuption is satisfied when we collect the data by performing a real experiment in its most general form, Error-In-Variable form 2.1. Considering a second-order LTI system - a-priori information about the system - we have:

$$y(k) = -\theta_1 y(k-1) - \theta_2 y(k-2) + \theta_3 u(k) + \theta_4 u(k-1) + \theta_5 u(k-2)$$

Additionally, considering EIV form:

$$\begin{cases} \tilde{y}(k) = y(k) + \eta(k) \\ \tilde{u}(k) = u(k) + \zeta(k) \end{cases}$$

Regarding the noise

It's correct that the same sensor might be used for obtaining data samples - suggensting that the noises are dependent. Nonetheless, we assume that all the systematic error of the sensor is taken into considerations, and $\zeta(.)$ **and** $\eta(.)$ **are undeterministic noises**.

Also, here, an **absolute error** is considered which is the form:
$\tilde{y}(.) = y(.) + \eta(.)$

In case of **relative error**, the following manipulations may be done which leads to a mathematically equivalent structure.
$\tilde{y}(.) = y(.)(1+\eta(.))$
Also in this case, the same result is going to be obtained.

By replacing $y(k)$ and $u(k)$ in the difference equation of the system the following is obtained:

$$\tilde{y}(k) - \eta(k) = -\theta_1(\tilde{y}(k-1)) - \eta(k-1) - \theta_2(\tilde{y}(k-2) - \eta(k-2)) + \theta_3(\tilde{u}(k) + \xi(k))$$
$$+\theta_4(\tilde{u}(k-1) + \xi(k-1))\theta_5(\tilde{u}(k-2) + \xi(k-2))$$
$$\Rightarrow$$
$$\tilde{y}(k) = -\theta_1\tilde{y}(k-1) - \theta_2\tilde{y}(k-2) + \theta_3\tilde{u}(k) + \theta_4\tilde{u}(k-1) + \theta_5\tilde{u}(k-2)$$
$$\boxed{+\theta_1\eta(k-1) + \theta_2\eta(k-2) - \theta_3\xi(k) - \theta_4\xi(k-1) - \theta_5\xi(k-2) + \eta(k)} =: +e(k)$$

Hence:

$$\tilde{y}(k) = -\theta_1\tilde{y}(k-1) - \theta_2\tilde{y}(k-2) + \theta_3\tilde{u}(k) + \theta_4\tilde{u}(k-1) + \theta_5\tilde{u}(k-2) + \mathbf{e(k)} \tag{2.11}$$

As it can be seen, considering the afformentioned assumption about noise and system, **the first assumption of the concistency property is hold!**.

As to the second assumption, it is needed to check whether the sequence $e(k)$, $k = 1, 2, \cdots, H$ is i.i.d, independent and identically distributed. To do so, the value of $e(k)$ is written for two time instances k and $k+1$:

$$e(k) = \theta_1\eta(k-1) + \theta_2\eta(k-2) - \theta_3\tilde{u}(k) - \theta_4\tilde{u}(k-1) - \theta_5\tilde{u}(k-2)$$
$$e(k+1) = \theta_1\eta(k) + \theta_2\eta(k-1) - \theta_3\tilde{u}(k+1) - \theta_4\tilde{u}(k) - \theta_5\tilde{u}(k-1)$$

As it can be seen, since both $e(k+1)$ and $e(k)$ depend on common variables, the sequence $e(.)$ is not i.i.d; **the second is not satisfied**.

In conclusion, Least-Square estimation applied to $\tilde{u}(.)$ and $\tilde{y}(.)$, collected from an experiment enjoying the EIV structure provide and estimation, $\hat{theta}_{LS}$ which does not enjoy consistency property. That is,

$$\lim_{H\to\infty} E[\theta_{LS}] \neq \theta$$

Results:
Least-Square estimation **DOES NOT** enjoy the consistency properties when applied to input-output data obtained by performing an experiment on a dynamicsl system in the presence of an uncertainty entering the problem according to the *EIV* or *Output-Error* setting.

In which cases, assuption 1 and 2 are both satisfied?

$$\begin{cases} \text{Assuption 1:} \\ \qquad \tilde{y}(k) = -\theta_1 \tilde{y}(k-1) - \theta_2 \tilde{y}(k-2) - \cdots - \tilde{y}(k-n) \\ \qquad +\theta_{n+1}\tilde{u}(k) + \theta_{n+2}\tilde{u}(k-1) + \cdots + \theta_{n+m+1}\tilde{u}(k-m) + \mathbf{e(k)} \\ \text{Assumption 2:} \\ \qquad e(.) \text{ is an i.i.d variable} \end{cases}$$

Let's consider, for the sacke of simplicity and without loss of generality, that the inputs are perfectly known.

$$\tilde{u}(.) = u(.)$$

$$\tilde{y}(k) = -\theta_1 \tilde{y}(k-1) - \theta_2 \tilde{y}(k-2) - \cdots - \tilde{y}(k-n)$$
$$+\theta_{n+1}u(k) + \theta_{n+2}u(k-1) + \cdots + \theta_{n+m+1}u(k-m) + e(k)$$

$$\Rightarrow$$

$$\tilde{y}\left[\theta_{n+1} + \theta_{n+2}q^{-1} + \cdots + \theta_{n+m+1}q^{-m}\right] == u(k)\left[1 + \theta_1 q^{-1} + \theta_2 q^{-2} + \cdots + \theta_n q^{-n}\right] + e(k)$$

$$\Rightarrow$$

$$\tilde{y} = \frac{[1 + \theta_1 q^{-1} + \theta_2 q^{-2} + \cdots + \theta_n q^{-n}]}{[\theta_{n+1} + \theta_{n+2}q^{-1} + \cdots + \theta_{n+m+1}q^{-m}]}u(k) + \frac{1}{[\theta_{n+1} + \theta_{n+2}q^{-1} + \cdots + \theta_{n+m+1}q^{-m}]}e(k)$$

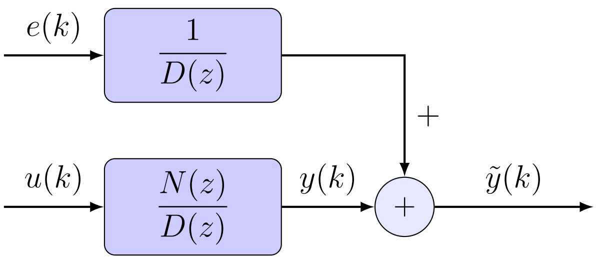
Now, considering that the equivalent of delay-operator in the z-domain is z:

$$\tilde{y} = G(z)u(z) + \frac{1}{D(z)}e(z)$$

where,

$$G(z) = \frac{N(z)}{D(z)} \begin{cases} N(z) = \left[1 + \theta_1 q^{-1} + \theta_2 q^{-2} + \cdots + \theta_n q^{-n}\right] \\ D(z) = \left[\theta_{n+1} + \theta_{n+2}q^{-1} + \cdots + \theta_{n+m+1}q^{-m}\right] \end{cases}$$

The block diagram scheme of the abovementioned equation is drawn hereunder.



LS estimation is going to make sense, if and only if the sensor used for collecting data affect the collected data with a measurement noise obtained as a random, white signal $e(k)$ that is filtered by the denominator of the system to be identified, **which does not make sense**!!!; the name of this setting is ***Error-in-Equation*** setting, which does not correspond to data collection in practical situation, since the sensor does not know how to filter the noise to obtain $e(k)$ that are probabilistically independent.

Our general conclusion is that Least-Square estimation should not be used when our collected via **a real experiment, EIV setting or OE setting**(OE is a subset of EIV problem, meaning that the input signal is perfectly known and the uncertainty affects output measurements dirrectly); that is, when we collect data from the experiment, even if the input is perfectly known, the output would be corrupted by additive or multiplicative noise by the data; using these data samples, LS method does not enjoy consistency property. However, Assumption 1 is satisfied and perfectly make sense when the LTI system to be identified is such that $D(z) = 1$.

$$D(z) = 1 \Rightarrow \begin{cases} \boxed{\text{Identification of } \textbf{FIR (finite impulse response)}} \\ \boxed{\text{Identification of } \textbf{static systems}} \end{cases}$$

The reason is that when $D(z) = 1$, the equation error $e(k)$ is actually playing the role of the output measurement error $\eta(k)$, which means i.i.d by definition, thereby satisfying also assumption 2. Now, some identification problems where both assumptions are satisfied.

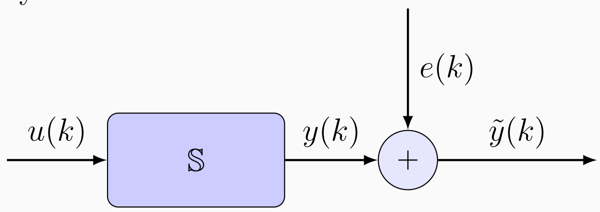
Examples that satisfies assuptions of the consistency property

1) A Finite-Pulse Response system is defined in the following fashion:
$\mathbb{S} : y(k) = \theta_1 u(k) + \theta_2 u(k-1) + \cdots + \theta_n u(k-n+1)$

Measured samples of the output have the following form:
$\tilde{y}(k) = y(k) + \eta(k)$
$\Rightarrow$
$\tilde{y} = \theta_1 u(k) + \theta_2 u(k-1) + \cdots + \theta_n u(k-n+1) + e(k)$
Where $e(.) = \eta(.)$.

2) Static Systems:
$\mathbb{S} : y(k) = f(u(k), \theta) = \theta_1 f_1(u(k)) + \theta_2 f_2(u(k)) + \cdots + \theta_n f_n(u(k))$
Measured samples of the output have the following form:
$\tilde{y}(k) = y(k) + \eta(k)$
$\Rightarrow$
$\tilde{y} = \theta_1 f_1(u(k)) + \theta_2 f_2(u(k)) + \cdots + \theta_n f_n(u(k)) + e(k)$
such basis functions as polinomials or sinusoidal functions.
Where again, $e(.) = \eta(.)$.

Examples of functions f_n can be any set of In this case, the error $e(k)$ enters the equation in an *Error-in-Equation* fashion, which is required to satisfy the second assumption of the consistency property of LS.



In these cases, both assumptions are satisfied, since, by assumption η is considered to be i.i.d, and since the error directly enters the output the first assumption is satisfied, which means:
$\lim_{H\to\infty} \hat{\theta}_{LS} = \theta$

what is critical is the combination of the two assumptions, and no matter what is the structure of the system, we can always manage to define $e(k)$ in a way that it satisfies the first assumption. The problem is that we also have to impose an additional condition, being i.i.d, on $e(k)$ so that also the second assumption is satisfied, while based on the system equation and the way we defined $e(k)$, this condition might not be satisfied.

Our next step is to reformulate this problem in a different fashion, by modifying assumption 2. In another words, it is aimed to replace assumption 2 by a "weaker assumption." This change of perspective leads to what we call *Set-membership* approach to system identification.

What about the unreachable and unobservable modes of the system?

This is a somehow phylosophical question, which can be answered in two way:

1. From the practical point of view, at most engineering cases, the system that is designed does not include unobservable modes that are unstable.
2. From a phylosophical point of view, these modes cannot be identified through input-output data collection.

Chapter 3

Set-membership identification

Set-membership approach is an alternative to the classical, statistical identification approach. Least-Square method is one possible example of statistical estimation algorithms in the context of estimation.

3.1 Set-membership identification of single-input-single-output LTI systems

Main ingredients

We consider a discrete-time system described in the following parametrized regression form:

$$y(k) = f(y(k-1), y(k-2), \cdots, y(k-n), u(k), u(k-1), \cdots, u(k-m), \theta_1, \theta_2, \cdots, \theta_{n+m+1}) \quad (3.1)$$

where $m \leq n$
A-priori information on the system:

- n and m are known
- $f \in \mathbb{F}$ where $\mathbb{F}$ is the class of model selected on the basis of our physical insight.

A-priori information on the noise, the main difference with respect to the previous discussion.

- the noise structure is known, i.e. the way the way uncertainty affects the input-output data)
- the noise is assumed to belong to konwn bounded set $\mathbb{B}$.

When the consistency property of LS was discussed, and in general statistical approach to system identification, the typicall assumption on the noise is that statistical distribution of the noise sequence, or the value of some moments of inertia of the noise is known, such as variance.

Here, the assumption is that the noise sequence η belongs to a bounded $\mathbb{B}$. Remember that assumption two of the consistency property was the crucial one, so a change of perspective is adopted on the second assumption, where we deal with the following problem:

Set-membership identification of LTI system under the assumption that the uncertainty affecting the data can be modelled as an equation error $e(k)$, which is exactly assumption 1 considered for the consistency theorem.

$$\begin{aligned}\tilde{y}(k) = &- \theta_1 \tilde{y}(k-1) - \theta_2 \tilde{y}(k-2) - \cdots - \tilde{y}(k-n) \\ &+ \theta_{n+1} \tilde{u}(k) + \theta_{n+2} \tilde{u}(k-1) + \cdots + \theta_{n+m+1} \tilde{u}(k-m) + \mathbf{e}(\mathbf{k})\end{aligned}$$

where,

$$e(k) \in \mathbb{B}e = \left\{\bar{e} = [e(1), e(2), \cdots, e(H)]^T\right\} : |e(k)| \leq \Delta e, \forall k$$

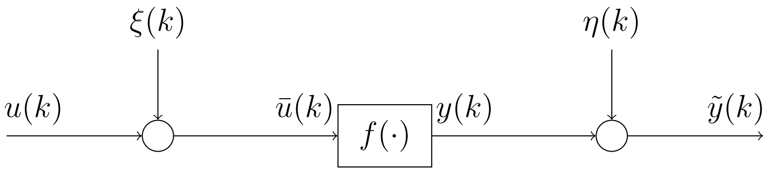
where Δe is a given real bounded constant.

In set-membership, we always have unknown and bounded assumption about the magnitude of noise.

The bound is considered a conservative bound.

Error-in-Variable setting

Let's assume that the data are actually collected from a real experiment, which is EIV setup. The block-diagram representation of this a-priori information is as follows.



Errors-in-variables (EIV) problems refer to the most general case where both the input and the output collected samples are affected by noise.

$$\xi = [\xi(1), \ldots, \xi(H)] \in \mathcal{B}_\xi$$
$$\eta = [\eta(1), \ldots, \eta(H)] \in \mathcal{B}_\eta$$

$\mathcal{B}_\xi$ and $\mathcal{B}_\eta$ are bounded sets described by ***polynomial constraints.***

Most common case:

$$\mathcal{B}_\eta = \{\eta : |\eta(k)| \leq \Delta_\eta\}, \quad \mathcal{B}_\xi = \{\xi : |\xi(k)| \leq \Delta_\xi\}$$

Here, $f(.)$ belong to a class of function $\mathbb{F}$, which is the a-priori assumption on the system.

Considering this a-priori information about the structure of the course, and a-priori information about the system, which is LTI second-order system. The noises can be reformulated as **Error-in-Equation**, which is the same as the first assumption of the consistency property of Least Square.

$$\tilde{y}(k) = -\theta_1\tilde{y}(k-1) - \theta_2\tilde{y}(k-2) + \theta_3\tilde{u}(k) + \theta_4\tilde{u}(k-1) + \theta_5\tilde{u}(k-2)$$
$$\boxed{+\theta_1\eta(k-1) + \theta_2\eta(k-2) - \theta_3\xi(k) - \theta_4\xi(k-1) - \theta_5\xi(k-2) + \eta(k)} =: +e(k)$$

Up till now, nothing is changes, we now that statistically this kind of error is not i.i.d. The difference in is regarding the second assumption about the noise, $e(k)$, which is the boundedness of the noise:

$$|e(k)| \leq \Delta e \forall k = 1, 2, ..., N$$

Provided that we find a way to compute the bound Δe, on $|e(k)|$, the identification problem can be correctly formulated in terms of **Equation-Error** structure in Set-membership framework. **However, Computing Δe is not an easy task**. Since it cannot be done without any assumption about θ_i, since they are to be identified.

Set-membership formulation of the problem identifying a second order LTI system assuming an Equation-Error structure for the uncertainty

Main ingredients:

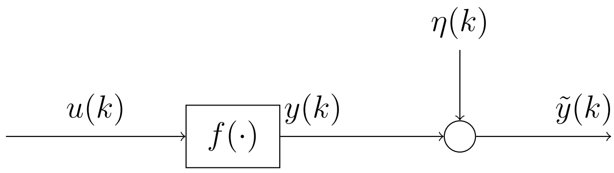
1. **a-priori information** about the system, second order LTI, and the noise, Equation-Error structure and boundedness.

2. **a-posteriori information** are experimentally collected input-output data samples.

Now, here, we have the concept of *Feasability Parameter Set.*

Feasability parameter set

The feasability parameter set $\mathbb{D}_\theta$ is the set of all the values of the parameter vector $\theta = [\theta_1\theta_2\cdots\theta_p]^T \in \mathbb{R}^p$, which are consistent with all the avialable information on the system and the noise, and all he collected data.
To better understand the meaning of FPS, $\mathbb{D}_\theta$, let's assume that we are collecting data according to the following **Output-Error** setup:



Since, we have the following form of noise:

$$y(k) = \tilde{y} - \eta(k)$$

and we know that η is bounded. we can obtain the following relationship about the $y(k)$, being that the real $y(.)$ signal is enveloped in the following fashion:

$$\tilde{y}(k) - \eta k \leq y(k) \leq \tilde{y}(k) + \eta k$$

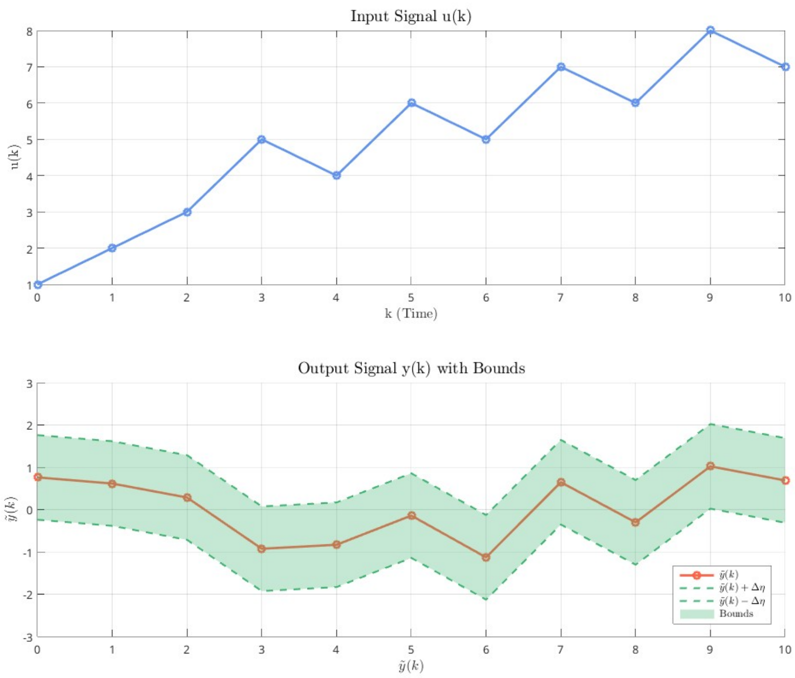


Figure 3.1: The bounds for y encompass all the elements of the θ vector, leading to a relation for y such that its graph is bounded between the represented bounds.

Professor's Quotes

In the statistical identification, assuming probabilistic characteristics for the noise, instead of having bounded interval around each single sample of the output that we collect, we have a normal distribution around each single point. Therefore, a correct formulation of the identification, again, in that context, lead to a set of models, but this time instead of having a bounded set of possible models solving the problem, we obtain a set that is statistically described, where each models have a certain probability of being the correct model.

Now, let's try to define mathematically, the FPS $\mathbb{D}_\theta$

$$\begin{aligned}\mathbb{D}_\theta = \{\theta = [\theta_1\, \theta_2\, \cdots\, \theta_p]^T \in \mathbb{R}^p \,|\, \tilde{y}(k) = &- \theta_1\tilde{y}(k-1) - \theta_2\tilde{y}(k-2) + \theta_3\tilde{u}(k) + \theta_4\tilde{u}(k-1) \\ &+ \theta_5\tilde{u}(k-2) + e(k) \,|\, \forall k > 3\,,\, |e(k)| \leq \Delta e\}\end{aligned}$$

However, this mathematical description is not correct, since here we define FPS as a subset of **parameter space**, $\mathbb{R}^p$, but the description of the unknown error is not in the parameter space. How can we reformulate? Although we don't know e, we know that this unknown error is bounded by Δe, so how can we exploit this information?

$$\mathbb{D}_\theta = \{\theta \in \mathbb{R}^p \,|\, e(k) = \tilde{y}(k) + \theta_1 \tilde{y}(k-1) + \theta_2 \tilde{y}(k-2) - \theta_3 \tilde{u}(k) - \theta_4 \tilde{u}(k-1) - \theta_5 \tilde{u}(k-2), \; \forall k > 3\,,\, |e(k)| \leq \Delta e\}$$

$$\Rightarrow \; \mathbb{D}_\theta = \{\theta \in \mathbb{R}^p \,:\, |\tilde{y}(k) + \theta_1 \tilde{y}(k-1) + \theta_2 \tilde{y}(k-2) - \theta_3 \tilde{u}(k) - \theta_4 \tilde{u}(k-1) - \theta_5 \tilde{u}(k-2)| \leq \Delta e, \; \forall k > 3\}$$

Now, we have obtained **an implicit description** of the set of **all the feasible solutions to our identification problem**, in terms of a set of inquality constraints only involving θ. $\mathbb{D}_\theta$ is now clearly a subset of the parameter space $\mathbb{R}^p$.

Graphical representation of $\mathbb{D}_\theta$ in a two-dimensional space case is something of this kind:

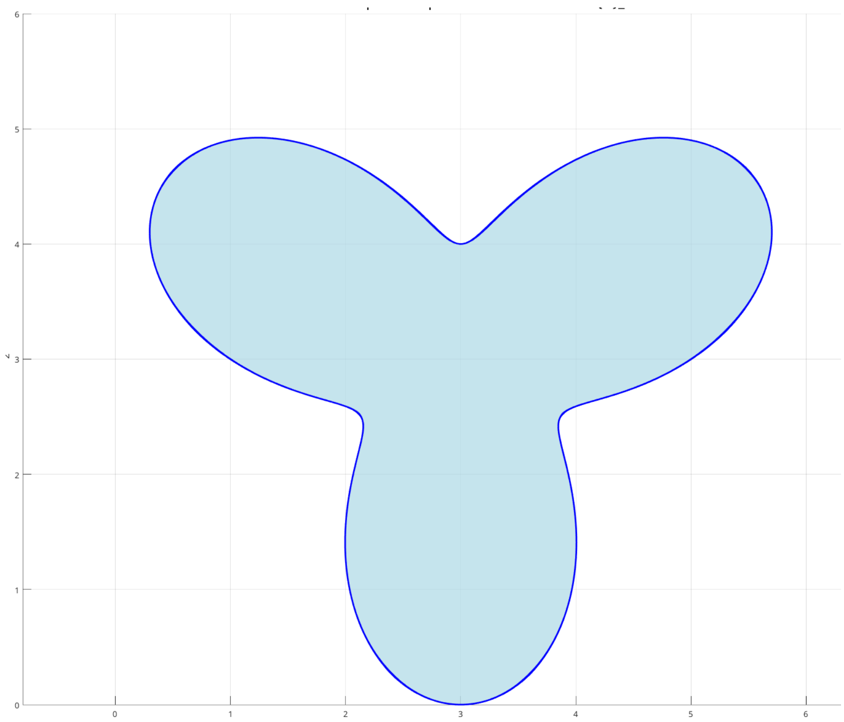


Figure 3.2: a generic two-dimensional FPS

Main features of $\mathbb{D}_\theta$ to be discussed:

I. **Boundedness of $\mathbb{D}_\theta$**

II. **Usefulness of $\mathbb{D}_\theta$**: What is the relationship between $\mathbb{D}_\theta$ and θ, the real value of the parameters.

The answer to II:

Assuming that the a-priori assumption about the system and about the noise are correct. then, θ, the true parameter vector, belong to the set $\mathbb{D}_\theta$.

The answer to I:

The boundedness of $\mathbb{D}_\theta$ depends on the way the data is collected. For the moment, let's assume that $\mathbb{D}_\theta$ is bounded.

Now that we have obtained an implicit description of $\mathbb{D}_\theta$, how can a useful model from $\mathbb{D}_\theta$ can be extracted, e.g. to simulate the behavior of the system under the study or to design a controller for such a system?

We consider two class of SM estimation algorithms, or estimators:

1. **Set-valued estimator:** defined as **an** estimation algorithm that **provides a** possibly conservative **description** of $\mathbb{D}_\theta$ in **a simplified geometric form** that can be easily used to simulate or control the system.

2. **Pointwise estimators:** defined as estimation algorithms that provide a single value of $\hat{\theta}$, which is an optimal estimate of θ **in some sense to be defined.**

Among all possible estimators in the class *1*, we consider the algorithm that provides the minimum volume box outer bounding $\mathbb{D}_\theta$.

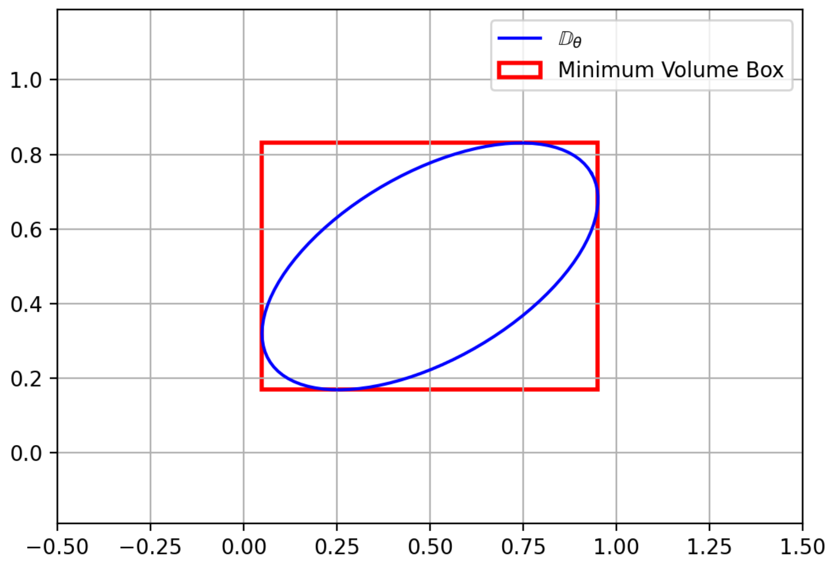


Figure 3.3: Parameter Uncertainty Intervals, considering the first case mentioned above.

This estimator is implicitly providing what we will call *Parameter Uncertainty Intervals*, **PUI**s defined as bellow:

$$PUI_{\theta_J} = \left[\, \underline{\theta_J}, \overline{\theta_J} \,\right]$$

where:

$$\underline{\theta_J} := \min_{\mathbb{D}_\theta} \theta_J$$

and

$$\overline{\theta_J} := \max_{\mathbb{D}_\theta} \theta_J = \min_{\mathbb{D}_\theta} (-\theta_J)$$

Since we know for sure, the true value of the parameter is inside the outerbox, we know that each PUI includes the true value of a parameter.

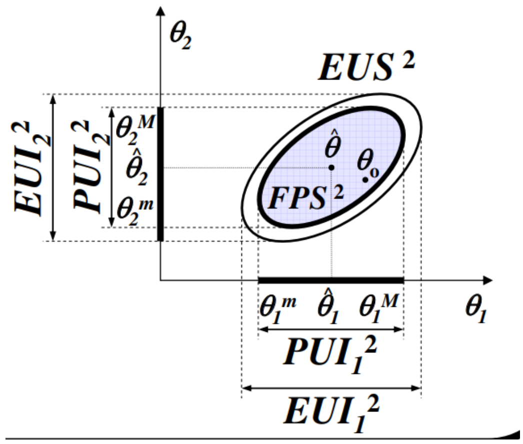


Figure 3.4: Parameter Uncertainty Intervals, considering the first case mentioned above.

Designing a controller for a system described in this manner

For instance, consider the following system:

$$G(s) = \frac{\theta_2}{s+\theta_1} \text{ such that: } \begin{cases} \theta_1 \in [\underline{\theta_1}, \overline{\theta_1}] \\ \theta_2 \in [\underline{\theta_2}, \overline{\theta_2}] \end{cases}$$

The Bode plot of such a transfer function is a range which can be seen in figure 3.5.

Assuming that you want to design a lead/lag controller in the frequency-domain. We need to consider the Nichol plot which is farthers from -180 and the design procedure is done. In this way after the design, rest assured, all the possible transfer functions would satisfy the requirements.

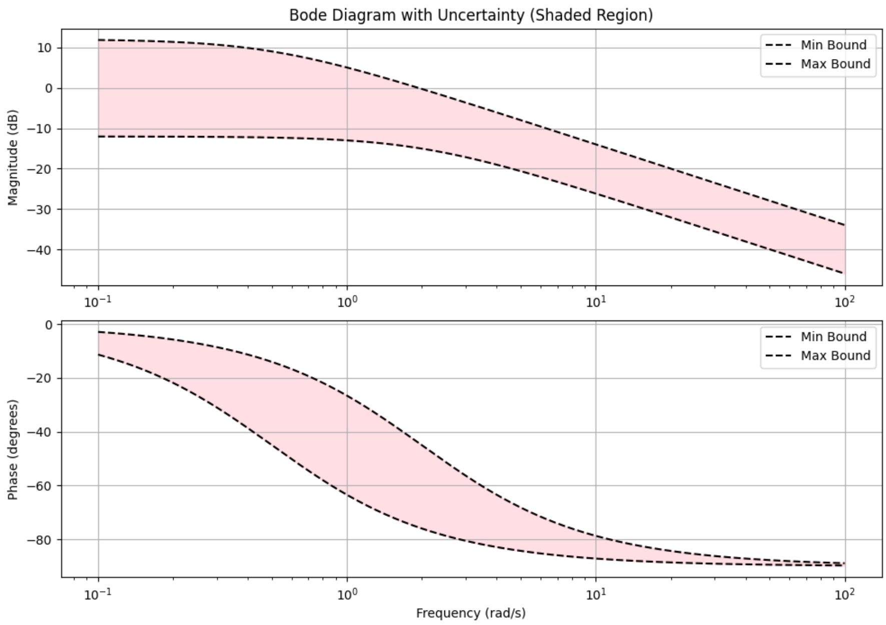


Figure 3.5: The envelop of all a transfer functions described in such a manner with uncertainties on the parameters.

What are the geometrical features of $\mathbb{D}_theta$ for the specific case when the system is LTI and the uncertanty affects the data can be described by means of a bounded equation error?

For example, consider the following assumption regarding a-priori information about the system dynamics.

$$G(z) = \frac{\theta_2 z}{z + \theta_1}$$
$$\Rightarrow G(q^{-1}) = \frac{\theta_2}{1 + \theta_2 q^{-1}}$$
$$\Rightarrow y(k) = \frac{\theta_2}{1 + \theta_1 q^{-1}} u(k)$$
$$\Rightarrow y(k) + \theta_1 y(k-1) = \theta_2 u(k)$$
$$\Rightarrow y(k) = -\theta_1 y(k-1) + \theta_2 u(k)$$

A-priori information about the noise is:

- Equation Error, $e(k)$, affects the system.
- $e(k)$ is bounded by a norm, $|e(k)| < \Delta e$.

A-posteriori information are the collected samples of $\tilde{u}(k)$, $\tilde{y}(k)$.

Based on the informations at hand:

$$\mathbb{D}_\theta = \left\{\theta \in \mathbb{R}^2 : \tilde{y}(k) = -\theta_1 \tilde{y}(k-1) + \theta_2 \tilde{u}(k) + e(k),\ |e(k)| < \Delta e \ \forall k = 1, 2, \cdots, N\right\}$$
$$\Rightarrow \mathbb{D}_\theta = \left\{\theta \in \mathbb{R}^2 : |\tilde{y}(k) + \theta_1 \tilde{y}(k-1) - \theta_2 \tilde{u}(k)| < \Delta e \ \forall k = 2, \cdots, N\right\}$$
$$\Rightarrow \mathbb{D}_\theta = \left\{\theta \in \mathbb{R}^2 : -\Delta e < \tilde{y}(k) + \theta_1 \tilde{y}(k-1) - \theta_2 \tilde{u}(k),\ \ \tilde{y}(k) + \theta_1 \tilde{y}(k-1) - \theta_2 \tilde{u}(k) < \Delta e \ \forall k > 2\right\}$$

Therefore, for $k = 2$ two constraints are obtained on FPS, leading to the regione between two parallel line.

$$-\Delta e < \tilde{y}(k) + \theta_1 \tilde{y}(k-1) - \theta_2 \tilde{u}(k) \Rightarrow \theta_2 \geq \frac{\tilde{y}(1)}{\tilde{u}(2)} + \frac{\tilde{y}(2) - \Delta e}{\tilde{u}(2)}$$

Which leads to the following region in the parameter space:
THE PLOT TO BE PLOTTED.

Here, from the other inequality we obtain the second constraint.

$$\tilde{y}(k) + \theta_1 \tilde{y}(k-1) - \theta_2 \tilde{u}(k) < \Delta e \Rightarrow \theta_2 \leq \frac{\tilde{y}(1)}{\tilde{u}(2)} + \frac{\tilde{y}(2) + \Delta e}{\tilde{u}(2)}$$

THE PLOT TO BE PLOTTED.

Exploiting *a-priori* information and two input-output samples, one can obtain an unbounded set for $\mathbb{D}_\theta$. It is evident that, even in the noise-free case, three data samples are needed to solve the system of equations to obtain the true values of the parameters: the order of the system plus the number of parameters.

For $k = 3$, we obtain two additional constraints, which lead to two parallel lines in the parameter space. This results in a bounded **Feasible Parameter Set (FPS)** $\mathbb{D}_\theta$. **Therefore, by our weak assumptions and with three data samples, we have obtained a bounded set as the *Feasible Parameter Set.***

If the *a-priori* assumptions are correct, the existence of an FPS is guaranteed. However, if the assumptions are not satisfiedeither because the system is nonlinear and not LTI, its order is different, or the magnitude of the noise is larger than assumedthe FPS will become a **null set**. This, in itself, is a positive feature of this method, providing feedback about our estimation. Nonetheless, the statistical method does not enjoy such a property. That is, regardless of the situation, the Least Squares (LS) algorithm will provide an estimation that may not appropriately describe the system.

In general, we can conclude that under the considered assumption (LTI system with bounded equation error) the FPS, $\mathbb{D}_\theta$, is a **polytope**, and therefore, the PUIs can be numerically computed by solving a set of *linear programming* (LP) problems.

example

$$\mathrm{PUI}_{\theta_1} = \left[\underline{\theta_1},\, \overline{\theta_1}\right]$$

Considering our LTI system:

$$\theta_1 = \min_{\theta \in \mathbb{D}_\theta} \theta_1 = \min_{\theta \in \mathbb{R}^n} \theta_1 \text{ subject to } \left\{ \begin{array}{c} \tilde{y}(k) + \theta_1 \tilde{y}(k-1) - \theta_2 \tilde{u}(k) < \Delta e \\ -\tilde{y}(k) - \theta_1 \tilde{y}(k-1) + \theta_2 \tilde{u}(k) < \Delta e \end{array} \right\}$$

In order to obtain the upper bound of PUI_{θ_1}, it is enough to do minimization for $-\theta_1$.

$$\theta_2 = \min_{\theta_1 \in \mathbb{R}^n} -\theta_1 \text{ subject to } \left\{ \begin{array}{c} \tilde{y}(k) + \theta_1 \tilde{y}(k-1) - \theta_2 \tilde{u}(k) < \Delta e \\ -\tilde{y}(k) - \theta_1 \tilde{y}(k-1) + \theta_2 \tilde{u}(k) < \Delta e \end{array} \right\}$$

A general LP problem is as follows:

$$\min_{x \in \mathbb{R}^n} C^T x \text{ subject to } Ax \preceq b$$

We say that an optimization problem is an LP problem if the problem can be exactly written as the minimization of a linear combination of optimization variable x subject to a set of linear inequalitiies and/or equalities.

LP problems are convex, or at any rate linear, optimization problems that can be solved to the global optimial solution in MATLAB by using the function ***linprog***. Therefore, all the time that we have a system with a dynamic that is linear with respect to the parameters, and that we have a bounded noise, we can reach the solve a linear optimization problem.

Considering **the second problem of the first lab**, which asks for the solution of the optimal l_∞ estimation problem, we consider the same setting used for the first problem, but we look for the following estimation:

$$\hat{\theta}_{l_\infty} = \arg\min_{\theta \in \mathbb{R}^n} \|\bar{y} - A\theta\|_\infty$$

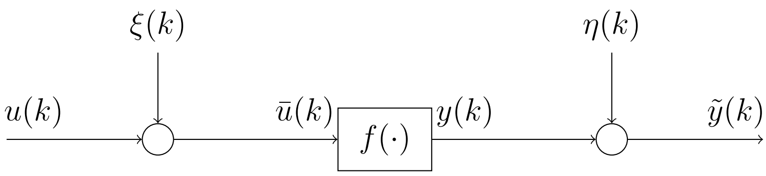
where $\|.\|_\infty$ is the l_∞ norm of a vector.
Hint for solving the problem:

$$\hat{\theta}_{l_\infty} = \arg\max_{\theta \in \mathbb{R}^n}(|\rho(1)|,\ |\rho(2)|,\ \cdots,\ |\rho(N-m)|)$$

Can this problem be written as an LP problem? the answer is yes and we have to do it to solve the second problem.

Set-membership estimation with EIV set-up

Now, it is aimed to consider a situation which is must more realistic with respect to the previous problem that was dealt with.
Previously, we discussed that, even if the noise affects the measurements in the real data acquisition setting in an EIV setting, we can reformulate EIV in an equation error. Nonetheless, we faced a major problem, which is the fact that the error $e(k)$ added to the equation is no more i.i.d.. In set-membership approach being discussed, this is no more a problem as long as it is guaranteed that the norm of this error is going to be bounded; here, there is no assumption on the statistical characteristic of this noise. Now, the problem is that computing a bound for the norm of the error is not an easy task, since after the reformulation of the problem the error norm involve yet-to-be-estimated parameters. Therefore, practically, it is difficult, if not impossible, to compute a bound on the noise, even if the bound of the noise affecting the measurements is known. This is suggesting us that maybe it is better to formulate the problem in another way, not in equation-error, but in EIV setting in a different formulation.



Errors-in-variables (EIV) problems refer to the most general case where both the input and the output collected samples are affected by noise.

$$\xi = [\xi(1), \dots, \xi(H)] \in \mathcal{B}_\xi$$
$$\eta = [\eta(1), \dots, \eta(H)] \in \mathcal{B}_\eta$$

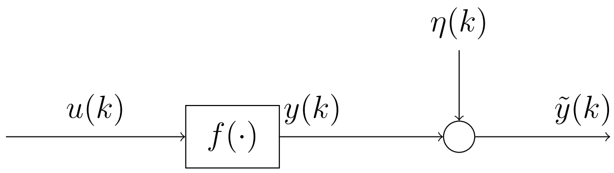
$\mathcal{B}_\xi$ and $\mathcal{B}_\eta$ are bounded sets described by ***polynomial constraints.***

Most common case:

$$\mathcal{B}_\eta = \{\eta : |\eta(k)| \leq \Delta_\eta\}\,, \quad \mathcal{B}_\xi = \{\xi : |\xi(k)| \leq \Delta_\xi\}$$

which is a hyper-dimensional box.

A subset of this problem, as was discussed also in the previous chapter, is Output-Error, OE, set-up, meaning that the input signal is perfectly known.



where,

- $$\eta = [\eta(1),\ \eta(2),\ \cdots,\ \eta(H)] \in \mathbb{B}_\eta$$
- $\mathbb{B}_\eta$ is a bounded set described by ***polynomial constraints***.
- Most common case: $\mathbb{B}_\eta = \{\eta :\ |\eta(k)| \leq \Delta_\eta\}$

The FPS for the general EIV problem is implicitly defined as follows:

$$\begin{gathered}\mathbb{D}_\theta = \{\theta \in \mathbb{R}^n :\ y(k) = f(y(k-1), y(k-2), \cdots, y(k-n), u(k), u(k-1), \cdots \\ , u(k-m), \theta_1, \theta_2, \cdots, \theta_{n+m+1}), k = n+1,\ n+2,\ \cdots,\ N,\ y(k) = \tilde{y}(k) - \eta(k), \\ u(k) = \tilde{u}(k) - \xi(k),\ k = 1,\ 2,\ \cdots,\ N,\ |\xi(k)| \leq \Delta\xi(k),\ |\eta(k) \leq \Delta\eta(k),\ k = 1,\ 2,\ \cdots,\ N\}\end{gathered}$$

The FPS for the case of **LTI descrete-time systems with EIV noise structure** is given by:

$$\mathbb{D}_\theta = \left\{\theta \in \mathbb{R}^p :\ (\tilde{y}(k) - \eta(k)) + \sum_{i=1}^{n} a_i(\tilde{y}(k-i) - \eta(k-i)) = \sum_{j=0}^{m} b_j(\tilde{u}(k-j) - \xi(k-j)),\right.$$
$$\left. t = n+1, \dots, H, \quad |\xi(k)| \leq \Delta\xi(k), \quad |\eta(k)| \leq \Delta\eta(k), \quad k = 1, \dots, H\right\}$$

This includes $2N$ new variables involved in the FPS which are not variables in parameter space. Now, these variables should be eliminated, and this is not possible without setting some conservative conditions. **If you come up with an idea to do so without considering conservative conditions, you can publish a paper**.

Since the FPS defined in the above equation depends on some additional unknownes(all the samples of the noise sequences). The constraints defining $\mathbb{D}_\theta$ cannot be rearranged in such a way to eliminate dependancies on such additional variables. To solve the problem, we need to extend the space of decision variables in by defining the *Extended Feasible Parameter Set*(EFPS).

$$\mathcal{D}_{\theta,\eta,\xi} = \{\theta \in \mathbb{R}^p, \eta \in \mathbb{R}^H, \xi \in \mathbb{R}^H : (y(k) - \eta(k) + \sum_{i=1}^{n} a_i(y(k-i) - \eta(k-i)) =$$
$$\sum_{j=0}^{m} b_j(u(k-j) - \xi(k-j)),\ t = n+1, \ldots, H,$$
$$|\xi(k)| \leq \Delta\xi(k),\ |\eta(k)| \leq \Delta\eta(k),\ k = 1, \ldots, H\ \}$$

As you can see in the box below, here, we are expanding the space we are considering for FPS, hence EFPS.

$$\boxed{\mathcal{D}_{\theta,\eta,\xi} = \{\theta \in \mathbb{R}^p, \eta \in \mathbb{R}^H, \xi \in \mathbb{R}^H\}} \tag{3.2}$$

A simplified graphical version is shown in the following graph. **Now, the FPS is the projection of the EFPS on the parameter space**.

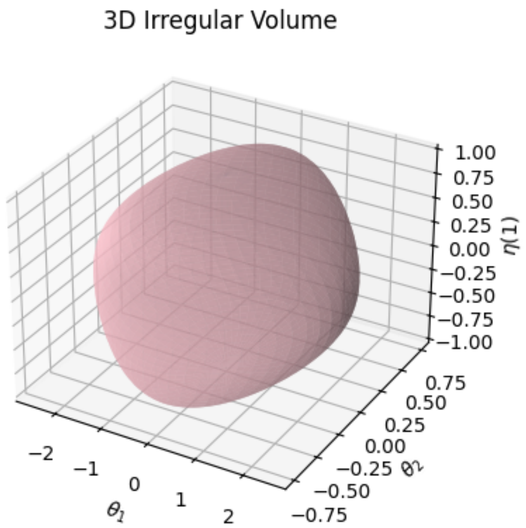


Figure 3.6: A simplified, general graph of EFPS.

In this case,

$$\underline{\theta}_k = \min_{\theta \in \mathbb{D}_\theta} \theta_k = \min_{\theta,\eta,\xi \in \mathbb{D}_{\theta,\eta,\xi}} \theta_k$$

That is,

$$\begin{aligned} \theta_k \quad = \quad & \min_{\theta \in \mathbb{D}_{\theta,\eta,\xi}} \theta_k \text{ subject to } \tilde{y}(k) - \eta(k) + \sum_{i=1}^{n} \theta_i(\tilde{y}(k-i) - \eta(k-i)) \\ & + \theta_{n+1}(\tilde{u}(k) - \xi(k)) + \sum_{j=1}^{m} \theta_{n+j+1}(\tilde{y}(k-j) - \xi(k-j)), \\ & \forall k > n+1,\ |\xi(k)| \leq \Delta\xi,\ |\eta(k)| \leq \Delta\eta,\ \forall k \end{aligned}$$

The drawback is that EFPS is a non-convex set defined by **polynomial** (*bilinear*) constraints.

Professor's Quote:

To the best of my knowledge, in the domain of non-convex functions, the only class that can be optimized up to the global minimum is the class of non-convex functions with polynomial constraints.

Boundedness of the noise

Practically, the assumption that the noise can take any value is unreasonable. Since sensors are used for measurements and are designed to be stable and precise, it is not realistic for the noise to take on any value. For this reason, it makes sense to assume bounded noise.

What signal to use as input signal?

If the signal applied to a system is simply a **ramp** or **step** signal, after the transient period, the output will reflect the generalized DC gain of the system when it reaches steady-state. Therefore, a staircase of steps with varying amplitudes combined with a random signal may be a good choice to stimulate different modes of the system.

Refer to Lab 01, Problem 2, for least squares with l_∞-norm.

Consider the following FPS. Due to its non-convexity, the optimizer may return a local minimum, which is not desirable.

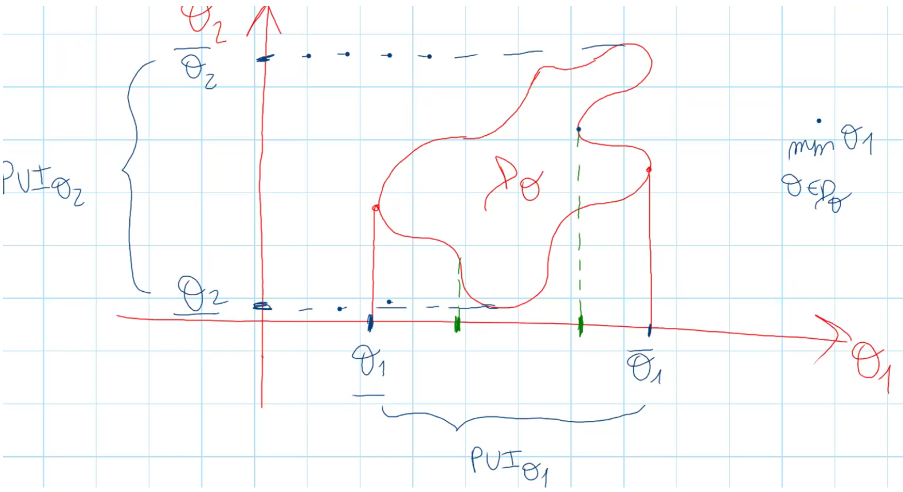


Figure 3.7: A simplified, general graph of EFPS.

Convex relaxation for polynomial optimization problem

General ideas and intuitions are as follows:
let's consider the following general polynomial optimizatin problem:

$$\min_x f_0(x) \text{ s.t. } \begin{cases} f_k(x) \leq 0 \, (k = 1, 2, \cdots, l) \\ f_k(x) = 0 \, (k = l+1, l+2, 1, \cdots, m) \end{cases}$$

where, f_0 and f_k are multivariable polynomials in the optimization variables$x$.

Example 1

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

The following is what we call *Polynomial Optimization Problem* **POP**
$\min(x_1^2 + x_2 x_3^3 + x_1^5 x_2^3 + 7x_3^2 + \cdots)$

for the second order case: $f_0(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_3 + \theta_4 x_1^2 + \theta_5 x_2^2$

Example 2

a constraint POP $\min(x_1^2 + 11x_2)$ s. t. $\begin{cases} x_1^2 - 4 = 0 \\ x_1^3 + 7x_2 - 13x_1 x_2 \leq 0 \end{cases}$

All POPs can be written in the so-called *epographic* form by introducing a new "slack" variable γ.

$$\begin{array}{ll} \min\limits_{x} & f_0(x) \\ \text{s.t.} & \\ & f_1(x) \geq 0 \\ & f_2(x) \geq 0 \\ & \vdots \\ & f_5(x) \geq 0 \\ & f_k(x) = 0 \end{array} \iff \begin{array}{ll} \min\limits_{x,\,t} t & \\ \text{s.t.} & \\ & f_0(x) \leq t \text{ or } t - f_0(x) \leq 0 \\ & f_1(x) \geq 0 \\ & f_2(x) \geq 0 \\ & \vdots \\ & f_5(x) \geq 0 \\ & f_k(x) = 0 \end{array}$$

By rewriting a generic POP in the epigraphic form, we recognize that a generic POP can be seen as the problem of minimizing a linear function over a non-convex set described by polynomial constraints. This is also true for unconstraint POP.

$$\min_{x} \; f_0(x) \iff \begin{array}{ll} \min\limits_{x,\,t} & t \\ \text{s.t.} & \\ t - f_0(x) \leq 0 & \end{array}$$

A graphical representation of this problem is as follows:

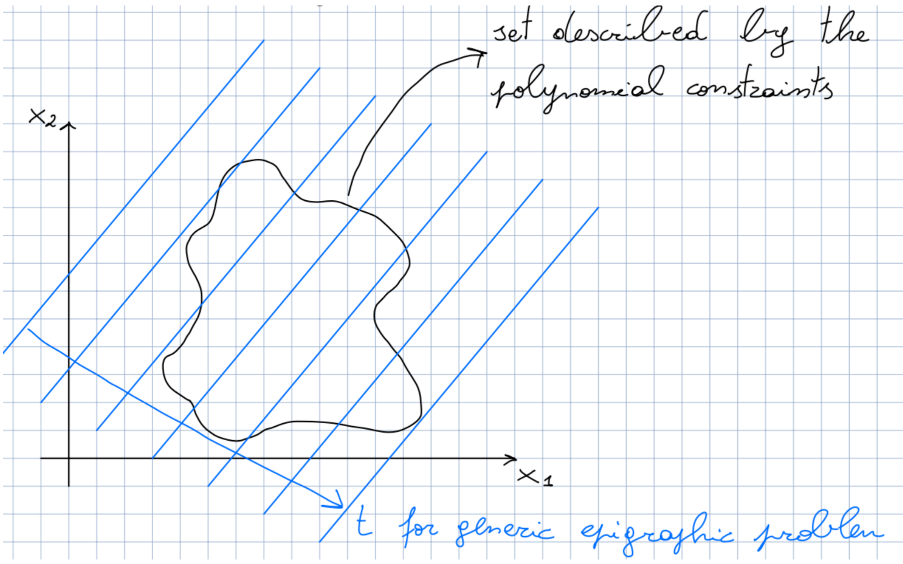


Figure 3.8: the geraphical representation of a generic epigraphic minization, each line represent different values of t.

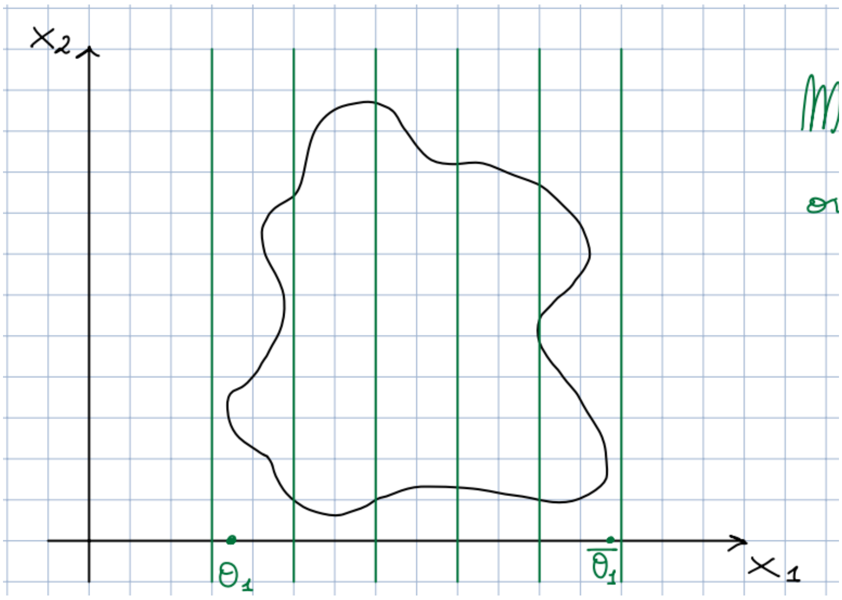


Figure 3.9: Graphical representation of a linear programming over θ_1

Important Remark

Since a generic POP can be rewritten in the epigraphic form that is the problem of minimizing a linear function over a non-convex set, the non-convexity of the problem is now completely embedded in the description of the set of constraint.

If we want to be able to compute the global optimal solution to the non-convex POP, we have to deal with the non-convexity of the set of constraints.

Now, the idea is to replace the non-convex set with a convex set such that the result of the global optima is equal to that of non-convex.

Convex-hull of a non-convex set

The convex-hull of a non-convex set S is the smallest convex set that is including S.

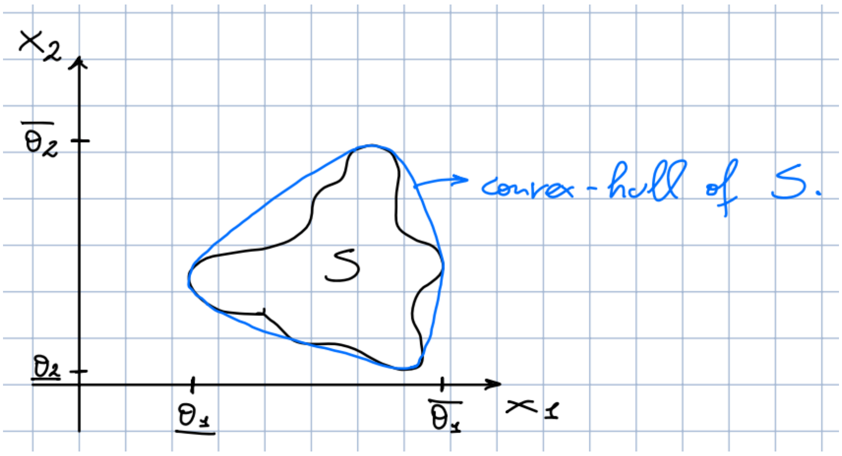


Figure 3.10: convex-hull of a generic non-convex set

The general idea here is that if we are able to write down equation describing convex-hull C_S of the non-convex set S:

$$\min_{x \in C_s} f(x) = \min_{x \in S} f(x)$$

where $f(x)$ is a linear function of x.

However, in general, computing the mathematical description of the convex haul is quite a difficult problem. For the case of POPS - a particular class of non-convex optimization problem - Lassere et al were able to compute **a convex relaxation** of S depending on a parameter which is called ***order of relaxation*** δ.

For a given fixed value of δ we have the following situation. For different order of relaxation, we obtain different convex sets including the original set. They proved by increasing the order of relaxation, we obtain convex sets that is the convex set of the original set. For the problems we are dealing with, the optimal solution is obtained with a fixed value of relaxation, and convergence is typically really fast. The issue is that the value of the δ is not known.

- Computation of parameter bounds (PUIs) require the solution to the global optimum of non-convex polynomial optimization problems.

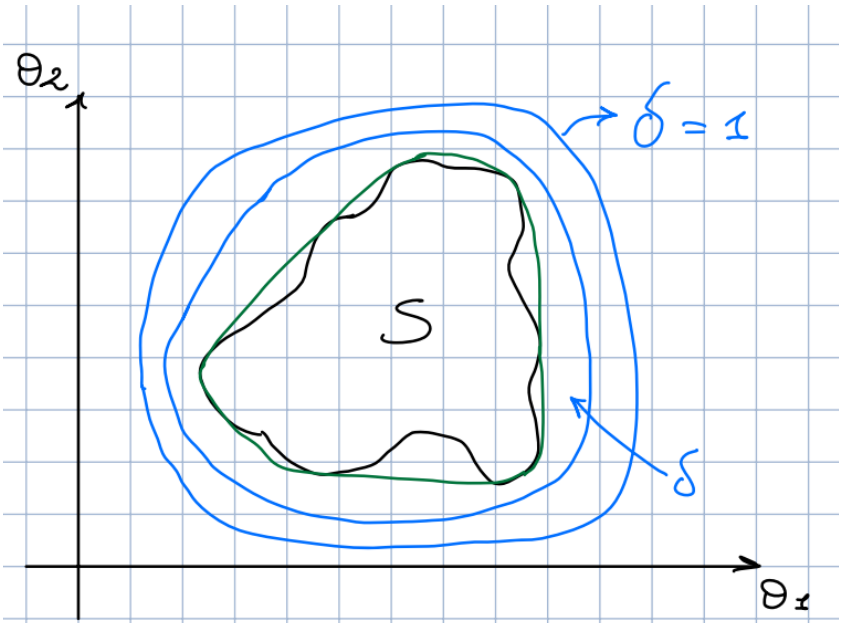


Figure 3.11: convex-hull of a generic non-convex set

- By applying *Moments theory* (Lasserre) or *Sum-of-squares (SOS) theory* (Chesi, Parrilo), it is possible to build a sequence of **convex** semidefinite (SDP) optimization problems whose size/complexity depends on a parameter called *order of relaxation* δ.
- For any given δ, the following inequalities hold

$$\underline{\theta}_k^\delta \leq \underline{\theta}_k, \quad \overline{\theta}_k^\delta \geq \overline{\theta}_k$$

- Furthermore,

$$\lim_{\delta \to \infty} \underline{\theta}_k^\delta = \underline{\theta}_k, \quad \lim_{\delta \to \infty} \overline{\theta}_k^\delta = \overline{\theta}_k$$

In gradient-based algorithms, we have the risk of underestimating the value of the parameters by trapping into local minima, while in this method, rest assured, the true value of the parameters is going to be included in the interval.

In order to solve POPs by means of Lassere convex-relaxation appraoch in MATLAB, we use the following two software, which are MATLAB toolboxes, that can be freely downloaded online:

- SparsePOP: given a POP and a value of the order of relaxation δ, provides **the SDP relaxed problem.**
- Sedumi: is called by SparsePOP to solve the obtained convex SDP.

The lecture after the lab 03

In the laboratory, we saw that SparsePOP computes convex relaxation for a given relaxation. The Sediumi toolbox is used to solve the relaxed optimization problem.

Selection of the order of relaxation

Let's call x^* the global optimal solution of a given POP, and x^* the solution of the corresponding convex relaxation of order δ.

QUESTION: How to selcet δ?
Well, from the theory we have the following results:
R1)

$$\lim_{\delta\to\infty} f(x^\delta) = f(x^*)$$

which implies, the following, to be unprecise:

$$\lim_{\delta\to\infty} = x^*$$

R2): $\delta) \geq \lceil \frac{n_{max}}{2} \rceil$, where n_{max} is the maximum degree of the polynomial describing the functional and the constraints.

R3) the complexity of SDP problem obtained by applying to the original convex relaxation techniques grows exponentially in the order of relaxatoin δ, as well as the number of optimization variables (AKA decision variables) of the original POP.

On the other hand, we have the following result too:

R4): For a large class of problems including most of the problems in this course, it is possible to prove that the convergence to the global optimal solution of the original POP is very fast and it is achieved for a finite value of δ. but still the result number 2 should be satisfied.

Let's analyze the impact of R1, R2, R3, and R4 on the applicationo of convex relaxation techniques to a set-membership identification problem of an LTI system with EIV assuption about the noise.

For SM Id. of LTI system with EIV of a polynomial of order 2, we can start with $\delta = 1$. Because we have bi-linear constraints, i.e. polynomial constraints of order 2. Number of optimization variable is given by:

$$P + 2N$$

Where P is the number of parameters to be identified and N is the number of pair of data samples. Since in system identification problem N is going to be quite large in many applications in order to obtain accurate PUI's, the complexity is going to be untractable. Nonetheless, the following result can be exploited:
R5) if the structure of the original POP satisfies a property called *running intersection property*,

it's possible to build a sequency of convex SDP relaxation, involving much less variables. (Sparse Convex - relaxation)

about SparsePOP

SparsePOP toolbox is able to automatically check whether the original POP satisfies the running intersection property, and if this is the case, it applies the *sparse convex relaxation.*

R6) The complexity of SDP problem obtained by applying the sparse convex relaxation to the original POP:

- **grows exponentially iwth the order of relaxation**.
- **grows linearly with the number of optimization variables**.

R6 tell us that the SM - ID problem for LTI system with EIV assumption is computationally tractable for a rather large value of input-output pair of experimentally collected data (N), since the problem satisfies the **running intersection property** and it is characterized by **bilinear constraints**, which allow us to start with relaxation $\delta = 1$.

considering that the problem is computationally tracktable having the constraints are bilinear. can we find a trick to solve a polynomial with large number of parameters, which may be computationally untracktable?

How to formulate the SM - ID problem for an LTI system of order n with EIV noise structrue, in terms of SparsePOP

:

Exam matter

The following procedure is what should be done to solve the exam problem.

1. First, let's write down the FPS and the EFPS. To simplify, here, we consider $n = 2$.
   FPS:

$$\mathbb{D}_\theta = \{\theta \in \mathbb{R}^5 : y(k) + \theta_1 y(k-1) + \theta_2 y(k-2) - \theta_3 u(k) - \theta_4 u(k-1) - \theta_5 u(k-2) = 0 \\ \forall k = 3,\, 4,\, \cdots,\, N, |\eta(k) \leq \Delta\eta, |\xi(k)| \leq \Delta\xi, \forall k \in \mathbb{N}\}$$

   EFPS:

$$\mathbb{D}_{\theta,\eta,\xi} = \{\theta \in \mathbb{R}^5,\, \eta \in \mathbb{R}^N,\, \mathbb{R}^N : \tilde{y}(k) - \eta(k) + \theta_1\tilde{y}(k-1) - \theta_1\eta(k-1) + \theta_2\tilde{y}(k-2) \\ -\theta_2\eta(k-2) - \theta_3\tilde{u}(k) + \theta_3\xi(k) - \theta_4\tilde{u}(k-1) + \theta_4\xi(k-1) - \theta_5\tilde{u}(k-2) + \theta_5\xi(k-2) = 0 \\ \forall k = 3,\, 4,\, \cdots,\, N, |\eta(k)| \leq \Delta\eta, |\xi(k)| \leq \Delta\xi, \forall k \in \mathbb{N}\}$$

   PUI:

$$PUI_k = \left[\underline{\theta}_k,\, \overline{\theta}_k\right]$$

where

$$\underline{\theta}_k = \min_{\theta \in \mathbb{D}_\theta} \theta_k = \min_{\theta,\eta,\xi \in \mathbb{D}_{\theta,\eta,\xi}} \theta_k \quad \text{s.t.}$$
$$\tilde{y}(k) - \eta(k) + \theta_1 \tilde{y}(k-1) - \theta_1 \eta(k-1) + \theta_2 \tilde{y}(k-2) - \theta_2 \eta(k-2)$$
$$-\theta_3 \tilde{u}(k) + \theta_3 \xi(k) - \theta_4 \tilde{u}(k-1) + \theta_4 \xi(k-1) - \theta_5 \tilde{u}(k-2) + \theta_5 \xi(k-2) = 0 \quad \forall k = 3, 4, \cdots, N,$$
$$\eta(k) + \Delta\eta \geq 0, \quad \Delta\eta - \eta(k) \geq 0, \quad \xi(k) + \Delta\xi \geq 0, \quad \Delta\xi - \xi(k) \geq 0 \quad \forall k \in \mathbb{N}$$

$$\overline{\theta}_k = \max_{\theta \in \mathbb{D}_\theta} \theta_k = \max_{\theta,\eta,\xi \in \mathbb{D}_{\theta,\eta,\xi}} \theta_k \quad \text{s.t.}$$
$$\tilde{y}(k) - \eta(k) + \theta_1 \tilde{y}(k-1) - \theta_1 \eta(k-1) + \theta_2 \tilde{y}(k-2) - \theta_2 \eta(k-2)$$
$$-\theta_3 \tilde{u}(k) + \theta_3 \xi(k) - \theta_4 \tilde{u}(k-1) + \theta_4 \xi(k-1) - \theta_5 \tilde{u}(k-2) + \theta_5 \xi(k-2) = 0 \quad \forall k = 3, 4, \cdots, N,$$
$$\eta(k) + \Delta\eta \geq 0, \quad \Delta\eta - \eta(k) \geq 0, \quad \xi(k) + \Delta\xi \geq 0, \quad \Delta\xi - \xi(k) \geq 0 \quad \forall k \in \mathbb{N}$$

2. Formulating the obtained POP in terms of SparsePOP data structure.
   Let's consider how to describe in SparsePOP terms the bilinear eqaulity constraints. Considering k = 3.

   $$\tilde{y}(3) - \eta(3) + \theta_1 \tilde{y}(2) - \eta(2) + \theta_2 \tilde{y}(1) - \eta(1) - \theta_3 \tilde{u}(3) + \xi(3) - \theta_4 \tilde{u}(2) + \xi(2) - \theta_5 \tilde{u}(1) + \xi(1) = 0$$

   Now, we have to shape our data structure for this inequality.

```
ineqPolySys{c}.typeCone = -1; equality constraint
ineqPolySys{c}.dimVar = 5 + 2*N;
% P: the number of variables involved, P = 5
% N: the number of pairs of input-output data samples
ineqPolySys{c}.degree = 2; %always 2,for bilinear equations
ineqPolySys{c}.noTerms = 12;
ineqPolySys{c}.supports = sup_matrix;
ineqPolySys{c}.coef = [y_tilde(3);-1;y_tilde(2);-1; ...]';
```

   Support matrix show have as many row as the terms in the constraints and as many constraints as the number of optimization variables $P + 2 * N$.

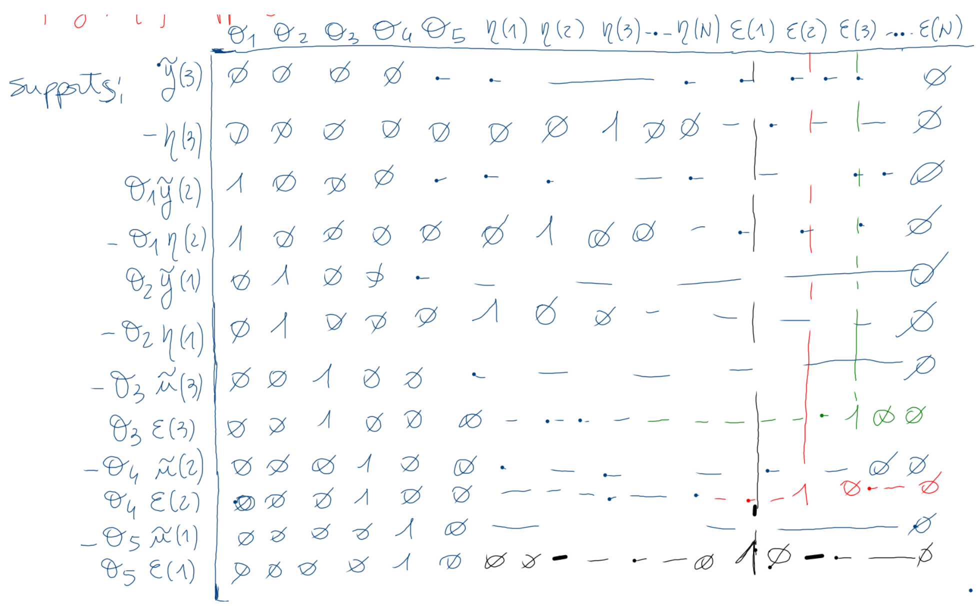


Figure 3.12: Support matrix for the equality constraints

Regarding the constraints on the boundedness of the noise, they can be imposed consider a lower bound and upper bound for the variables corresponding to the noise. Look at the solution of lab 03.

3.2 Set-membership identification of multi-input-multi-output LTI system

Now that we have considered SISO LTI system, a new direction can be considering MIMO systems. Again, to formulate our problem, we should consider our a-priori and a-posteriori informations.

A-priori information on the system

The system is a MIMO LTI system with p input and q outputs, described by means of a **matrix transfer function** $G(q{-}1)$, where the elements of it are SISO transfer function.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_q \end{bmatrix} = \begin{bmatrix} G_{11}(q^{-1}) & G_{12}(q^{-1}) & \dots & G_{1p}(q^{-1}) \\ G_{21}(q^{-1}) & G_{22}(q^{-1}) & \dots & G_{2p}(q^{-1}) \\ \vdots & \vdots & \ddots & \vdots \\ G_{q1}(q^{-1}) & G_{q2}(q^{-1}) & \dots & G_{qp}(q^{-1}) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \end{bmatrix}$$

Discussion about the structure this matrix transfer function

In general, The denuminator of these transfer functions can be different, but considering that the transfer function can be obtained from steady-state representation of the system, $C * (sI - A)^{-1}B + D$, before the possible cancellations, The denuminator of all of these transfer functions would be the same. Having this consideration, less parameters should be identified. On the otherhand, in this case, the possible cancellations are hard to be recognized, due to the range obtained for each zero, PUI. In another possible case, some a-priori information regarding the degree of the transfer functions might be available. In that, we know that the order of a given output with respect to a given input is going to be of a specific value.

The generic output y_i is given by:

$$y_i = G_{i1}(q^{-1}) + G_{i1}(q^{-1}) + \cdots + G_{i1}(q^{-1}) \;\; \forall \;\; i = 1,\, 2,\, \cdots,\, q$$

where each output y_i only depends on the past samples of the output itself and the samples of the inputs u_1, u_2, $\cdots$, u_p. Identification of a **MIMO LTI** system with q outputs is equivalent to the identification of q **MISO LTI systems**.

Therefore, we can focus on the problem of set-membership identification of MISO LTI systems.

Set-membership identification of LTI MISO with p inputs

A-priori information about the system:

$$y = G\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \end{bmatrix}$$

where

$$G = [G_1, \, G_2, \, \dots, \, G_p]$$

and in turn,

$$G_i(q^{-1}) = \frac{\beta_{0i} + \beta_{1i}q^{-1} + \cdots + \beta_{pi}q^{-ni}}{1 + \alpha_{1i}q^{-1} + \cdots + \alpha_{qi}q^{-ni}}$$

here, ni, is the dynamical order of G_i.

A-priori information about the noise:
We assume an EIV structure for the noise affecting the data; that is,

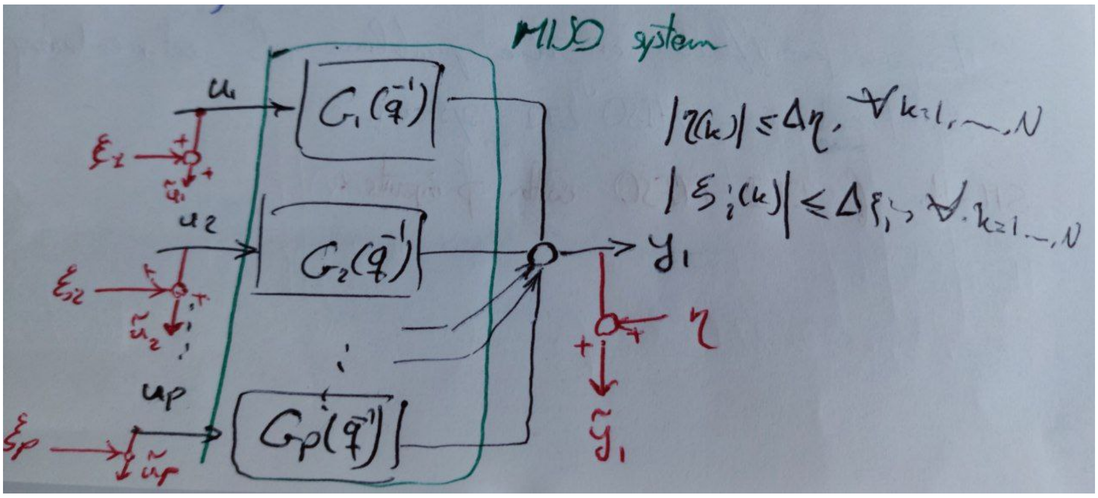


Figure 3.13: The scheme of the error entering the system in an EIV setting.

A-posteriori information: N samples of the inputs and the output sequences experimentaly collected.
Let's define the feasible parameter set:

$$\mathcal{D}_a = \left\{ \theta \in \mathbb{R}^{\sum_{k=1}^{2n+1}} : \begin{array}{l} y = G_1u_1 + G_2u_2 + \cdots + G_pu_p, \\ y = \tilde{y} + \eta, \\ u_1 = \tilde{u}_1 + \xi_1, \, \cdots \, , \, u_p = \tilde{u}_p + \xi_p, \\ |\xi_1| \leq \Delta\xi_1, \, \cdots \, , \, |\xi_p| \leq \Delta\xi_p, \, |\eta| \leq \Delta\eta, \, \forall k \in \mathbb{N} \end{array} \right\}$$

So,

$$\mathcal{D}_a = \left\{ \theta \in \mathbb{R}^{\sum_{k=1}^{2n+1}} : \begin{array}{l} y = \dfrac{\beta_0^1 + \beta_1^1 q^{-1} + \beta_2^1 q^{-2} + \cdots + \beta_n^1 q^{-n}}{1 + \alpha_1^1 q^{-1} + \alpha_2^1 q^{-2} + \cdots + \alpha_n^1 q^{-n}} u_1 + \cdots \\ + \dfrac{\beta_0^p + \beta_1^p q^{-1} + \beta_2^p q^{-2} + \cdots + \beta_n^p q^{-n}}{1 + \alpha_1^p q^{-1} + \alpha_2^p q^{-2} + \cdots + \alpha_n^p q^{-n}} u_p, \\ y = \tilde{y} + \eta, \\ u_1 = \tilde{u}_1 + \xi_1, \, \cdots, \, u_p = \tilde{u}_p + \xi_p, \\ |\xi_1| \leq \Delta\xi_1, \, \cdots, \, |\xi_p| \leq \Delta\xi_p, \, |\eta| \leq \Delta\eta, \, \forall k \in \mathbb{N} \end{array} \right\}$$

Chapter 4

Laboratory 01: solution

4.1 Problem 01

Upon successful completion of this homework, students will

1. Be able to compute Least Squares (‘2 norm) and Least ‘ norm parameter estimation for Discrete-time LTI systems.
2. Be able to analyze properties and limitations of the considered estimators

The plant to be estimated is a continuous-time LTI dynamical system assumed to be exactly described by the following transfer function:

$$G_p(s) = \frac{100}{s^2 + 1.2s + 1}$$

Assuming the sampling time $T_s = 1$, the following script is used for discretizing the transfer function of the sysetm, using *zoh*, or zero-order-hold method.

$$G_d = c2d(G_p, Ts,' zoh')$$

The result going to be of the following form:

$$G_d(z) = \frac{N(z)}{D(z)}$$

The result of this command is as follows:

$$G_d(z) = \frac{32.24z + 21.41}{z^2 - 0.7647z + 0.3012}$$

Now, the input of the system is considered to be samples of a uniformly distributed random variable. The command $u = rand(N, 1)$ is used at this end. N is considered to be 1000. Then, the command $y =$*lsim*(G_d, u, Ts) is used to simulate **noise-free** output samples. Here, it is assumed that the input of the system is exactly known.

θ_{LS} in ideal noise-free setting

The regression matrix A and output vector b, are shaped as follows. It is worthwhile to mention that, in this case, since the samples are noise free, $N = 7$ suffice to obtain exactly the the values of the parameters.

$$y_{NoiseFree} = A_{NoiseFree}\theta$$

where $y = [y(3)y(4)...y(N)]^T, \theta = [\theta_1\theta_2...\theta_5]^T)$, and

$$A = \begin{bmatrix} -y(2) & -y(3) & u(3) & u(2) & u(1) \\ -y(3) & -y(2) & u(4) & u(3) & u(2) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -y(N-1) & -y(N-2) & u(N) & u(N-1) & u(N-2) \end{bmatrix}$$

The command used for deriving $\theta's$ is:

$$\theta_{LS} = y\ A$$

It can be seen that the parameters obtained in this way are exactly parameters of the discretized transferfunction, except for arithmatic roundings.

$$\begin{bmatrix} \theta_{LS} & \theta_{Gd} \\ -0.7647 & -0.7647 \\ 0.3012 & 0.3012 \\ -0.0000 & 0 \\ 32.2369 & 32.2369 \\ 21.4103 & 21.4103 \end{bmatrix}$$

θ_{LS} in Equation-Error setting

In the second part, it is asked to repeat the process of estimation, simulating an error that affect the equation. In this case the collected measurement y are give by:

$$D_d(q^{-1})y_t = N_d(q^{-1})u_t + e_t$$

This assumption about the noise structure is theoretical, and it is not according to read data acquisition setting, which is Error-in-Variable setting or, assuming input to be exactly known Output-Error setting

Here, the error, e, is considered to be a normaly distributed noise with standard deviation 5.

$$e = 5 * randn(N, 1)$$

To create output data samples that is affected in this manner, the noise entering each output sample should be filtered by $D(z)$the denuminator of G_d. Therefore the following MATLAB code should be used:

$$[\text{num, den}] = tfdata(Gd,'v')y\_EE = lsim(Gd, u) + lsim(tf(1, den), e)$$

In this case, the regression matrix is shaped by the noisy output samples. Hence: $y_{EE} = [y_{EE}(3)\ y_{EE}(4)...y_{EE}(N)]^T$

$$A = \begin{bmatrix} -y_{EE}(2) & -y_{EE}(3) & u(3) & u(2) & u(1) \\ -y_{EE}(3) & -y_{EE}(2) & u(4) & u(3) & u(2) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -y_{EE}(N-1) & -y_{EE}(N-2) & u(N) & u(N-1) & u(N-2) \end{bmatrix}$$

The result of the Least square algorithm is going to be as follow:

$$\begin{bmatrix} \theta_{EE} & \theta_{Gd} \\ -0.7762 & -0.7647 \\ 0.3116 & 0.3012 \\ 0.2078 & 0 \\ 32.2405 & 32.2369 \\ 21.0843 & 21.4103 \end{bmatrix}$$

As it can be seen, the result is very close to the observed values. Since this structure of the noise satisfies both assumptions of the consistency property of Least Squares, it is guaranteed that as N tends to infinity, the values of θ will converge to the true values, regardless of how large the standard deviation of the noise affecting

θ_{LS} in Output-Error setting

The noise in this setting directly affect the output measurements

$$y_t = G_d(q^{-1})u_t + \eta_t$$

This setting is a good structure compliant with the real data acquisition setting.

Here, the error, η_t, is considered to be a normaly distributed noise with standard deviation 5.

$$eta = 5 * randn(N, 1)$$

To create output data samples that is affected in this manner, the following MATLAB code should be used:

$$y\_OE = lsim(Gd, u) + eta$$

In this case, the regression matrix is shaped by the noisy output samples. Hence: $y_{OE} = [y_{OE}(3)\; y_{OE}(4) ... y_{OE}(N)]^T$

$$A = \begin{bmatrix} -y_{OE}(2) & -y_{OE}(3) & u(3) & u(2) & u(1) \\ -y_{OE}(3) & -y_{OE}(2) & u(4) & u(3) & u(2) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -y_{OE}(N-1) & -y_{OE}(N-2) & u(N) & u(N-1) & u(N-2) \end{bmatrix}$$

The result of the Least square algorithm is going to be as follow:

$$\begin{bmatrix} \theta_{OE} & \theta_{Gd} \\ -0.5369 & -0.7647 \\ 0.1352 & 0.3012 \\ -0.2887 & 0 \\ 32.0653 & 32.2369 \\ 28.1921 & 21.4103 \end{bmatrix}$$

Since this noise structure does not satisfy the second assumption of the consistency property of the least square method, no matter how large N is, the estimation values are not going to converge to the real values.

4.2 Problem 02

Now, we want to convert the minimization of our cost function in a linear programming problem.

$$\theta_\infty = \arg \min_{\theta \in \mathbb{R}^5} \|\tilde{y} - A\theta\|_\infty$$

Considering $\gamma_j = b_j - a_j\theta$, where a_j is the j^{th} row of matrix A, based on the definition of the infinity norm, we have:

$$\|\gamma\|_\infty = \max\left(|\gamma_1|, |\gamma_2|, \ldots, |\gamma_{N-n}|\right)$$

Thus, the optimization problem can be written as:

$$\theta_\infty = \arg \min_{\theta \in \mathbb{R}^5} \max\left(|\gamma_1|, |\gamma_2|, \ldots, |\gamma_{N-n}|\right)$$

However, this problem is not linear, so we introduce a slack variable t and reformulate the optimization as:

$$\theta_\infty = \arg \min_{\theta \in \mathbb{R}^5, t \in \mathbb{R}} t \quad \text{subject to:}$$

$$\begin{cases} |\gamma_1| \le t \\ |\gamma_2| \le t \\ \vdots \\ |\gamma_{N-n}| \le t \end{cases}$$

This ensures the problem is now linear. Now, we add this extra parameter t as the 6th parameter to θ, and matrices A and b for the MATLAB function `linprog` can be constructed accordingly.

For each constraint $|\gamma_j| \leq t$, we rewrite it as:

$$|b_j - a_j\theta| \leq t$$

This leads to two inequalities for each j:

$$b_j - a_j\theta \leq t \quad \text{and} \quad -(b_j - a_j\theta) \leq t$$

So we obtain:

$$[-a_j \quad -1]\begin{bmatrix}\theta \\ t\end{bmatrix} \leq -b_j \quad \text{and} \quad [a_j \quad -1]\begin{bmatrix}\theta \\ t\end{bmatrix} \leq b_j$$

We aim to minimize the ℓ_1-norm:

$$\|\gamma\|_1 = \sum_{j=1}^{N-n} |\gamma_j| = \sum_{j=1}^{N-n} |b_j - a_j\theta|$$

To linearize the problem, we introduce slack variables s_j such that:

$$b_j - a_j\theta \leq s_j \quad \text{and} \quad -(b_j - a_j\theta) \leq s_j$$

Thus, the optimization problem becomes:

$$\min_{\theta\in\mathbb{R}^5, s\in\mathbb{R}^{N-n}} \sum_{j=1}^{N-n} s_j$$

subject to:

$$\begin{cases} b_j - a_j\theta \leq s_j \\ -(b_j - a_j\theta) \leq s_j \end{cases}$$

In matrix form, this can be written as:

$$\begin{bmatrix} A & I \\ -A & I \end{bmatrix}\begin{bmatrix}\theta \\ s\end{bmatrix} \leq \begin{bmatrix} b \\ -b \end{bmatrix}$$

where A is the matrix with rows a_j, I is the identity matrix, and b is the vector of known values.

Chapter 5

Laboratory 02: solution

5.1 the first problem

In this problem we assume that the plant to be identified is exactly described as a discrete-time LTI models described by the following transfer function (in the q^{-1} operator):

$$G_p(q^{-1}) = \frac{\theta_2}{1 + \theta_1 q^{-1}}$$

where

$$\theta = [-0.52]$$

Now, it is required to compute a set-membership estimation of the parameters of the system by means of the following procedure:
Assume that the uncertainty affecting the input output data enters the problem according to an equation error structure. Furthermore, to perform the simulation, assume that collected input sequence $\tilde{u}$ and the unknwon equation error sequences e are as follows :

$$\tilde{u} = [4,\ -3,\ 2,\ 1]^T$$

and

$$e = [0.05,\ -0.25,\ 0.3,\ -0.5]^T$$

Perform a simulation of model in order to collect the input and output data to be used for identification. The output sequence *tildey* has to be computed according to the following equation (equation error structure):
Based on the transfer function at hand the difference equation of the system is as follow:

$$y(k) = -\theta_1 y(k-1) + \theta_2 u(k)$$

Considering an *Equation-Error* structure for the noise, the following relationship is obtained:

$$\tilde{y}(k) = -\theta_1 \tilde{y}(k-1) + \theta_2 u(k) + e(k)$$

Where it is assumed that the error e is known to be bounded as $|e(k)| \leq \Delta e$ and assuming as initial condition $\tilde{y}$.

Considering the difference equation of the system, the following information can be obtained:

- $n_a = 1$ which represent the order of the system, which corresponds to the number of previous samples of y needed to express the difference equation of the system.
- n_b the number of input samples in the difference equation.
- $t_{min} = \max([n_a + 1, n_b]$

MATLAB code for simulation of the system is as follows. Take into account that, alternatively, we could use:

$$\tilde{y} = \mathrm{lsim}(G_p, u) + \mathrm{lsim}(\mathrm{tf}(1, D, 1), e);$$

```
close all
clear variables
clc
format compact

%% Defining system
z = tf('z',1);
theta = [-0.5 2];
Gp = theta(2)/(1 + theta(1)*z)

u_tilde = [4 -3 2 1]';
e = [0.05 -0.25 0.3 -0.5]';

%% Simulation
% y_tilde(k) = -theta_1*y_tilde(k-1) + theta_2 u_tilde(k) + e(k)
na = 1; % The number of state variables or last samples of y
nb = 1; % The number of past inputs
N = 4;
t_min = max([na+1 , nb]);
y_tilde = zeros(4,1);
% Initial value

for k = t_min:N
    y_tilde(k) = -theta(1)*y_tilde(k-1) + theta(2)*u_tilde(k) + e(k);
end

%% Upper and lower bound
delta_e = 0.5;
y_up = y_tilde + delta_e;
y_low = y_tilde - delta_e;
plot(1:N, [y_up, y_tilde, y_low])
```

Listing 5.1: Set-membership estimation with theta bounds and simulation

In order to obtain *Feasible Uncertainty Set*, it is required to do the following manipulations, considering the definition of FPS we have:

$$\mathbb{D}_\theta = \{\theta | \tilde{y}(k) + \theta_1 \tilde{y}(k-1) + \theta_2 \tilde{u}(k)| \leq \Delta e \forall k\}$$

Writing exampding the equations for t_{min} to $N = 4$, which is the number of samples, we obtain:

$$
\begin{aligned}
|\tilde{y}(2) + \theta_1 \tilde{y}(1) + \theta_2 \tilde{u}(2)| &\leq \Delta e \\
|\tilde{y}(3) + \theta_1 \tilde{y}(2) + \theta_2 \tilde{u}(3)| &\leq \Delta e \\
|\tilde{y}(4) + \theta_1 \tilde{y}(3) + \theta_2 \tilde{u}(4)| &\leq \Delta e
\end{aligned}
$$

and therefore,

$$
\begin{aligned}
-\Delta e - \tilde{y}(2) \geq \theta_1 \tilde{y}(1) + \theta_2 \tilde{u}(2) \leq \Delta e - \tilde{y}(2) \\
-\Delta e - \tilde{y}(3) \geq \theta_1 \tilde{y}(2) + \theta_2 \tilde{u}(3) \leq \Delta e - \tilde{y}(3) \\
-\Delta e - \tilde{y}(4) \geq \theta_1 \tilde{y}(3) + \theta_2 \tilde{u}(4) \leq \Delta e - \tilde{y}(4)
\end{aligned}
$$

In each of these relationships, each inequality is a constraint in parameter space that needs to be satisfied so that the equation error becomes less that Δe. For finding $\mathbb{D}_\theta$ in the parameter space, we can consider inequalities as equality, and by doing so, we obtain the equation of 3 pairs of parallel lines on the parameter space, which in this case is a two-dimensional plane. After writing the following MATLAB code, the intersections of all the regiones between the top bound and lower bounds lead to FPS. The plot of the feasible uncertaintly set in the parameter space is as follows:

```
%% Theta_1 variation
theta_1 = linspace(-5, 5, 1000);
figure,
hold on
for k = t_min:N
    theta_2_up = (+delta_e + y_tilde(k) + theta_1*y_tilde(k-1)) / u_tilde(k);
    theta_2_low = (-delta_e + y_tilde(k) + theta_1*y_tilde(k-1)) / u_tilde(k);
    plot(theta_1, theta_2_low, 'b')
    plot(theta_1, theta_2_up, 'r')
end

%% Brute-force approach (not recommended)
figure,
hold on
for theta_1 = -100:0.01:100
    for theta_2 = -100:0.01:100
        for k = t_min:N
            f = 1; % Flag
            if(abs(y_tilde(k) + theta_1*y_tilde(k-1) - theta_2*u_tilde(k)) > delta_e)
                f = 0;
                break
            end
        end
        if (f == 1)
            plot(theta_1, theta_2, '+');
        end
    end
end
```

It can be seen that the "real value" of the parameters is one of point at the top left corner of the $\mathbb{D}_\theta$.

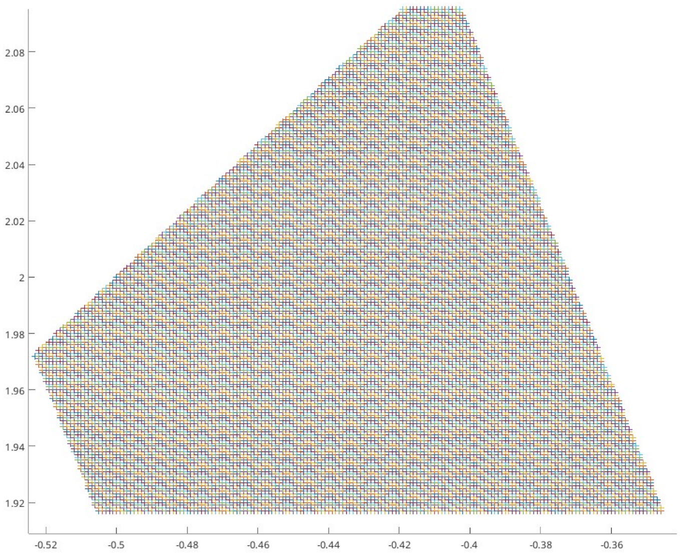


Figure 5.1: FPS represented in the parameter space; the horizontal axis is θ_1 and the vertical axis is θ_2

5.2 The second problem

In this problem we assume that the plant to be identified is exactly described as a discrete-time LTI models described by the following transfer function:

$$G_p(q^{-1}) = \frac{N(q^{-1})}{D(q^{-1})} = \frac{\theta_3 + \theta_4 q^{-1} + \theta_5 q^{-2}}{1 + \theta_1 q^{-1} + \theta_2 q^{-2}} \tag{5}$$

where:

$$\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5] = [-0.7647, 0.3012, 0, 32.24, 21.41]$$

Assuming that the uncertainty affecting the input-output data enters the problem according to an equation error structure, the following simulation is to be performed. The input sequence $u(t)$ is a random sequence uniformly distributed in $[0, 1]$, and the output sequence $y(t)$ is computed using the following equation:

$$y(t) = -\theta_1 y(t-1) - \theta_2 y(t-2) + \theta_3 u(t) + \theta_4 u(t-1) + \theta_5 u(t-2) + e(t) \tag{7}$$

where $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]$ are the model parameters, and $e(t)$ is a random sequence uniformly distributed in $[-\Delta e, \Delta e]$. The goal is to collect input and output data to be used for identification purposes.

The simulation setup includes:

- A random input sequence $u(t)$, uniformly distributed in $[0, 1]$.
- A random error sequence $e(t)$, uniformly distributed in $[-\Delta e, \Delta e]$.

- Output $y(t)$ calculated according to the equation error model.

```
clear variables
close all
format compact
rng('default')
%% defining the system
%LTI 2nd order, a-priori info
theta = [-0.7647,0.3012,0,32.24,21.41];

num = [theta(3) theta(4) theta(5)]
den = [1 theta(1) theta(2)];

Gp = tf(num,den,-1);

%% Simulation
N = 1000;
na = 2;
nb = 3;
tmin = max([na+1,nb]);

y = zeros(N,1);
u = rand(N,1);
delta_e = 0.5; %consider 0.1, 1, 10, 100
for t = tmin:N
    e = 2*delta_e*rand -delta_e; %uniform between[-delta_e,delta e]
    y(t) = -theta(1)*y(t-1) -theta(2)*y(t-2) + theta(3)*u(t) + theta(4)*u(t-1)+...
        theta(5)*u(t-2)+ e;
end
```

Listing 5.2: Set-membership estimation with theta bounds and simulation

The relation for feasible parametric set is as follows:

$$\mathbb{D}_\theta = \{\theta \in \mathbb{R}^5 : \tilde{y} - e \leq y \leq \tilde{y} + e \ \text{ s.th. } \ |e(.)| \leq \Delta_e\}$$

Then, we can formulalte a linprog problem that can be solved as follows:

```
%the rest of the code
%% PUI problem solving by linprog
F = eye(5);
A1 = [-y(tmin-1:N-1), -y(tmin-2:N-2) u(tmin:N) u(tmin-1:N-1) u(tmin-2:N-2)];
A =[A1;-A1]; %for linprog
b1 = y(tmin:N) + delta_e;
b2 = -y(tmin:N) + delta_e;
b = [b1;b2];

% options = optimoptions('linprog', 'ScaleProblem', 'obj-and-constr', 'Display', 'none')
    ;
% x = linprog(sign * F(i,:)', A, b, [], [], [], [], options);

PUI = zeros(5,2);
sign = -1; %initialization

for j = 1:2
    sign = -1*sign;
    for i = 1:5
        PUI(i,j) = F(i,:)*linprog(sign*F(i,:)',A,b);
    end
end

PUI
```

Listing 5.3: Set-membership estimation with theta bounds and simulation

Chapter 6

Laboratory 02b: solution

In order to solve this problem, *SparsePOP* and *Sedumi*, toolboxes should be installed, there exist one tutorial file.

consider the following problem:

$$x^* = \arg\min_x x_1^2 + x_2^2 - 4x_1 - 6x_2 + 13$$
$$\text{s.t.}$$

$$x_1x_2 - 7x_2 = 10$$

$$x_2 \geq -\frac{1}{2}x_1 + 1$$

The cost function here is a polynomial, which is a sum of *monomial* terms. Each monomial is of the following form:

$$\alpha x_1^{C_1} x_2^{C_2} \cdots x_n^{C_n}$$

Drawing the contour

```
clear; close all; clc;

%% Part 1: plot
x1_lw = -15; x1_up = +15;
x2_lw = -12; x2_up = +12;

x1 = linspace(x1_lw, x1_up);
x2 = linspace(x2_lw, x2_up);
% objective function
[X1,X2] = meshgrid(x1,x2);
Z = X1.^2 + X2.^2 - 4.*X1 - 6.*X2 + 13;
figure, contour(X1,X2,Z,5*(1:50)');

% equality constraint
x2_1 = linspace(0.01, x2_up);
x2_2 = linspace(x2_lw, -0.01);
x1_1 = (10+7.*x2_1)./x2_1;
x1_2 = (10+7.*x2_2)./x2_2;
hold on; plot(x1_1,x2_1, 'k',x1_2,x2_2, 'k');
axis([x1_lw,x1_up,x2_lw,x2_up]);
```

```
% inequality constraint
x2_ineq = -0.5*x1 + 1;
hold on; plot(x1, x2_ineq, 'b');
patch([x1(1); x1(end); x1_up; x1_lw], ...
    [x2_ineq(1); x2_ineq(end); x2_up; x2_up], ...
    'blue', 'FaceColor', 'blue', 'FaceAlpha', 0.1);
```

The resultant graph is as follows:

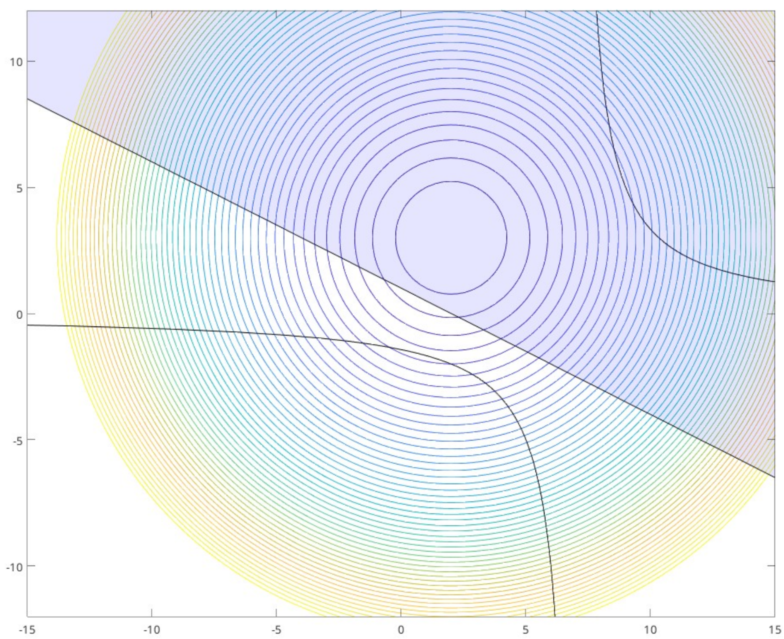


Figure 6.1: The contour of the polynomial as well as the graph of the constraints

Let's create the polynomial to be minized, named **objPoly**:

Definig the polynomila to be optimized

```
% objective function
objPoly.noTerms = 5;
objPoly.dimVar = 2;
objPoly.typeCone = 1;  %always to be set to 1
objPoly.degree = 2; %the degree of the poly
objPoly.supports = [2 0; 0 2; 1 0; 0 1; 0 0];
objPoly.coef = [1, 1, -4, -6, 13]'; %should be a column vector
```

Pay attention that the support matrix should be made according to the following scheme:

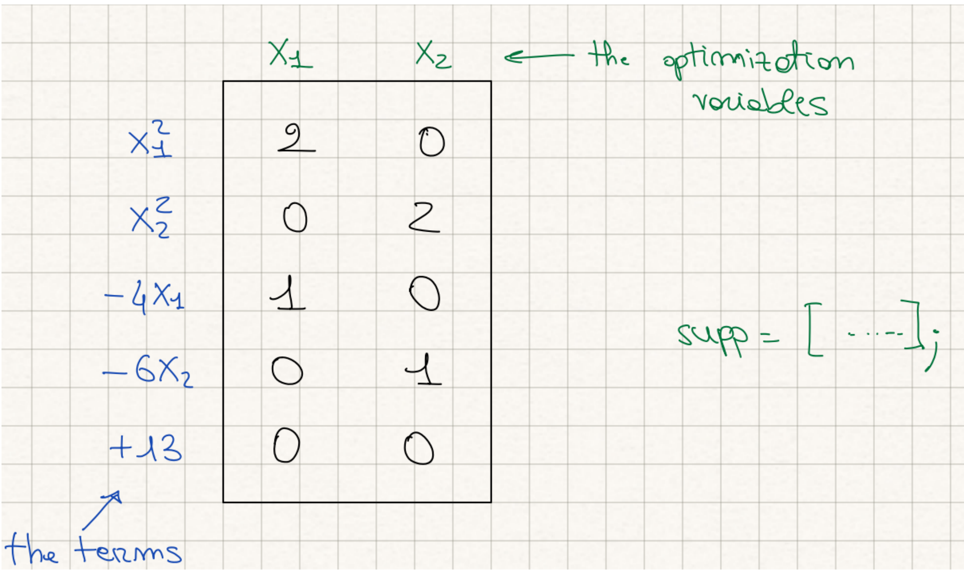


Figure 6.2: The contour of the polynomial as well as the graph of the constraints

The inequality and equality constraints should be defined as cell array structures. Pay attention that:

- **Equality**: .typeCone = -1
- **Inequality**: .typeCone = 1

Definig the polynomila to be optimized

```
% equality constraint
c = 1;
ineqPolySys{c}.noTerms = 3;
ineqPolySys{c}.dimVar = 2;
ineqPolySys{c}.typeCone = -1; % equality
ineqPolySys{c}.degree = 2;
ineqPolySys{c}.supports = [1 1; 0 1; 0 0];
ineqPolySys{c}.coef = [1 -7 -10]';
% inequality constraint
c = 2;
ineqPolySys{c}.noTerms = 3;
ineqPolySys{c}.dimVar = 2;
ineqPolySys{c}.typeCone = +1; % inequality
ineqPolySys{c}.degree = 1;
ineqPolySys{c}.supports = [0 1; 1 0; 0 0];
ineqPolySys{c}.coef = [1 0.5 -1]';
```

After defining the lower bounds and upper bounds for the optimization variables, the **relaxation order** should be chosen. For this lab, it is recommended that one tries with relaxation orders 1, 2, 3, 4 to solve the problem.

Solving the optimization problem

```
% lower bound
lbd = -1e10*ones(2,1);
% upper bound
ubd = +1e10*ones(2,1);

param.relaxOrder = 3;
param.POPsolver = 'interior-point';

[a,b,POP] = sparsePOP(objPoly, ineqPolySys, lbd, ubd, param);
sol_relaxed = POP.xVect;
sol_refined = POP.xVectL;

hold on;
plot(sol_relaxed(1),sol_relaxed(2),'r*');
plot(sol_refined(1),sol_refined(2),'m*');
```

For relaxation order 1 and 2, it can be seen that the solution of the optimization problem does not respect the constraints. That is, the solution is not acceptable.

With the relaxation order 3, the problem has an acceptable solution.

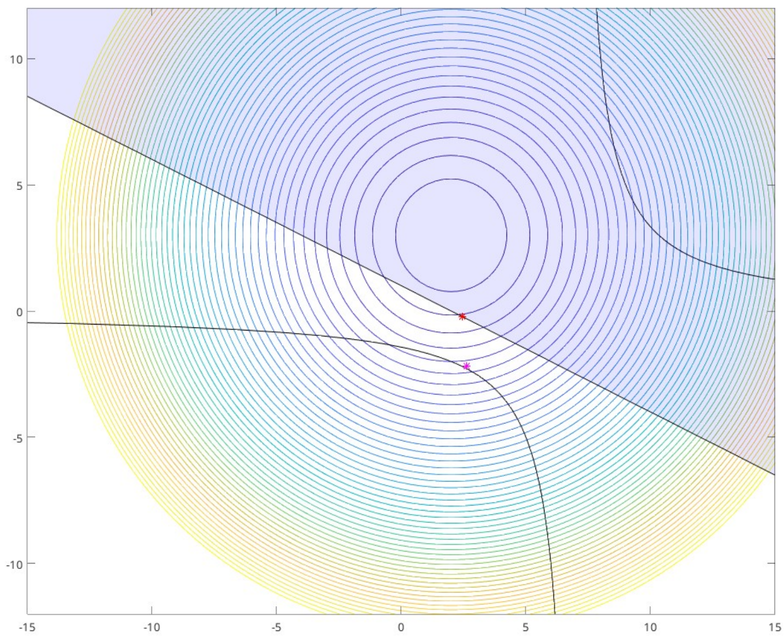


Figure 6.3: The solution of the problem with relaxation order 1, the red star spots the relaxed solution and the other star the refined solution, which is the result of sedumi

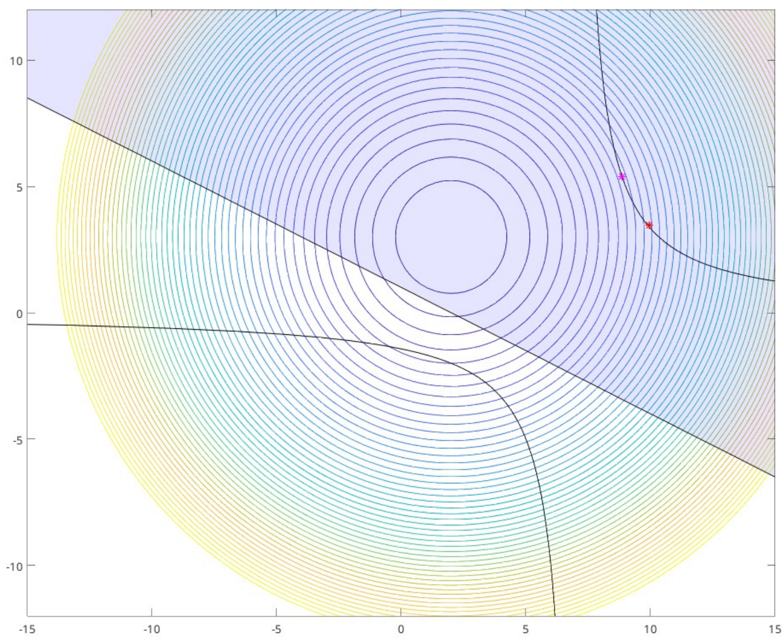


Figure 6.4: The solution of the problem with relaxation order 3