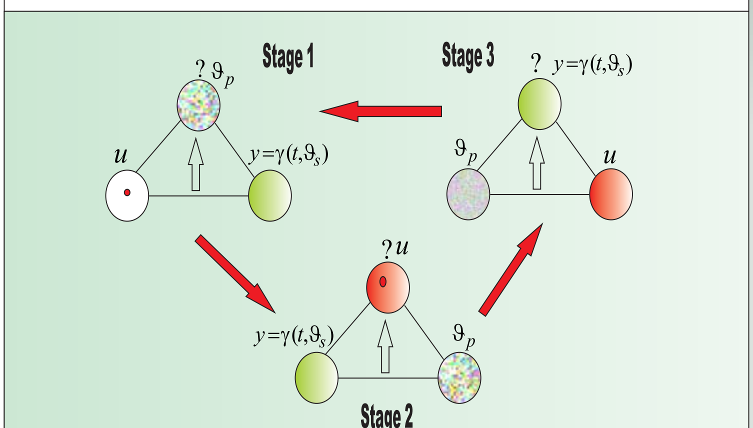
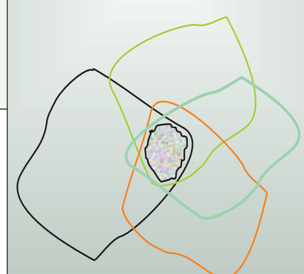
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

PHD THESIS

INTERVAL ROBUST CONTROL FOR NONLINEAR FLAT SYSTEMS





AUTHOR

JULIO CÉSAR
DEHESA VALENCIA

DIRECTOR
PHD. JESUS PICÓ I MARCO

DECEMBER, 2012

Summary

This thesis mainly focuses on the robust control of nonlinear flat systems. The main goal is to determine a family of robust controllers in order to assure the fulfillment of a desired specifications set under parametric uncertainty in the process. The family of robust controllers are determined with a new approach of robust possibilistic control together with the theory of flat systems. The specifications and parametric uncertainty are established through the intervals. Modal Interval Arithmetic and Analysis Quantified Sets Inversion Algorithms are applied to find solution sets. Different problems of robust control are solved such as: Solution sets referred to the attainable specifications by a family of controllers, as well as the determination of the maximum uncertainty admitted by a nominal controller.

This thesis develops a new methodology of robustness analysis of controllers based on differential flatness, where the use of a feedforward is required. The methodology developed is applied to different processes, especially to fed-batch bioreactors given the importance of these high density stirred tank reactors for efficient industrial production of proteins and enzymes.

Resumen

Esta tesis se enfoca principalmente en el control robusto de sistemas no lineales planos. El objetivo principal es determinar una familia de controladores robustos con la finalidad de asegurar el cumplimiento de un conjunto de especificaciones deseadas bajo incertidumbre paramétrica en el proceso. La familia de controladores robustos se determina con un nuevo enfoque de control robusto posibilistico conjuntamente con la teoría de los sistemas planos. Las especificaciones e incertidumbre paramétrica se establecen mediante intervalos. Se aplican la Aritmética Intervalar Modal y el Análisis de Algoritmos de Inversión de Conjuntos Cuantificados para encontrar los conjuntos de soluciones. Se resuelven diferentes problemas de control robusto tales como: Conjuntos de soluciones referidos a las especificaciones alcanzables por una familia de controladores, así como la determinación de la incertidumbre máxima admitida por un controlador nominal. En esta tesis se desarrolla una nueva metodología de análisis de robustez de controladores basados en platitud diferencial, donde el uso de una pre alimentación es requerida. La metodología desarrollada es aplicada a diferentes procesos, específicamente a bioreactores fed-batch, dada la importancia de estos reactores de alta densidad de tanque agitado para la producción industrial eficiente de proteínas y enzimas.

Resúm

Aquesta tesi es centra fonamentalment al control robust de sistemes no lineals plans. L'objectiu principal és determinar una família de controladors robustos amb la finalitat d'assegurar el compliment d'un conjunt d'especificacions desitjades baix incertesa paramètrica al procés. La família de controladors robustos es determina mitjançant un nou enfocament de control robust possibilistic, conjuntament amb l'ús de la teoria de sistemes plans. Les especificacions e incertesa paramètrica s'estableixen mitjançant intervals. S'apliquen la Aritmètica Intervalar Modal i l'Anàlisi d'Algorismes de Inversió de Conjunts Quantificats per a trobar els conjunts de solucions. Es resolen diferents problemes de control robust tals com trobar conjunts de solucions referides a les especificacions abastables per una família de controladors, així com la determinació de la incertesa màxima admesa per un controlador nominal. En aquesta tesi es desenvolupa una nova metodologia de anàlisi de robustesa de controladors basats en planitut diferencial, on es requereix l'ús d'una prealimentació. La metodologia desenvolupada s'aplica a diferents exemples de processos per tal avaluar la seva validesa.

Acknowledgment

I want to express my gratitude to all people and institutions that in one way or another were with me at all times to complete this work. First of all, I want to thank the Technological Institute of Comitancillo and the General Direction of Technological Higher Education (Mexico) for giving me the opportunity to continue studying. I am completely grateful with the Polytechnic University of Valencia for accepting me as a student in the doctoral program and to the National Association of Colleges of higher education (México) for the financial support granted. I express my gratitude to all the teachers who participated in the doctoral program. Thanks to all of them, I met and explored new fields of knowledge. In particular, I wish to thank to PhD. Jesús Picó I Marco for guiding me in the development of this work. I am completely grateful to him for his knowledge, experience and comments made to conclude this work. I want to thank also to PhD. Pau Herrero i Viñas for giving me his library over modal intervals and comments made in this thesis. I am completely grateful to all personalities of the international action office for their support and service in many stays that I had at the Polytechnic University of Valencia. In particular, thanks to D. Miguel Sánchez Pradas and Geraldine Bustamante. Thanks to my friends who shared with me many times. In particular to Juan, Miguel, Rene, Tomas, Gilberto and Carlos.

Finally, I dedicate this work with all my love to my family. Especially to my wife Elba del Carmen Cabrera Sibaja, to my daughters Lesley and Sharon and my son Jheshua Dehesa Cabrera.

Table of Contents

1 Justification, objectives and contributions

The effective control of practical time-varying systems with parametric uncertainties and external disturbances is one of the main topics in the study of the design for robust control systems. The general objective in the robust control of processes is to achieve the stability and robustness of the closed-loop system based on some suitable performance index. Generally, robust control of uncertain systems is achieved when they operate effectively over a specified range of system variations (eg. parametric variations). This deterministic approach contrasts sharply with many other adaptive control schemes, in which on-line identification and global parameter convergence properties are needed. Furthermore, no statistical information of the system variations is required to fulfill the desired robust dynamic behavior.

This thesis develops a new approach to robust possibilistic control of nonlinear flat systems with parametric uncertainty. The following problems are solved for these kind of systems: 1)To determine a family of possible controllers in which flat outputs can remain inside some specified regions, 2) To find a space of achievable specifications for a family of controllers and 3) To determine the maximum allowable uncertainty for a nominal controller.

These problems are addressed as Quantified Constraints Satisfaction Problems, which involve different elements such as: a set of quantified variables (specification parameters, process parameters), a set of domains, a set of constraints with inclusion relations and a set of algebraic functions obtained from the application of the theory of flat systems. The solutions to the previous problems are obtained through the aplication of Quantified Sets Inversion Algorithms.

1.1 Thesis's goals

The main objective can be subdivided into the following subgoals:

1. To find a family of control signals to ensure that the system output will stay within a set of desired values.
2. To formulate a Quantified Constraints Satisfaction Problem (QCSP) to find a family of controllers in order to guarantee the satisfaction of the specifications.

3. To formulate a QCSP to obtain solutions spaces for the attainable specifications by a family of controllers.
4. To formulate a QCSP to determine the maximum uncertainty admitted by a nominal controller.
5. To find solution sets of state feedback controller parameters for which robust performance holds.
6. To implement Quantified Sets Inversion Algorithms (QSIA) to get solution sets relating to the controller spaces, plant spaces and specification spaces.
7. To establish a methodology of robust controllers design for nonlinear flat systems.
8. To develop applications to show the validity of the approach.

1.2 Summary of the original contributions of this work

The main original contributions of this thesis are the following:

- To develop of a new methodology of robust controllers design for nonlinear flat systems.
- To propose some Quantified Constraints Satisfaction Problems for nonlinear flat systems.
- To solve main problems of robust control for nonlinear flat systems with Quantified Sets Inversion Algorithms.

1.3 Thesis organization

The solutions suggested in this thesis are presented and explained in the subsequent chapters which are structured as follows:

- Problem statement: Chapter 2. A subclass of invertible single-input single-output nonlinear dynamic systems are described and the general approach of the robust controllers design procedure is annunciated. It explains how to reconstruct the bounding regions of state variables and controllers from the output space. The general problems to be solved are exposed. First, the definition of hard and soft specifications is given. Then, the general problem of finding a family of robust controllers is presented. Other problems to solve are exposed, such as obtaining the set of attainable specifications by a family of controllers and problems related to the tunning parameters of feedback controllers for trajectory tracking. Finally, a basic example is developed.
- Preliminaries: Chapter 3. This Chapter includes a summary of the theory of flat systems. It describes the most important properties of flatness, its definition as well as differences between controllers based on differential flatness and controllers based on feedback linearization. It gives some

examples of systems that are flat as well as systems that are not. The Chapter summarizes flatness in the context of parametric uncertainty and the parameterization of control signals as a function of a region of flat outputs and its derivatives. Also, this chapter describes the process to solve optimization problems of an objective function based on differential flatness, and highlights the most important properties of Modal Interval Arithmetic. Potential problems that can be solved with this arithmetic and aspects of semantic interpretation problems in the design of robust controllers. Finally, it defines the elements involved in Quantified Constraints Satisfaction Problems.

- Approach to robust possibilistic control of nonlinear flat systems: Chapter 4. The approach of robust possibilistic control in terms of set inclusion is described. It proposes general approaches of Quantified Constraints Satisfaction Problems (design of robust controllers for nonlinear flat systems) considering the uncertainties in the process. It specifies the rules that must be met in a Quantified Sets Inversion Algorithm to get solution spaces relating controllers, plants and spaces of achievable specifications. Finally, the semantic of design for tuning the parameters of a feedback controller are given. The controller structure based on differential flatness is obtained. The controller parameters obtained ensure that the output of the feedback system are inside some specification intervals.
- Applications: Chapter 5. As applications, examples to linear and nonlinear systems are developed. For both cases, families of robust controllers and attainable specifications by some elements of the family are obtained. The fulfillment of specifications under parametric uncertainty are verified by means of robustness tests.
- Conclusions and future works: Chapter 6. General conclusions and future works are mentioned.

2 Problem statement

2.1 Introduction to the nonlinear flat systems

This thesis mainly focuses on proposing and solving different problems that might arise in the robust control of nonlinear flat systems using a possibilistic approach. The main objective is maintain the output of the controlled system within a specifications region under variations of the plant parameters.

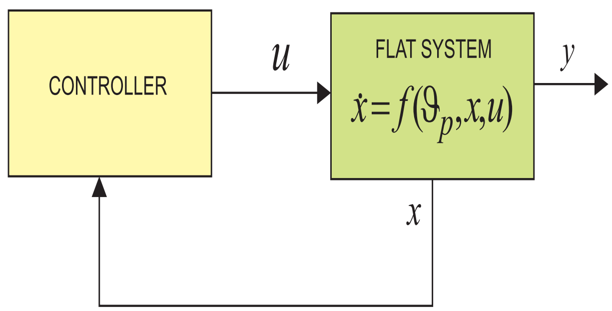


Fig. 2.1. Controlled system.

A subclass of invertible single-input single-output nonlinear dynamic systems will be considered. The system dynamic model is defined by a set of ordinary differential equations of the form $\dot{x} = f(\vartheta_p, x, u)$, where x is the state vector, u the control signal and ϑ_p the set of uncertain plant parameters. These systems are known as differentially flat systems. Differentially flat systems are used in situations where it is required that the trajectory of the system follow a desired output. When a system is flat means that we can move from output space to the input space and viceversa, as it is indicated in Figure 2.2. We can see in the output and input space a single point at each end. These points represent single trajectories.

These systems have the property of admitting an algebraic equivalent representation of the dynamic system. The control input and the state variables are reconstructed from the output and output derivatives **Fliess et al. (1995b)**; **Sira-Ramírez and Agrawal (2004)** as it is depicted in Figure 2.3

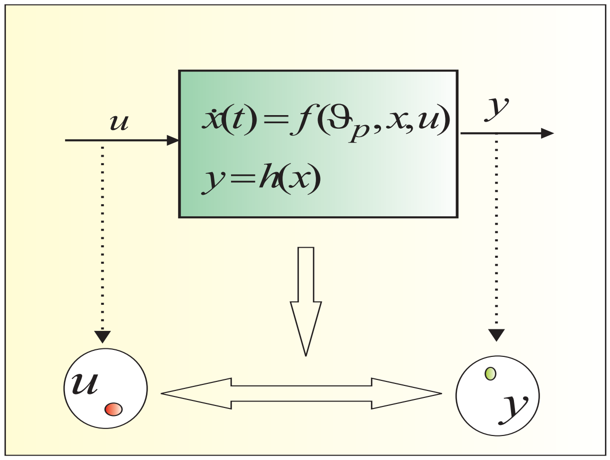


Fig. 2.2. Differentially flat system.

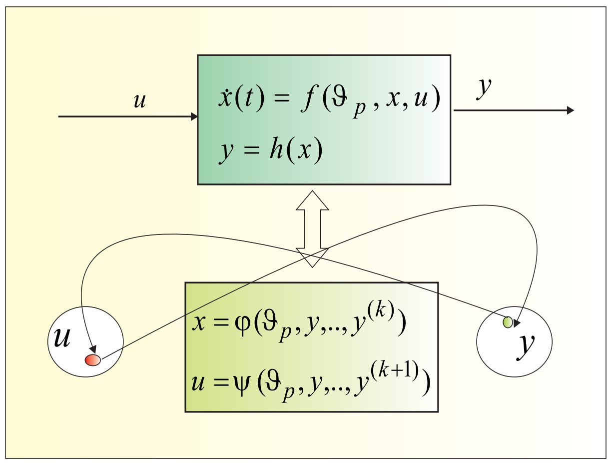


Fig. 2.3. Algebraic equivalent representation of the dynamic system.

The flat output functions can be constructed and planned with polynomial functions depending on the time and specification parameters. These functions are derived with respect to time and from them we can reconstruct all the variables of the system, as it is depicted in Figure 2.4. The reconstruction of state variables and control signal are commonly obtained considering a single trajectory for the flat output, a nominal plant $\overline{\vartheta}_p$ and a single point of the specification parameters $\overline{\vartheta}_s$. The control signal obtained via inversion or via flatness is named feedforward controller.

If the control signal (feedforward controller) is fixed to the input of the dynamic system and the initial conditions of the system are known, then we

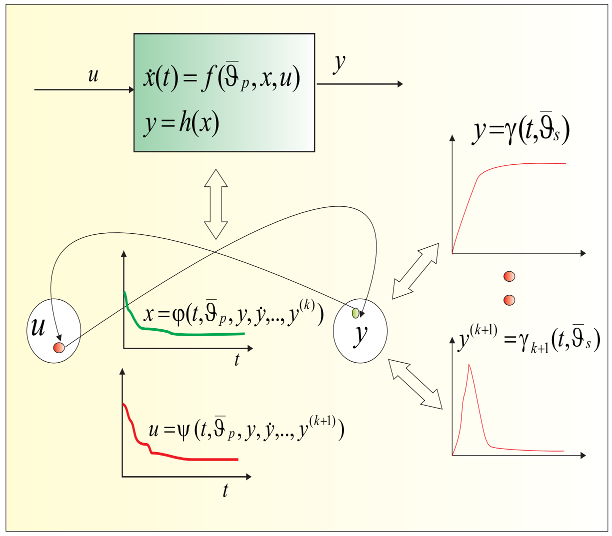


Fig. 2.4. Inversion of the dynamics of the system.

can back to the output space. A representation is shown in Figure 2.5. If during the operation of the system, there are variations in the plant parameters, it is possible that the feedforward controller can not drive the system to the planned trajectory. In the controller design technique based on differential flatness, the feedforward controller is used to take the system to the planned trajectory and to correct errors due to changes in plant parameters and errors in the measurements, a new feedback control input is imposed such that the dynamics associated with the error is asymptotically stable. This is achieved through a proper selection of design parameters, properly placing in the left half-plane of the complex plane, the roots of the characteristic polynomial associated with the error. The controller design technique based on differential flatness is similar to the controller design technique based on feedback linearization in the sense that in both techniques the output is derived several times until the control signal appears as well as the procedure to set the parameters of the feedback controller in both techniques are similar. It should be noted that the flatness property can be combined with other controller design techniques such as: sliding mode, passivity, based on Lyapunov, backstepping, among many others.

We consider the controller design technique based on flatness together with a possibilistic approach to determine feedback control laws that are robust under variations of the plant parameters and ensure that the output of the process, states and controllers are within a prespecified region. In the following section the approach is explicated.

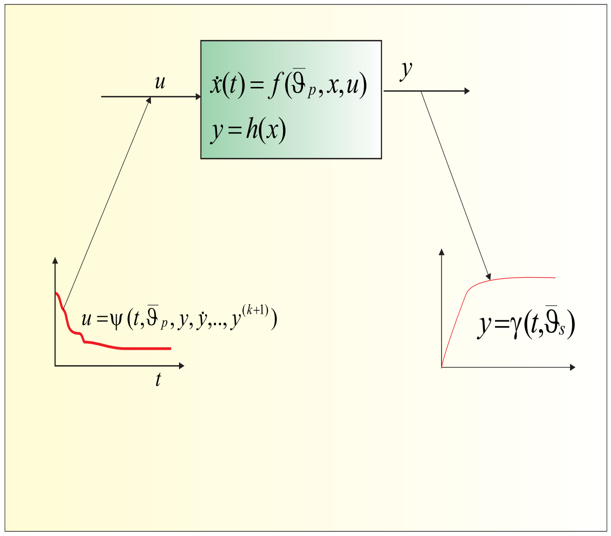


Fig. 2.5. Controlled dynamical system in open-loop.

2.2 General approach of the robust controllers design procedure

2.2.1 Basic notations

The general approach of possibilistic robust control for fuzzy plants can be consulted in **Bondia et al. (2005)**. The main idea is to propose a set of hard and soft specifications for the process output in terms of intervals. For instance, the fuzzy set depicted in Figure 2.6, it represents a specifications set parametrized by the parameters $\widetilde{\theta}_s$. The hard specification corresponds to the core and the soft specification to the support of the fuzzy set.

In this thesis we will only focus on the two cut levels of the fuzzy specifications set: the cuts $\alpha = 0$ (support) and $\alpha = 1$ (core). The support corresponds to soft specifications. That is, for any process uncertainty within a given set, one wants to ensure that at least the soft specifications will be fulfilled. The core corresponds to the most restricted specifications; the hard ones. One would ideally like that these are fulfilled as far as possible.

We will use some notations and representations of the parameters as follows: ϑ_p is a plant parameters set, $\bar{\vartheta}_p$ is a nominal plant, ϑ_s is a set of specification parameters, $\bar{\vartheta}_s$ is a fix point of specification parameters, ϑ_k is a set of controller parameters, $\bar{\vartheta}_k$ is a nominal controller. So, the parameters that represent sets will be represented as shaded regions and parameters that are fixed points will be represented as a single point within a region. A representation can be seen in Figure 2.7.

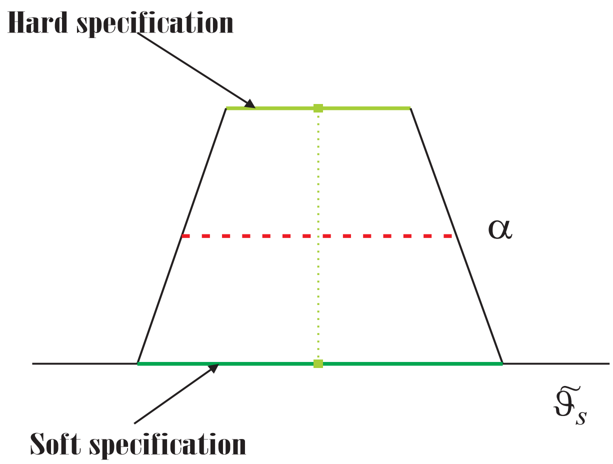


Fig. 2.6. Specifications described as a fuzzy set.

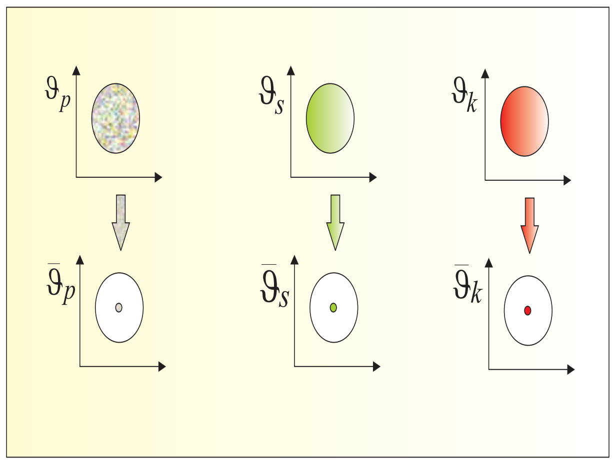


Fig. 2.7. Shaded regions represent sets and a single point inside a region represents a specific parameter.

In this thesis, we will consider that the specifications are valued trajectory set in time, parametrized by θ_s as it is indicated in Figure 2.8.

The function that defines the specifications is smooth, continuous in time and differentiable.

In Figure 2.9 we can see that a point $\bar{\vartheta}_s$ in the specifications space corresponds to a nominal trajectory $y = \gamma_y(t, \bar{\vartheta}_s)$ in the output space.

On the other hand, a hard or soft specification ϑ_s represents a specifications region $y = \gamma_y(t, \vartheta_s)$ in the output space as it is depicted in Figure 2.10.

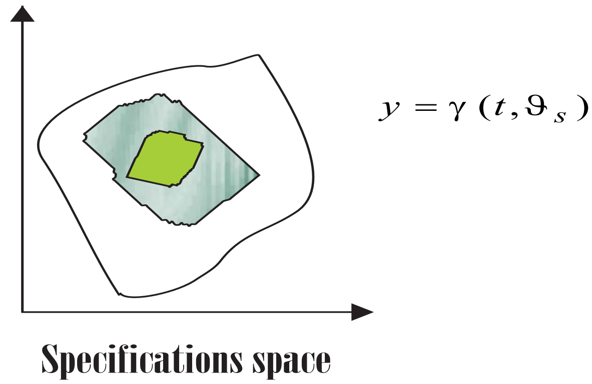


Fig. 2.8. Specifications space.

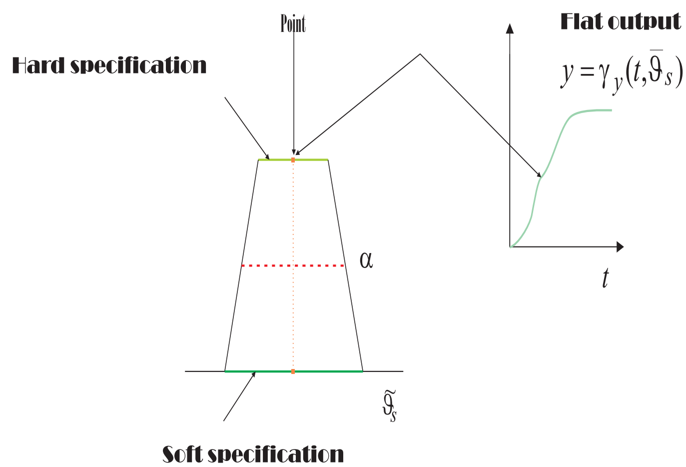


Fig. 2.9. A point $\bar{\vartheta}_s$ in the specifications space corresponds to a nominal trajectory $y = \gamma_y(t, \bar{\vartheta}_s)$ in the output space.

2.2.2 Expanding the output space

In order to maintain the trajectories of the system output inside two bounding regions, the approach starts by expanding the output space from hard and soft specifications. In the output space are included the polynomial functions that define the flat output and its derivatives. The output and its derivatives are considered as trajectories regions in time, parametrized with interval parameters. In Figure 2.11 a representation is indicated.

2.2.3 Expanding the state and input space from output space

The next step of the approach is to obtain a bounding region for states and control input. From hard and soft specifications we obtained the output

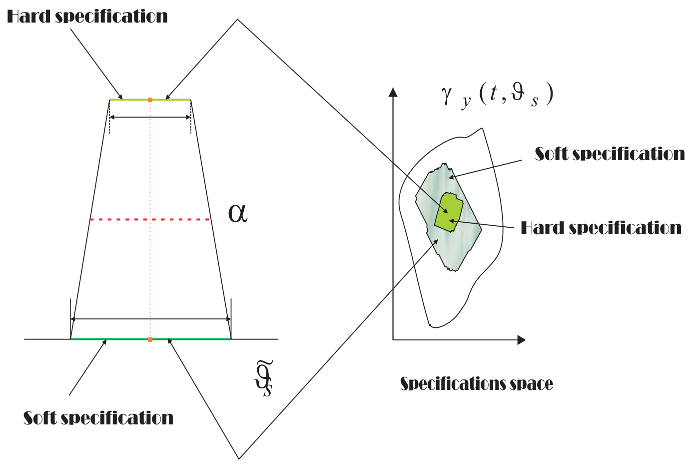


Fig. 2.10. The hard or soft specifications ϑ_s represent trajectory regions $y = \gamma_y(t, \vartheta_s)$ in the output space.

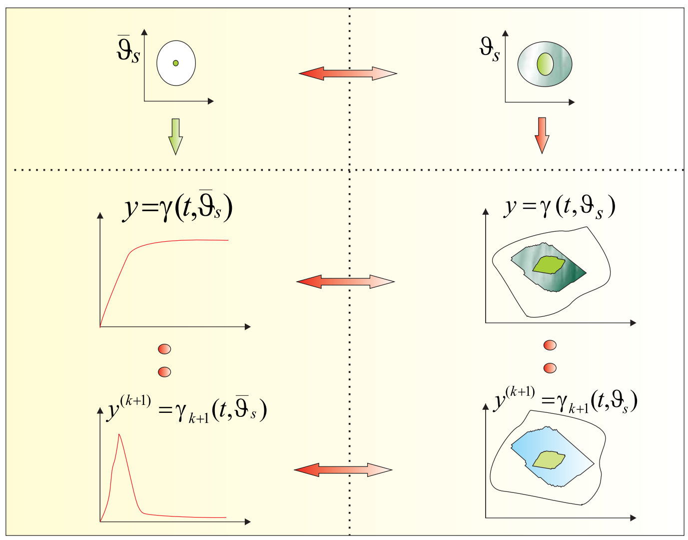


Fig. 2.11. Expanding the output space from hard and soft specifications.

space and from output space we can obtain the states and input space. These regions are obtained considering the output space, the equivalent algebraic representation of the dynamic system as well as a nominal plant.

In Chapter 3, we will see that if the flat output and its derivatives are trajectories regions in time, the reconstruction of the state variables and controllers, are trajectory regions in time. These trajectory regions are in function of specification parameters and plant parameters. So, $y = \gamma_y(t, \vartheta_s)$,

$x = \varphi(t, \vartheta_s, \vartheta_p)$ and $u = \psi(t, \vartheta_s, \vartheta_p)$. If $\vartheta_k = \{\vartheta_s, \vartheta_p\}$, then the controllers can be expressed as $u = \psi(t, \vartheta_k)$. The reconstruction of the bounding regions of the state variables and controllers in function of specification parameters and nominal plant is indicated in Figure 2.12.

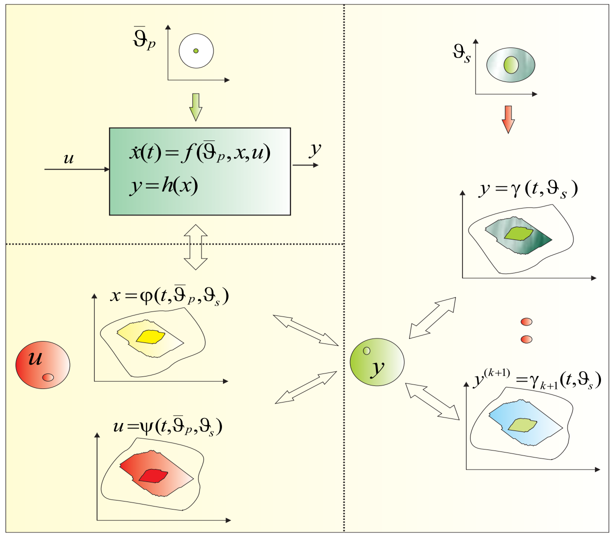


Fig. 2.12. Reconstruction of bounding regions of state variables x and controllers u in function of specification parameters ϑ_s and nominal plant $\bar{\vartheta}_p$.

As the equivalent algebraic system is reused in different operations, on some cases with specific parameters, on other cases with specific parameters and parameters that represent sets, the bounding regions of the output, states and controllers are saved to different bounding variables. The bounding region of the output flat $y = \gamma(t, \vartheta_s)$ is saved to $\gamma_y(t, \vartheta_s)$, the bounding region of the states variables $x = \varphi(t, \vartheta_s, \overline{\vartheta}_p)$ to $\gamma_x(t, \vartheta_s, \overline{\vartheta}_p)$ and the bounding region of the controllers $u = \psi(t, \vartheta_s, \overline{\vartheta}_p)$ to $\gamma_u(t, \vartheta_s, \overline{\vartheta}_p)$. $\gamma_y(t) = [\underline{\gamma}_y(t, \vartheta_s), \overline{\gamma}_y(t, \vartheta_s)]$, $\gamma_x(t) = [\underline{\gamma}_x(t, \vartheta_s, \overline{\vartheta}_p), \overline{\gamma}_x(t, \vartheta_s, \overline{\vartheta}_p)]$ and $\gamma_u(t) = [\underline{\gamma}_u(t, \vartheta_s, \overline{\vartheta}_p), \overline{\gamma}_u(t, \vartheta_s, \overline{\vartheta}_p)]$ are the upper and lower limits of the bounding regions of the flat output, state variables and controllers respectively. A representation is indicated in Figure 2.13.

With the finality that trajectories of the process output c_y, states φ and controllers ψ are within its bounding regions γ_y, γ_x and γ_u we can expand the parameter space of the plant ϑ_p to study the robustness of a feedforward controller $u = \psi(t, \overline{\vartheta}_s, \overline{\vartheta}_p)$. A representation is indicated in Figure 2.14. In general cases, the plant parameters can be included in all functions c_y, φ and ψ. This means that a plant of the solution set is one that satisfies the inclusion relation $c_y \subseteq \gamma_y \wedge \varphi_x \subseteq \gamma_x \wedge \varphi_u \subseteq \gamma_u$. This specific case corresponds

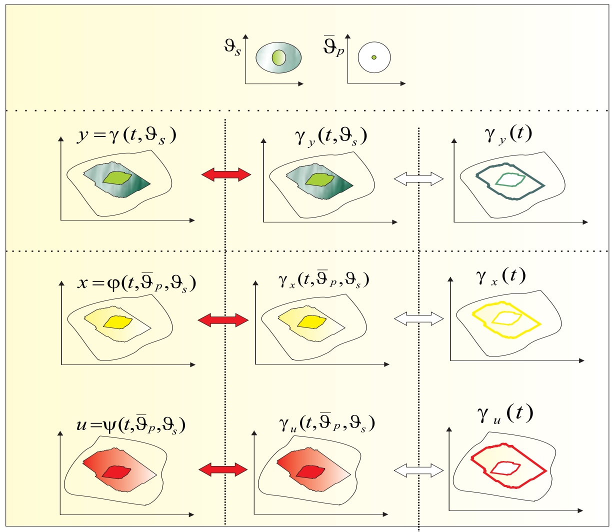


Fig. 2.13. Bounding regions of the flat output $\gamma_y(t)$, state variables $\gamma_x(t)$ and controllers $\gamma_u(t)$.

to a process output that depends on the state variables and plant parameters $y = h(x, \vartheta_p)$. Other cases where the process output depends explicitly on the state variables, control input and plant parameters $y = h(x, u, \vartheta_p)$ can be consulted in Chapter 3.

Since the flat output will be in function of the state variables, there will be an interaction of spaces between specifications, states, controllers and plants as it is indicated in Figure 2.15.

The previous problem and others to be exposed in next sections will be treated as Quantified Constraints Satisfaction Problems. We will use the Modal Interval Analysis (MIA), developed by the SIGLA/X group **Gardeñes et al. (2001)** and Quantified Sets Inversion Algorithms **Herrero et al. (2005)**; **Herrero (2006)**, together with the property of flatness of the nonlinear systems to obtain the solution sets.

2.2.4 Phases of the robust controllers design procedure

The robust controllers design procedure is develop in two phases. The fisrt phase is realized in open-loop. The dynamics of the system are inverted in order to find a set of nominal feedforward. The inversion procedure is divided in three stages. In the first stage (see Figure 2.16), the robustness of a nominal controller under parametric uncertainty of the plant is determined such that some specifications are fulfilled.

In the second stage (see Figure 2.17), a set of controllers that satisfy some specifications under parametric uncertainty of the plant is determined.

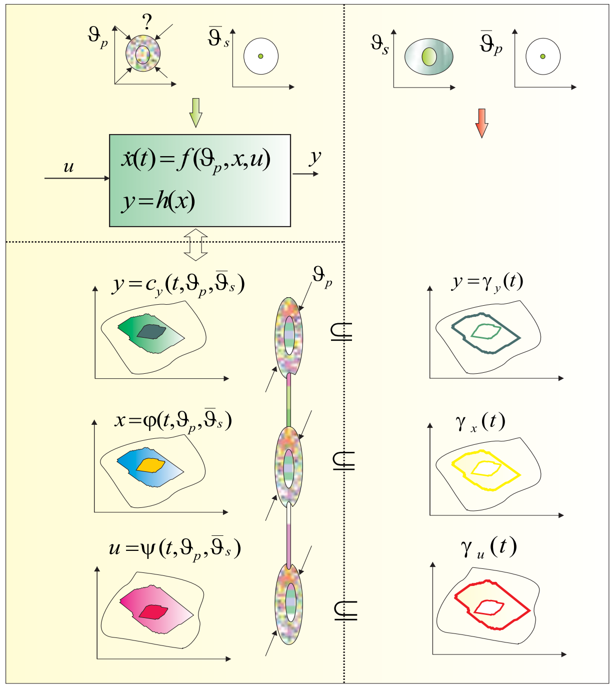


Fig. 2.14. Maximizing the space of plants to determine the robustness of a feed-forward controller $u = \psi(t, \overline{\vartheta}_s, \overline{\vartheta}_p)$.

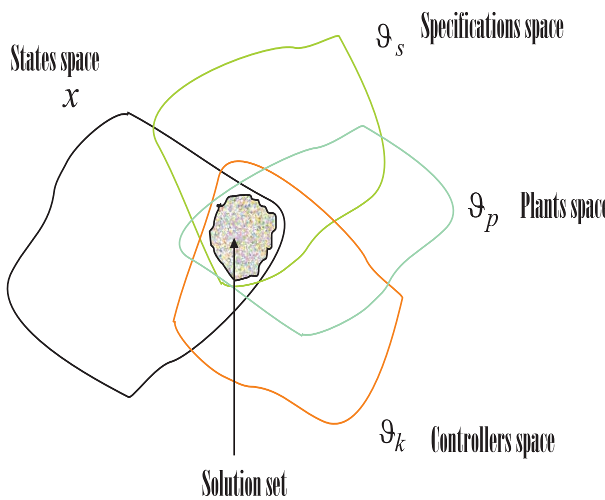


Fig. 2.15. Solution set of a Quantified Constraints Satisfaction Problem.

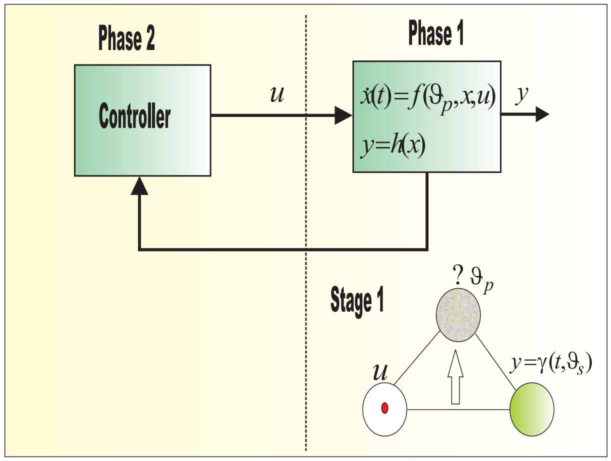


Fig. 2.16. First inversion stage.

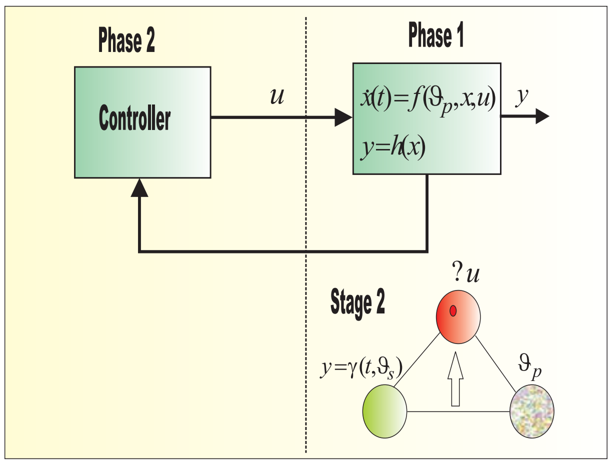


Fig. 2.17. Second inversion stage.

In the third stage (see Figure 2.18), a set of attainable specifications by some nominal controllers of the resulting family from previous stage is determined considering variations of the plant parameters.

The different stages of the inversion procedure are realized with the equivalent algebraic representation of the dynamic system as it is indicated in Figure 2.19. We can see that in each of the stages, the base of the triangles represents the data in which we began the inversion procedure and the question mark located on the upper edge represents the solution spaces that will be solved. In this thesis, the inversion procedure begins in the first stage and

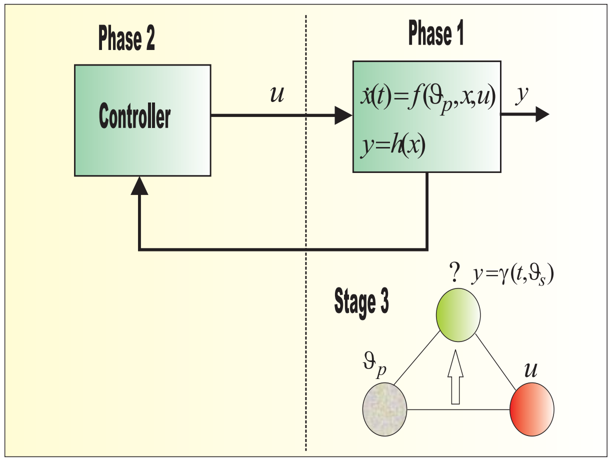


Fig. 2.18. Third inversion stage

the results of one stage are used in the following. For the general case, one can begin at any stage and return to it.

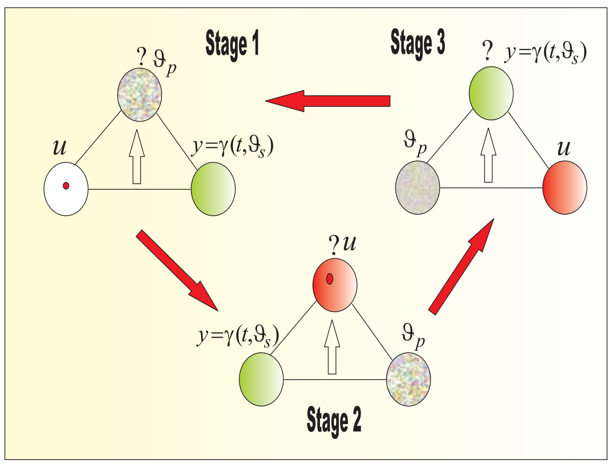


Fig. 2.19. Stages of the first phase of the inversion procedure.

The second phase is realized in closed-loop. In this phase a robust nominal feedforward obtained of the first phase is selected. The nominal feedforward is fixed in the control scheme of the controlled system to track the desired output, and an error feedback is added to correct errors due to noise or variations of the plant parameters.

The inversion procedure of the second phase consists of two stages. In the first stage (see Figure 2.20), a set of parameters of the feedback controller are determined such that some specifications are met under variations of the plant parameters.

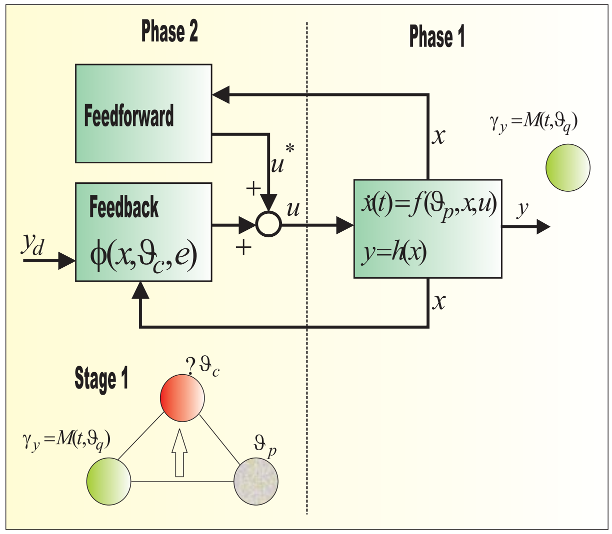


Fig. 2.20. First stage of the phase 2.

In the second stage (see Figure 2.21), the attainable specifications by some feedback controllers are determined under parametric uncertainty of the plant.

2.3 Main problems to be solved

The main problems to solve in the first phase are announced as follows:

1. Given hard and soft specifications $y = \gamma_y(t, \vartheta_s)$, one nominal plant $\bar{\vartheta}_p$ and a nominal controller $u = \psi(t, \bar{\vartheta}_s, \bar{\vartheta}_p) \Rightarrow \psi(t, \bar{\vartheta}_k)$, determine two regions of uncertainty of the plant (hard and soft plants) ϑ_p such that some specifications are fulfilled. A representation of this supposition is indicated in Figure 2.22.
   The pre-established nominal plant will stay within the plants space. Therefore, for the nominal controller, we are obtaining the allowable uncertainty in the plant so that some soft (hard) specifications are fulfilled. This first stage corresponds to study the robustness of a nominal controller such that the desired output is within the specification region. With specification intervals ϑ_s and one nominal plant $\bar{\vartheta}_p$ we computed

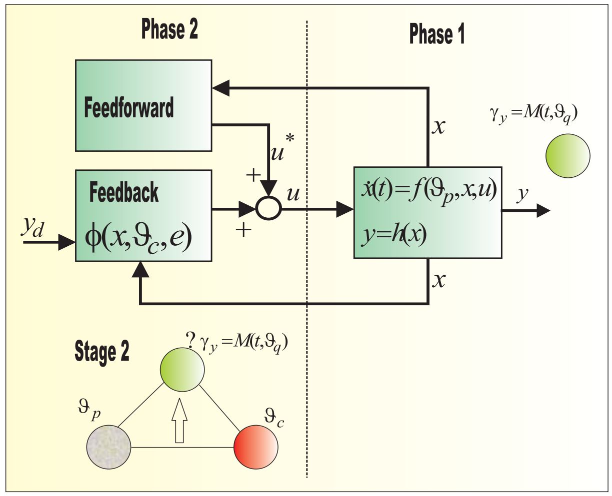


Fig. 2.21. Second stage of the phase 2.

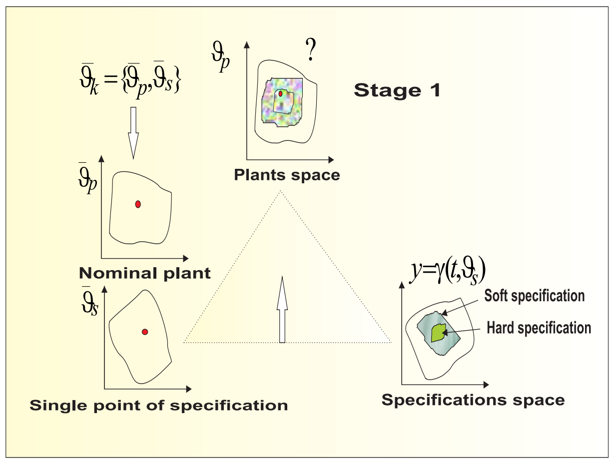


Fig. 2.22. Determination of addmisible uncertainty by a nominal controller $u = \psi(t, \bar{\vartheta}_s, \bar{\vartheta}_p)$, given hard and soft specifications $y = \gamma_y(t, \vartheta_s)$ and a nominal plant $\bar{\vartheta}_p$.

the desired region for the flat output $\gamma_y(t)$ and two bounding regions for the states $\gamma_x(t)$ and controllers $\gamma_u(t)$ as it is represented in Figure 2.23. The bounding regions of the states and controllers represent operation regions (search regions) where we may possibly find solutions. After previous step, with a point of specification $\bar{\vartheta}_s$, we will evaluate the possible set of plant parameters such that the process output c_y is within the

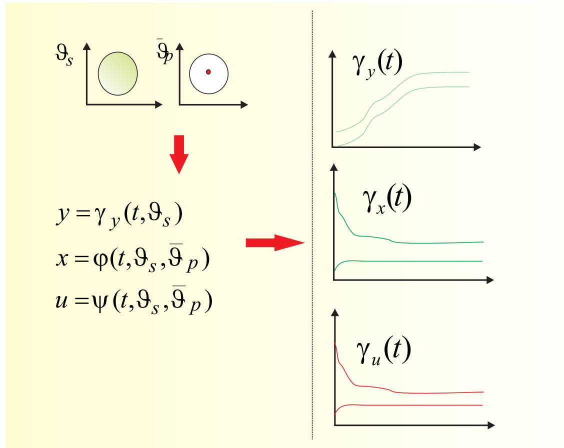


Fig. 2.23. Desired region for the flat output $\gamma_y(t)$ and two bounding regions for the states $\gamma_x(t)$ and controllers $\gamma_u(t)$.

specified region $c_y \subseteq \gamma_y$ and the constraints $\varphi \subseteq \gamma_x$ and $\psi \subseteq \gamma_u$ are met. This fact is indicated in Figure 2.24.

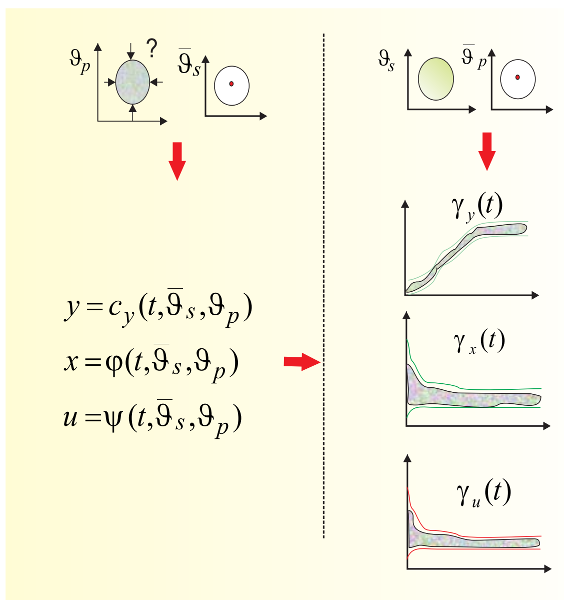


Fig. 2.24. Considerations to determine the robustness of a nominal controller $u = \psi(t, \bar{\vartheta}_s, \bar{\vartheta}_p) \Rightarrow u = \psi(t, \bar{\vartheta}_k)$.

In this work, bounding regions of the states and controllers are established as it was explained previously. But the approach can be applied for nonlinear flat systems with input and states constraints or in cases where the bounding regions can be given by the process control engineer. The different problems to resolve in this thesis are raised with this approach.

2. Given hard and soft specifications $\gamma_y(t, \vartheta_s)$ and a family of plants ϑ_p determine hard and soft controllers $u = \psi(t, \vartheta_s, \vartheta_p) \Rightarrow \psi(t, \vartheta_k)$ to guarantee the satisfaction of some hard and soft specifications for all the plants in the family. This fact is indicated in Figure 2.25.

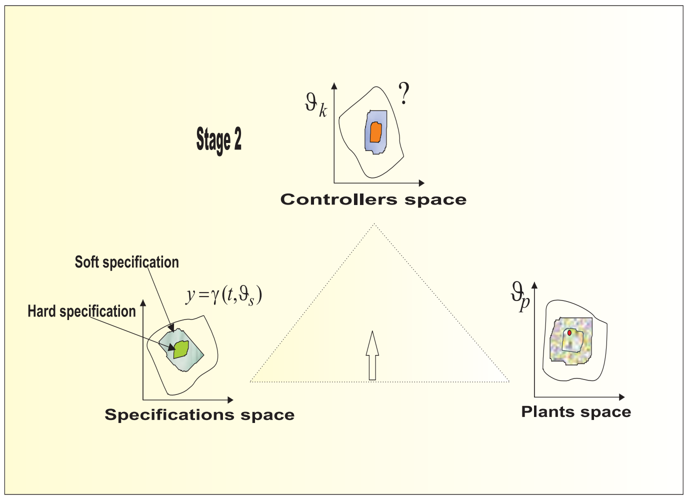


Fig. 2.25. Determination of hard and soft controllers $u = \psi(t, \vartheta_k)$ to guarantee the satisfaction of some hard and soft specifications $y = \gamma(t, \vartheta_s)$ for all the plants in the family ϑ_p.

3. Given hard and soft plants ϑ_p and hard and soft controllers $u = \psi(t, \vartheta_s, \vartheta_p) \Rightarrow \psi(t, \vartheta_k)$, determine attainable hard and soft specifications $y = \gamma_y(t, \vartheta_s)$ by some hard and soft controllers for all the hard and soft plants ϑ_p. In Figure 2.26 we indicate the hard and soft specifications that we want to find where the pre-established hard and soft specifications on the first problem will be within the specifications space.

To verify the results, we will fix a nominal controller to the input of the nonlinear system and we will verify if the outputs remain within the ranges of specifications under parametric uncertainty as it is indicated in Figure 2.27.

In the second phase, the problem to solve is as follows: Given some closed-loop hard and soft specifications $y = M(t, \theta_q)$ and a robust nominal controller $u^* = \psi(t, \bar{\vartheta}_k)$ selected from the first phase. Determine the set of hard and

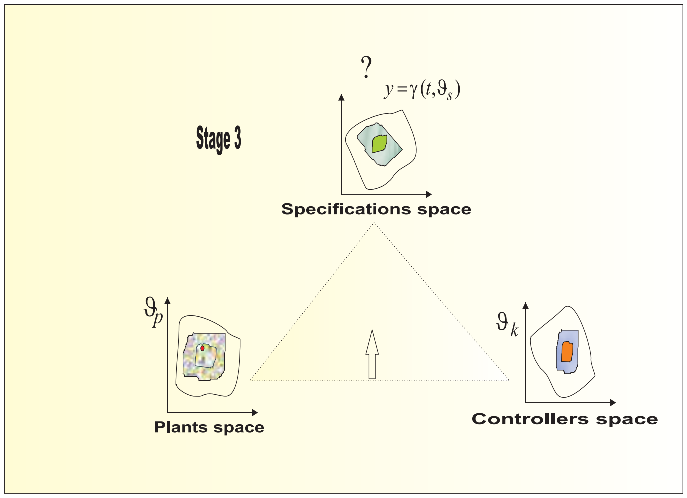


Fig. 2.26. Determination of hard and soft attainable specifications $y = \gamma_y(t, \vartheta_s)$ by some hard and soft controllers $u = \psi(t, \vartheta_k)$ for all the hard and soft plants ϑ_p.

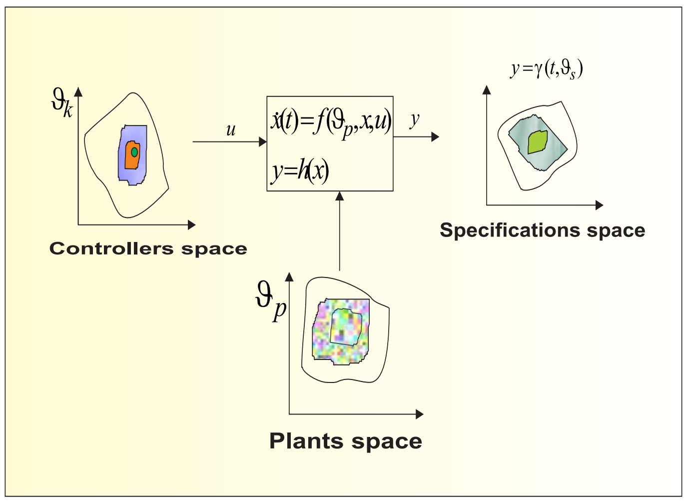


Fig. 2.27. Attainable specifications $y = \gamma_y(t, \vartheta_s)$ by some controllers $u = \psi(t, \vartheta_k)$ under parametric uncertainty of the plant ϑ_p.

soft parameters ϑ_c of the feedback controller $u = u^* + \phi(x, \vartheta_c, e)$ such that the process output is within desired specifications $y = M(t, \theta_q)$ (see Figure 2.28). In this same phase some robustness tests will be performed. That is, to verify that a feedback controller $u = u^* + \phi(x, \vartheta_c, e)$, will keep the output of the process within the region of hard specifications, under soft uncertainty of the plant.

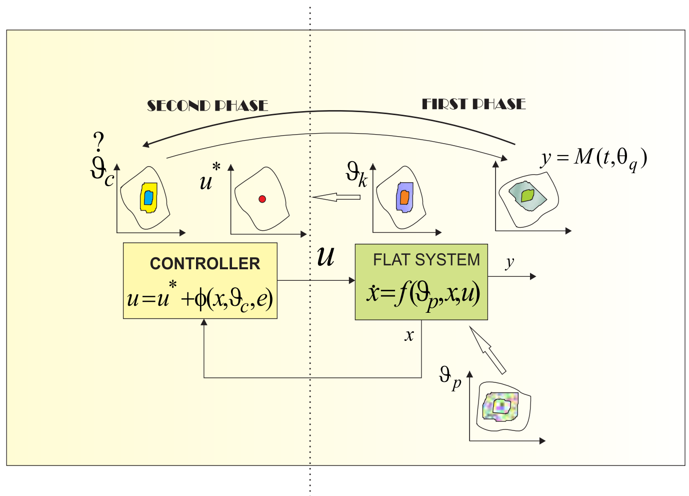


Fig. 2.28. Robust feedback controller.

2.4 Example applied to a first-order linear system

An example applied to a first-order linear system to explain the methodology followed in this thesis will be developed. Let us consider the following linear dynamic system:

$$\dot{x} = -ax + bu \tag{2.1}$$

a and b are plant parameters, x the state variable and u the control input. If we select the flat output $y = x$, then we can obtain the control input u and state x as follows:

$$\begin{array}{l} x = y \Rightarrow \varphi(t, y) \Rightarrow \varphi(t, \vartheta_s), x = c_y(t, \vartheta_s) \\ \dot{x} = -ax + bu \\ \dot{y} = -ay + bu \\ u = \frac{1}{b}[\dot{y} + ay] \Rightarrow \psi(t, \vartheta_p, y, \dot{y}) \Rightarrow \psi(t, \vartheta_p, \vartheta_s) \end{array} \tag{2.2}$$

We can see that the control input u (feedforward controller) is function of the plant parameters a, b, of the flat output y and of its derivative $\dot{y}$. The desired trajectory for the flat output can be planned with the following function:

$$y = y_o + (1 - y_o)(1 - e^{-t/\tau}) \tag{2.3}$$

y_o is a specification parameter for the flat output, τ is a specified time constant. The time t is considered as a set of points within the interval $0 \leq t \leq t_f$, being t_f the final time. Let us derive with respect to the time the equation (2.3) to obtain $\dot{y}$ because it is required in the expression of u from equation (2.2). Thus

$$\dot{y} = (1 - y_o)(\tfrac{1}{\tau})(e^{-t/\tau}) \tag{2.4}$$

if the equations (2.3) and (2.4) are replaced into the equation (2.2), the control input is expressed in function of plant parameters a, b, specification parameters y_o, τ and time t as follows

$$u = \tfrac{1}{b}[(1 - y_o)(\tfrac{1}{\tau})(e^{-t/\tau}) + a(y_o + (1 - y_o)(1 - e^{-t/\tau}))] \tag{2.5}$$

We will make some notations. $\vartheta_s = \{\mathbf{y}_o = [\underline{y}_o, \overline{y}_o], \tau\}$ contain the specification parameters. $\bar{\vartheta}_s = \{y_o = (\bar{y}_o - \underline{y}_o)/2, \tau\}$ is a point of specification, $\vartheta_p = \{\mathbf{a} = [\underline{a}, \overline{a}], \mathbf{b} = [\underline{b}, \overline{b}]\}$ are uncertain intervals of the plant. $\bar{\vartheta}_p = \{a, b\}$ is a nominal plant.

The regions for the flat output can be computed with ϑ_s and t from equation (2.3). The equation is denoted as:

$$\mathbf{y} = \mathbf{y}_o + (1 - \mathbf{y}_o)(1 - e^{-t/\tau}) \Rightarrow \gamma_y(t, \vartheta_s) \Rightarrow \gamma_y(t) \tag{2.6}$$

Now, let us specify a hard $\mathbf{y}_o = [0.1, 0.15]$ and soft $\mathbf{y}_o = [0.07, 0.2]$ specification, the time constant $\tau = 0.1$, and the interval of time $0 \leq t \leq 1$. With these values from equation (2.6) we computed two regions for the trajectory of the flat output $\gamma_y(t)$. The results are indicated in Figure 2.29.

In a similar way, the bounding regions for the control input can be computed with $\{\vartheta_s, \bar{\vartheta}_p\}$ and t from equation (2.5). The equation is denoted as:

$$\begin{aligned}\mathbf{u} = \tfrac{1}{b}[(1 - \mathbf{y}_o)(\tfrac{1}{\tau})(e^{-t/\tau}) + \\ a(\mathbf{y}_o + (1 - \mathbf{y}_o)(1 - e^{-t/\tau}))] \Rightarrow \gamma_u(t, \vartheta_s, \bar{\vartheta}_p) \Rightarrow \gamma_u(t)\end{aligned} \tag{2.7}$$

Let us propose two bounding regions for the control input as it is indicated in Figure 2.30.

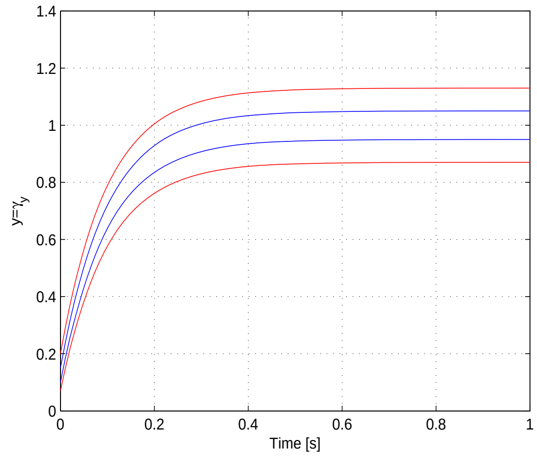


Fig. 2.29. Flat output regions computed from equation (2.6) with hard $\mathbf{y}_o = [0.1, 0.15]$ and soft $\mathbf{y}_o = [0.07, 0.2]$ specifications, $\tau = 0.1$ and $0 \leq t \leq 1$.

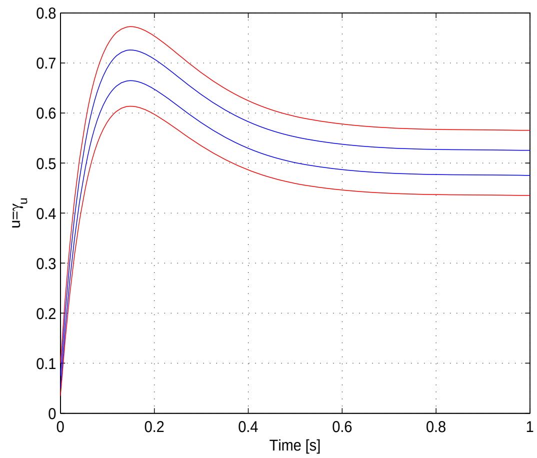


Fig. 2.30. Two bounding regions for the control input $\gamma_u(t)$ with hard $\mathbf{y}_o = [0.1, 0.15]$ and soft $\mathbf{y}_o = [0.07, 0.2]$ specification, $\tau = 0.1$, $0 \leq t \leq 1$ and nominal plant $a = 0.5, b = 1$.

Before studying the robustness of a feedforward controller $u = \psi(t, \bar{\vartheta}_s, \bar{\vartheta}_p)$, we will perform the following notation. A trajectory for the flat output can be computed with $\{\bar{\vartheta}_s\}$ and t from equation (2.3). In this case, the equation is expressed as:

$$y = y_o + (1 - y_o)(1 - e^{-t/\tau}) \Rightarrow \gamma_y(t, \bar{\vartheta}_s) \tag{2.8}$$

In a similar way, a family of feedforward controllers $u = \psi(t, \bar{\vartheta}_s, \vartheta_p)$ can be computed with $\{\bar{\vartheta}_s, \vartheta_p\}$ and t from equation (2.5). So, the notation corresponds to:

$$\begin{aligned}&\mathbf{u} = \tfrac{1}{\mathbf{b}}[(1 - y_o)(\tfrac{1}{\tau})(e^{-t/\tau}) + \\ &\mathbf{a}(y_o + (1 - y_o)(1 - e^{-t/\tau}))] \Rightarrow \psi_u(t, \bar{\vartheta}_s, \vartheta_p)\end{aligned} \tag{2.9}$$

Let us compute the maximum permissible uncertainty by a feedforward controller $\psi_u(t, \bar{\vartheta}_s, \bar{\vartheta}_p)$. The problem will be solved through an interval optimization approach. A representation for a general case is indicated in Figure 2.31. The approach consists in solving an optimization problem, maximizing the plants space ϑ_p and verifying that $c_y(t, \overline{\vartheta}_s) \subseteq \gamma_y(t) \wedge \varphi_x(t, \overline{\vartheta}_s, \vartheta_p) \subseteq \gamma_x(t) \wedge \psi_u(t, \overline{\vartheta}_s, \vartheta_p) \subseteq \gamma_u(t)$. So, if all constraints are met for a particular plant ϑ_p, the plant is stored in a solution set.

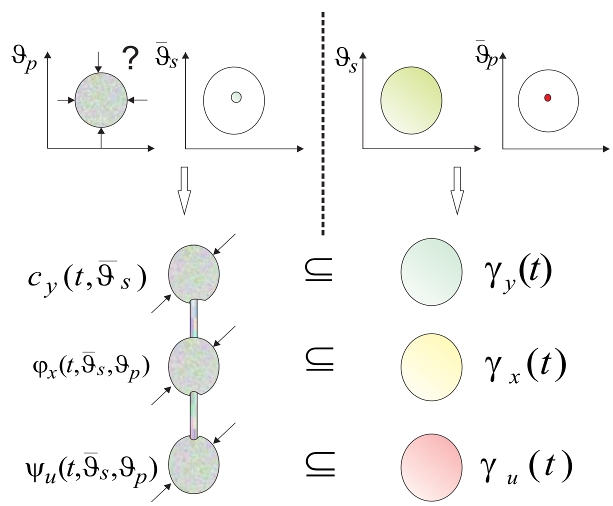


Fig. 2.31. Interval optimization approach to obtain the maximum permissible uncertainty ϑ_p by a feedforward controller $\psi_u(t, \bar{\vartheta}_s, \bar{\vartheta}_p)$.

The previous problem can be raised as a Quantified Constraints Satisfaction Problem as follows:

$$\begin{aligned}\Sigma_{\forall\exists} = \{&\vartheta_p \in R | \forall (t \in \boldsymbol{t}^{'}) \exists (\vartheta_s \in \boldsymbol{\vartheta}_s^{'}) \\ &(c_y(t, \bar{\vartheta}_s) \subseteq \gamma_y(t) \wedge \\ &\psi_u(t, \bar{\vartheta}_s, \vartheta_p) \subseteq \gamma_u(t))\}\end{aligned} \tag{2.10}$$

This equation is interpreted in the following way: Determine the maximum admissible uncertainty (ϑ_p) by a feedforward controller ($\psi_u(t, \bar{\vartheta}_p, \bar{\vartheta}_s)$) and for all t within the interval $t^{'}$ such that will ensure that some specifications $\exists(\vartheta_s, \boldsymbol{\vartheta}_s^{'})$ are met and that the constraints will be satisfied. This kind of expressions will be developed and explained with more details in next chapters.

Given hard $\mathbf{y}_o = [0.1, 0.15]$ and soft $\mathbf{y}_o = [0.07, 0.2]$ specifications, a nominal plant $a = 0.5$, $b = 1$, $\tau = 0.1$, $0 \leq t \leq 1$ and a point of specification $y_o = (0.15 - 0.1)/2 = 0.0250$ within the hard region, two regions of uncertainty of the plant (hard and soft plants) are depicted in Figure 2.32

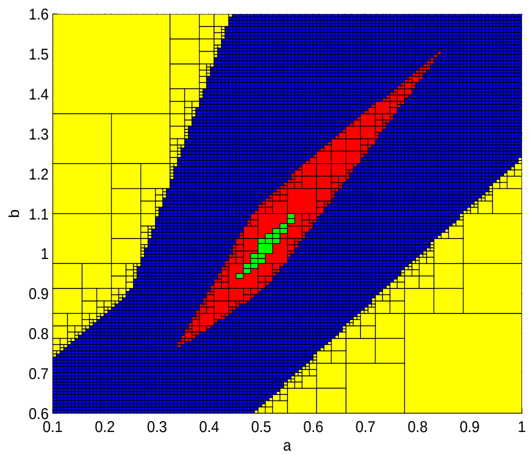


Fig. 2.32. The green and red boxes represent the solution set (hard and soft uncertainty regions). The yellow boxes are outside of the solution set and the blue boxes are undefined.

Let us select two boxes of parameters (hard and soft plants) from Figure 2.32. For instance the hard $\mathbf{a} = [0.5, 0.51], \mathbf{b} = [1, 1.01]$ and soft $\mathbf{a} = [0.5, 0.55], \mathbf{b} = [0.99, 1.1]$ plants. We can examine the equation (2.7) with previous parameters and a fixed specification $y_o = (0.15 - 0.1)/2 = 0.0250$ to determine the effect of the uncertain parameters in the computation of a family of feedforward controllers. The equation (2.7) is expressed as follows:

$$\mathbf{u} = \frac{1}{\mathbf{b}}[(1-y_o)(\frac{1}{\tau})(e^{-t/\tau})+ \mathbf{a}(y_o + (1-y_o)(1-e^{-t/\tau}))] \Rightarrow \psi_u(t, \bar{\vartheta}_s, \vartheta_p) \quad (2.11)$$

Two regions of hard and soft feedforward controllers are depicted in Figure 2.33.

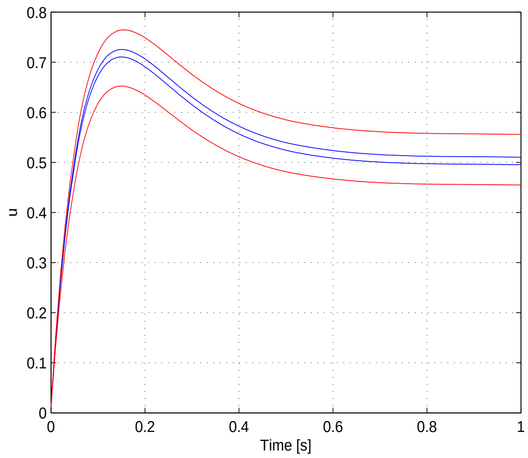


Fig. 2.33. Two regions of hard and soft feedforward controllers $\psi_u(t, \bar{\vartheta}_s, \vartheta_p)$ computed with hard $\mathbf{a} = [0.5, 0.51], \mathbf{b} = [1, 1.01]$ and soft $\mathbf{a} = [0.5, 0.55], \mathbf{b} = [0.99, 1.1]$ plants, a fixed specification $y_o = (0.15 - 0.1)/2 = 0.0250$, $\tau = 0.1$ and $0 \leq t \leq 1$.

In Figure 2.33, we can see that in stable state we obtain a family of hard and soft nominal feedforward controllers $\mathbf{u}^*$. The family of hard and soft nominal feedforward controllers can be also computed with equation (2.2). As in stable state $\dot{y} = 0$ and $y = \bar{y}$ then

$$\mathbf{u} = \mathbf{u}^* = \frac{1}{\mathbf{b}}[\mathbf{a}\bar{y}] \Rightarrow u^*(t, \vartheta_p, \bar{\vartheta}_s) \quad (2.12)$$

Thus, with a point of specification $y_o = 0.0250$, $t = 1$, $\tau = 0.1$, $\bar{y} = y_o + (1-y_o)(1-e^{-t/\tau}) = 1$ and hard $\mathbf{a} = [0.5, 0.51], \mathbf{b} = [1, 1.01]$ and soft $\mathbf{a} = [0.5, 0.55], \mathbf{b} = [0.99, 1.1]$ plants, from equation (2.12) we obtain the family of hard $\mathbf{u}^* = [0.49, 0.51]$ and soft $\mathbf{u}^* = [0.45, 0.55]$ nominal feedforward controllers. As we can see, the familiy of nominal feedforward controllers are within the ranges indicated in Figure 2.33.

Let us compute the attainable specifications by some feedforward controllers $\psi_u(t, \bar{\vartheta}_s, \bar{\vartheta}_p)$. The problem will be solved in a similar way as the previous case. A representation for a general case is indicated in Figure 2.34. The approach consists in solving an optimization problem maximizing the specifications space ϑ_s and verifying that $c_y(t, \vartheta_s) \subseteq \gamma_y(t) \wedge \varphi_x(t, \vartheta_s, \bar{\vartheta}_p) \subseteq \gamma_x(t)$

$\wedge\ \psi_u(t,\vartheta_s,\bar{\vartheta}_p) \subseteq \gamma_u(t)$. So, if all constraints are met for a particular specification ϑ_s, the specification is stored in a solution set. As ϑ_s will be varied, we will used the auxiliary specification parameter ϑ_{ss} to define the region of the flat output (as was defined previously).

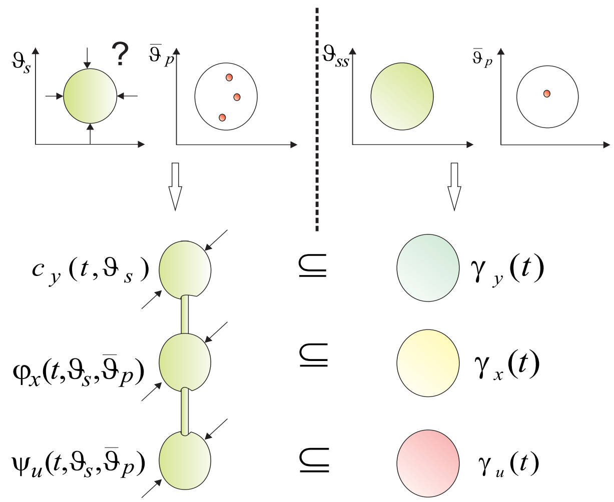


Fig. 2.34. Interval optimization approach to obtain the attainable specifications ϑ_s by some feedforward controllers $\psi_u(t,\bar{\vartheta}_s,\bar{\vartheta}_p)$ (maximization of the specifications space ϑ_s).

The previous problem can be raised as a Quantified Constraints Satisfaction Problem as follows:

$$\begin{aligned}\Sigma_{\forall\exists} = \{&\vartheta_s \in R | \forall(t \in \boldsymbol{t}^{'})\exists(\vartheta_p \in \boldsymbol{\vartheta}^{'}_p)\\ &(c_y(t,\vartheta_s) \subseteq \gamma_y(t) \wedge\\ &\psi_u(t,\vartheta_s,\bar{\vartheta}_p) \subseteq \gamma_u(t))\}\end{aligned} \tag{2.13}$$

This equation is interpreted in the following way: Determine the maximum attainable specifications (ϑ_s) by a feedforward controller ($\psi_u(t,\bar{\vartheta}_p,\bar{\vartheta}_s)$) for all t within the interval $\boldsymbol{t}^{'}$ such that constraints will be satisfied under parametric uncertainty in the plant (ϑ_p).

First, let us fix the values of ϑ_{ss} and $\bar{\vartheta}_p$. Thus, for the hard $\vartheta_{ss} = \{\mathbf{y}_o = [0.1, 0.15], \tau = 0.1\}$ and soft $\vartheta_{ss} = \{\mathbf{y}_o = [0.7, 0.2], \tau = 0.1\}$ specifications and nominal plant $\bar{\vartheta}_p = \{a = 0.5, b = 1\}$, we obtain $\gamma_y(t,\vartheta_{ss}) \Rightarrow \gamma_y(t)$, $\gamma_u(t,\vartheta_{ss},\overline{\vartheta}_p) \Rightarrow \gamma_u(t)$ for $0 \leq t \leq 1$.

Let us consider a hard $\psi_u(t,\bar{\vartheta}_p,\bar{\vartheta}_s)$ feedforward controller with the following parameters $\bar{\vartheta}_p = \{a = 0.55, b = 1.05\}, \bar{\vartheta}_s = \{y_o = 1, \tau = 0.1\}$ and two soft $\psi_u(t,\bar{\vartheta}_p,\bar{\vartheta}_s)$ feedforward controllers with the following values

$\bar{\vartheta}_p = \{a = 0.7, b = 1.3\}, \bar{\vartheta}_p = \{a = 0.4, b = 0.84\}$ and $\bar{\vartheta}_s = \{y_o = 1, \tau = 0.1\}$. From plants space indicated in Figure 2.35, we can see the location of selected plants. The hard feedforward controller satisfies the range of hard specifications $\mathbf{y}_o = [0.1, 0.18]$. The first soft feedforward controller satisfies the range of soft specifications $\mathbf{y}_o = [0.1, 0.1430]$ and the second all the range of soft specifications $\mathbf{y}_o = [0.07, 0.2]$.

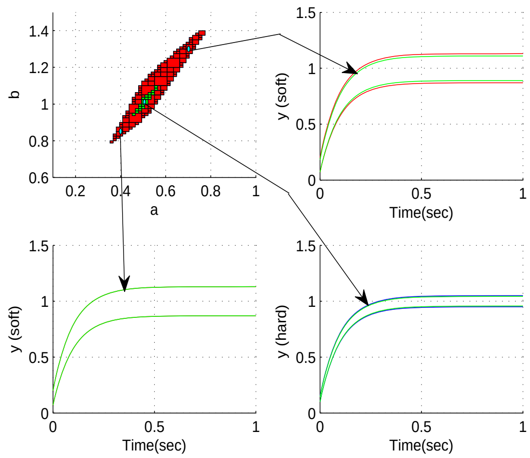


Fig. 2.35. Attainable specifications by some feedforward controllers. The hard $\psi_u(t, \bar{\vartheta}_p, \bar{\vartheta}_s)$ feedforward controller is selected with parameters $\bar{\vartheta}_p = \{a = 0.55, b = 1.05\}, \bar{\vartheta}_s = \{y_o = 1, \tau = 0.1\}$ and the two soft $\psi_u(t, \bar{\vartheta}_p, \bar{\vartheta}_s)$ feedforward controllers with values $\bar{\vartheta}_p = \{a = 0.7, b = 1.3\}, \bar{\vartheta}_p = \{a = 0.4, b = 0.84\}$ and $\bar{\vartheta}_s = \{y_o = 1, \tau = 0.1\}$.

Let us realize some robustness tests to the family of hard $\mathbf{u}^* = [0.49, 0.51]$ nominal feedforward in open-loop to verify if the output y for the system $\dot{x} = f(x, \vartheta_p, u^*)$ remains within the hard region $y = [0.94, 1.04]$, under soft $\vartheta_p = \{\mathbf{a} = [0.5, 0.55], \mathbf{b} = [0.99, 1.1]\}$ parametric uncertainty and initial conditions $x(0) \in [0.86, 1.12]$ for all time future t. The system in open-loop will be controlled with the nominal feedforwards $\mathbf{u}^* = [0.49, 0.51]$ introducing at certain time instant variations of the process parameters. According to the results presented in Figure 2.36, the solution of the system $\dot{x} = f(x, \vartheta_p, u^*)$ belongs to the region $[0.94, 1.04]$ for all future time t. So, the region $[0.94, 1.04]$ is an invariant region with respect to the dynamical system $\dot{x} = f(x, \vartheta_p, u^*)$ **Marquez (2003)**.

All the previous steps correspond to the first phase of design. In the second phase, let us design a state feedback control law and let us determine its hard and soft parameters such that the output is within the region of hard and soft specifications under hard and soft uncertainty of the plant.

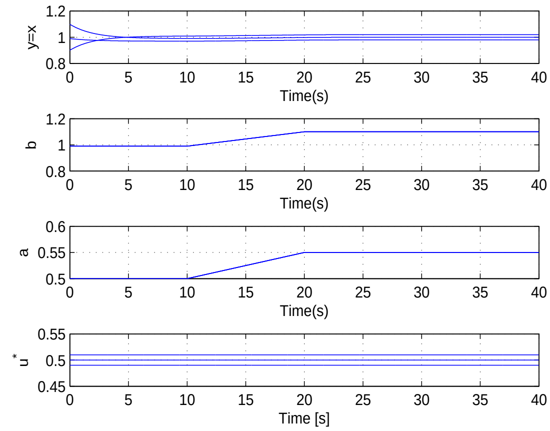


Fig. 2.36. Robustness test to the family of hard $\mathbf{u}^* = [0.49, 0.51]$ nominal feedforward.

From equation (2.2) the feedforward controller was obtained as follows:

$$u = \frac{1}{b}[\dot{y} + ay] \tag{2.14}$$

let us select the new control input $v = \dot{y}$ and express the previous equation in terms of the state

$$u = \frac{1}{b}[v + ax] \tag{2.15}$$

The closed-loop dynamics of the system can be stablished with the equation $\dot{y} + k(y - \bar{y}) = 0$. Which can be made asymtotically stable by a suitable choice of the design parameter k. Let us search the set of hard and soft parameters to k, such that the desired specifications are met. Closed-loop dynamics can be expressed in terms of the state variables as follows:

$$\dot{y} = -k(x - \bar{x}) \tag{2.16}$$

therefore, the state feedback control law is expressed of the following form

$$u = \frac{1}{b}[-k(x - \bar{x}) + ax] \tag{2.17}$$

The state feedback controller can be expressed in function of the nominal feedforward (u^*) determined with equation (2.12) of the following way

$$u = u^* + \frac{1}{b}[-kx + \bar{x}(k-a) + ax] \quad (2.18)$$

now, let us define a desired region (closed-loop reference model) in time with the following interval function

$$M(t, \theta_q) = \boldsymbol{\theta}_{q1}(1 - \boldsymbol{\theta}_{q2} exp(-\boldsymbol{\theta}_{q3} t)) \quad (2.19)$$

we desired that the output of the feedback system is within the reference model $M(t, \theta_q)$. Where $\boldsymbol{\theta}_{q1}$ is the interval of the hard and soft speficifications, $\boldsymbol{\theta}_{q2}$ and $\boldsymbol{\theta}_{q3}$ are intervals to fix the response speed in time of the function $M(t, \theta_q)$, t is a time interval. If we define the hard $\boldsymbol{\theta}_{q1} = [0.94, 1.04]$ and soft $\boldsymbol{\theta}_{q1} = [0.86, 1.12]$ specification and $\boldsymbol{\theta}_{q2} = [0.6, 1]$, $\boldsymbol{\theta}_{q3} = [0.3, 1]$ and $t := \{t \in R | 0 \leq t \leq 40\}$ the bound interval of the hard and soft interval function $M(t, \theta_q)$ is depicted in Figure 2.37.

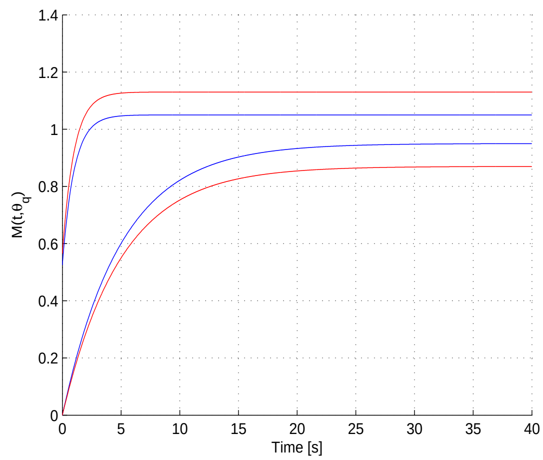


Fig. 2.37. Reference model $M(t, \theta_q) = \boldsymbol{\theta}_{q1}(1 - \boldsymbol{\theta}_{q2} exp(-\boldsymbol{\theta}_{q3} t))$ with hard $\boldsymbol{\theta}_{q1} = [0.94, 1.04]$ and soft $\boldsymbol{\theta}_{q1} = [0.86, 1.12]$ specification and intervals $\boldsymbol{\theta}_{q2} = [0.6, 1]$, $\boldsymbol{\theta}_{q3} = [0.3, 1]$.

In Figure (2.38) we make a geometric representation of the interval optimization approach considering the feedback dynamic system. We applied a nominal feedforward u^* determined previously in the state feedback law (2.18) and we found intervals $\vartheta_c = \{[\underline{k}, \overline{k}]\}$ of the feedback controller such that the trajectory of the state variable $x = \mu_c l(t, \vartheta_p, \vartheta_c)$ is within the limits of the reference model $\gamma_y = M(t, \theta_q)$.

A more formal expression can be expressed as follows:

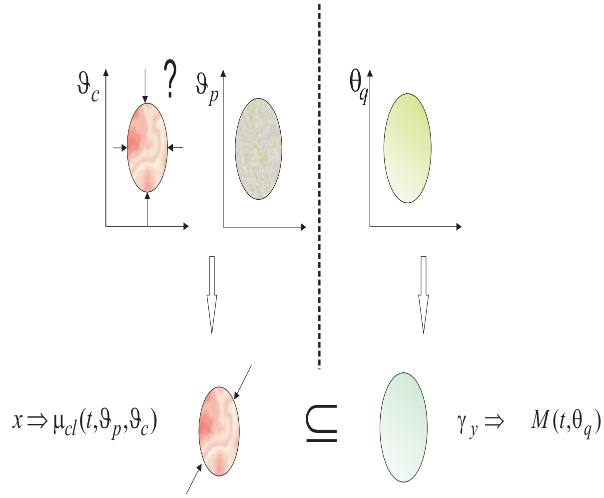


Fig. 2.38. Representation of the interval optimization approach considering the feedback dynamic system.

$$\begin{aligned} \Sigma_{\forall\exists} = \{&\vartheta_c | \forall(t \in \boldsymbol{t}^{'})\forall(\vartheta_p \in \boldsymbol{\vartheta}_p^{'})\exists(\theta_q \in \boldsymbol{\theta}_q^{'}) \\ &(x(t, \vartheta_p, \vartheta_c) \subseteq M(t, \theta_q)\} \end{aligned} \tag{2.20}$$

When solving the previous problem, we obtained the hard $\mathbf{k} = [0.22, 1.43]$ and soft $\mathbf{k} = [0.1, 1.87]$ intervals. Let us select three parameters within the hard interval $k = 1.4$, $k = 0.75$ and $k = 0.25$ and one parameter $k = 1.9$ that is outside of soft interval. In Figure 2.39 we can see that the state feedback controller with parameters within the hard interval satisfies the specifications under soft parametric uncertainty in the plant, and when we proved the parameter that is outside of soft interval, the state feedback controller did not meet the hard and soft specifications at some instants of time t.

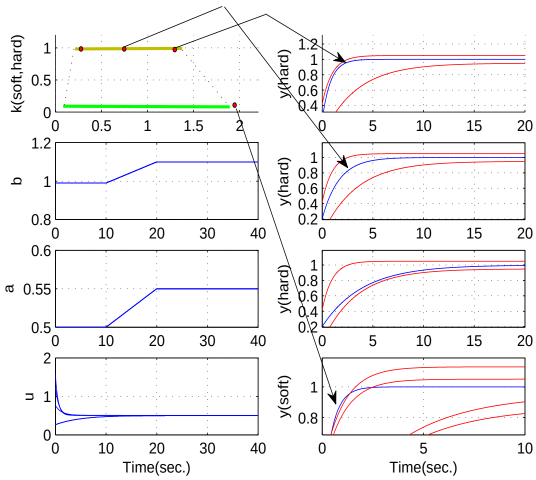


Fig. 2.39. Robustness test to the state feedback controller with uncertainties.

3 Preliminaries

The main objective of robust control is to develop feedback control laws that are robust against plant model uncertainties and changes in dynamic conditions. A system is robustly stable when the closed-loop is stable for any chosen plant within the specified uncertainty set. This system has robust performance if the closed-loop system satisfies performance specifications for any plant model within the specified uncertainty description. The most used robust control approaches are: H_∞, theory of control based on Lyapunov and Sliding Mode Control. A new approach to robust possibilistic control based on flatness, using tools of Modal Interval Analysis, is developed in this thesis. In this Chapter different tools are presented such as: the flatness theory under parametric uncertainty, semantics and properties from Modal Interval Analysis to find solution sets, and technical issues related to Quantified Constraints Satisfaction Problems.

3.1 Introduction

In this thesis, we will focus on the robust control of a subclass of nonlinear systems known as nonlinear flat systems. Differential flatness is a property of nonlinear systems which allows us to perform trajectories planning of a system. The planning is done using some functions that define the desired output trajectories. With mathematical manipulation one finds the state variables and control inputs in terms of the outputs and a finite number of its derivatives. Differential flatness becomes a subject of study when the uncertainty is considered in the process. Specifically the outputs are defined as operation regions instead of unique trajectories. Since we deal with intervals, all the obtained algebraic equations as a result of the flat systems theory application are analyzed with tools of the Modal Interval Analysis. With the new approach of robust possibilistic control the inverse dynamics problem is transformed into a sets intersection problem between specifications, states, controllers and plants.

3.2 Differential flatness

Differential flatness has been presented in both the differential algebraic setting **Fliess et al. (1995b)** and the differential geometric setting **Fliess et al. (1999)**. In both cases the motion planning issues have been addressed and flatness has been shown to be a property which is related to the trajectories of a system. A system of ordinary differential equations (3.1)

$$\dot{x}(t) = f(x(t), u(t)) \tag{3.1}$$

It is said to be differentially flat, if there exist a set variables (denoted as flat outputs) equal to the number of inputs such that all states and inputs can be determined from these outputs and its derivatives. More precisely, if the system (3.1) has states $x \in R^n$, and inputs $u \in R^m$ the system is flat if we can find outputs $y \in R^m$ of the form

$$y = h(x, u, \dot{u}, ..., u^{(p)}) \tag{3.2}$$

such that

$$\begin{aligned} x &= \varphi(y, \dot{y}, ..., y^{(k)}) \\ u &= \psi(y, \dot{y}, ..., y^{(k+1)}) \end{aligned} \tag{3.3}$$

where $f(x, u) \in R^n$ is a smooth vector field, h, φ and ψ are smooth functions in open sets of $R^{n+m(p+1)}$,$R^{m(k+1)}$ and $R^{m(k+2)}$, respectively. Also, p and k are integer numbers. From equation (3.3) the state trajectory and the corresponding open-loop control can be obtained directly from a simple parametrization of the flat output. This parametrization can be performed in order to fulfill a given control objective and/or take into account some physical constraints on the system.

Any (full-state) feedback linearizable system is differentially flat by choosing the flat output as the feedback linearizing output. Indeed, differential flatness can be showed to be equivalent to dynamic feedback using a class of invertible dynamic feedbacks **Charlet et al. (1989)**; **Fliess et al. (1995a)**; **Fliess et al. (1995c)**; **Martin (1993)**; **Martin et al. (2000).** Hence, the class of systems which is differentially flat is essentially the same as dynamically feedback linearizable systems.

To illustrate this point of view, considering the feedback linearization problem for a single-input single-output nonlinear control system **Isidori (1985)**; **Nijmeijer et al. (1990)**

$$\begin{aligned} \dot{x}(t) &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{3.4}$$

Suppose that the output y has relative degree n, so that the system is full-state linearizable. If we derive the output we get:

$$\dot{y} = \dot{h}(x) = \frac{dh}{dx}\frac{dx}{dt} \tag{3.5}$$

as $\dot{x} = f(x) + g(x)u$, then

$$\dot{y} = \frac{dh}{dx}f(x) + \frac{dh}{dx}g(x)u \tag{3.6}$$

The variation of the output $h(x)$ with respect to vector field $f(x)$ can be indicated in terms of Lie's derivatives as $L_f h(x) = \frac{dh(x)}{dx}f(x)$. Similarly, the variation of the output $h(x)$ with respect to the vector field $g(x)$ is represented as $L_g h(x)$. So, we obtain the following expression to $\dot{y}$:

$$\dot{y} = L_f h(x) + L_g h(x)u \tag{3.7}$$

as the output has relative degree n, and if the order of the system is greater than one, then the control will not appear even in the expression of $\dot{y}$. Therefore,

$$\dot{y} = L_f h(x) \tag{3.8}$$

back again, we derive the output to obtain:

$$\begin{aligned} \ddot{y} &= \frac{d}{dx}[\dot{y}]\frac{dx}{dt} = \frac{d}{dx}[L_f h(x)](f(x) + g(x)u) \\ \ddot{y} &= \frac{d}{dx}[L_f h(x)]f(x) + \frac{d}{dx}[L_f h(x)]g(x)u \end{aligned} \tag{3.9}$$

$\frac{d}{dx}[\dot{y}]f(x)$ is the variation of the first derivative of output with respect to $f(x)$ in terms of Lie derivatives is equal to $L_f\dot{y}$. As $\dot{y} = L_f h(x)$, then $L_f\dot{y} = L_f L_f h(x) = L_f^2 h(x)$ so

$$\ddot{y} = L_f^2 h(x) + L_g L_f h(x)u \tag{3.10}$$

on similar way, if the control input do not appears in $\ddot{y}$, we continue to derive the output until appears the control signal. In general, in nonlinear systems with relative degree n the control signal appears on the n-th derivative of the output. So, the general expression for the n-th derivative is as follows:

$$y^{(n)} = L_f^n h(x) + L_g L_f^{n-1} h(x)u \tag{3.11}$$

if the new control input is defined as $v = y^{(n)}$, the control u obtained from the equation 3.11 is as follows:

$$u = \frac{-L_f^n h(x)}{L_g L_f^{n-1} h(x)} + \frac{1}{L_g L_f^{n-1} h(x)} v \tag{3.12}$$

by replacing u from the equation (3.12) to the equation (3.11) we verify that satisfies $y^{(n)} = v$. To verify that the dynamics of the system is transformed to a linear and controllable, we set a new coordinate z in terms of the output and its derivatives such as:

$$\begin{aligned}
z_1 &= y = h(x) \\
z_2 &= \dot{y} = L_f h(x) \\
z_3 &= \ddot{y} = L_f^2 h(x) \\
&. \\
&. \\
z_n &= y^{(n-1)} = L_f^{n-1} h(x)
\end{aligned} \tag{3.13}$$

the transformed system is as follows:

$$\begin{aligned}
\dot{z}_1 &= z_2 \\
\dot{z}_2 &= z_3 \\
\dot{z}_3 &= z_4 \\
&. \\
&. \\
\dot{z}_n &= v
\end{aligned} \tag{3.14}$$

as the new coordinate z is function of the state x, we can denote it as $z = \phi(x)$ and $x = \phi^{-1}(z)$. That is, we can get x from z and vice versa. A representation is depicted in Figure 3.1.

If $\alpha(x) = \frac{-L_f^n h(x)}{L_g L_f^{n-1} h(x)}$ and $\beta(x) = \frac{1}{L_g L_f^{n-1} h(x)}$ then the control signal from equation (3.12) can be expressed as:

$$u = \alpha(x) + \beta(x) v \tag{3.15}$$

In figure 3.2 the control scheme based on state feedback linearization is depicted.

There are different ways to set the control input v. We will describe some of them given that for controllers design based on differential flatness, we can set the control input v on similar way.

A stabilizing feedback controller can be obtained by setting the control input $v = y^{(n)}$ with a closed-loop characteristic polynomial as follows:

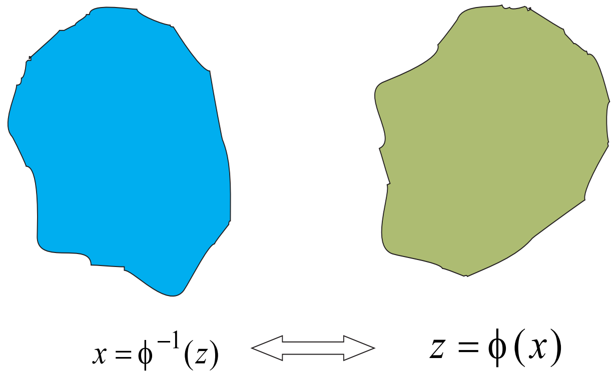


Fig. 3.1. Coordinates $z = \phi(x)$ and $x = \phi^{-1}(z)$.

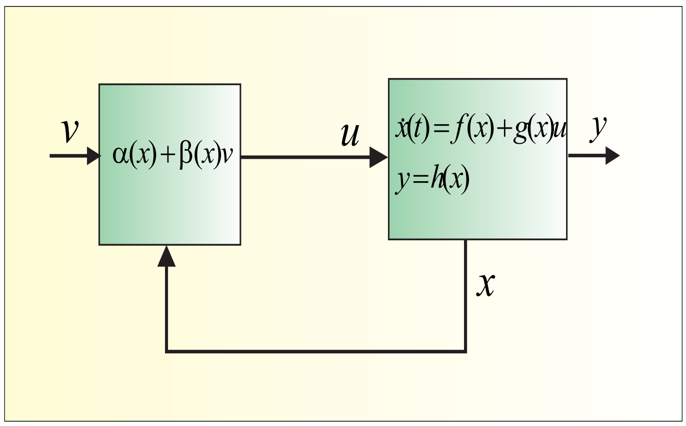


Fig. 3.2. State feedback linearization.

$$
\begin{aligned}
&y^{(n)} + k_{n-1}y^{(n-1)} + k_{n-2}y^{(n-2)} + .. + k_0 y = 0 \\
&v = y^{(n)} = -k_{n-1}y^{(n-1)} - k_{n-2}y^{(n-2)} - .. - k_0 y
\end{aligned}
\tag{3.16}
$$

where the set of design parameters $\{k_{n-1}, k_{n-2}, .., k_0\}$ are selected such that the close-loop characteristic polynomial has its roots appropriately placed on the left half-plane of the complex plane s. From equation (3.13) we saw that $y^{(n-1)} = L_f^{(n-1)}h(x)$, $y^{(n-2)} = L_f^{(n-2)}h(x)$ and $y = h(x)$. This means that output and its derivatives can be expressed as a transformation function of the states. So, the equation (3.16) will be represented as $\phi(x, \vartheta_c)$ where $\vartheta_c = \{-k_{n-1}, -k_{n-2}, .., -k_0\}$.

$$
v = \underbrace{-k_{n-1}L_f^{(n-1)}h(x) - k_{n-2}L_f^{(n-2)}h(x) - .. - k_0 y}_{\phi(x,\vartheta_c)}
\tag{3.17}
$$

replacing $v = \phi(x, \vartheta_c)$ from equation (3.17) in equation (3.15), the control input u is expressed as follows:

$$u = \alpha(x) + \beta(x)\phi(x, \vartheta_c) \tag{3.18}$$

the control scheme based on feedback linearization to drive states to zero is indicated in Figure 3.3.

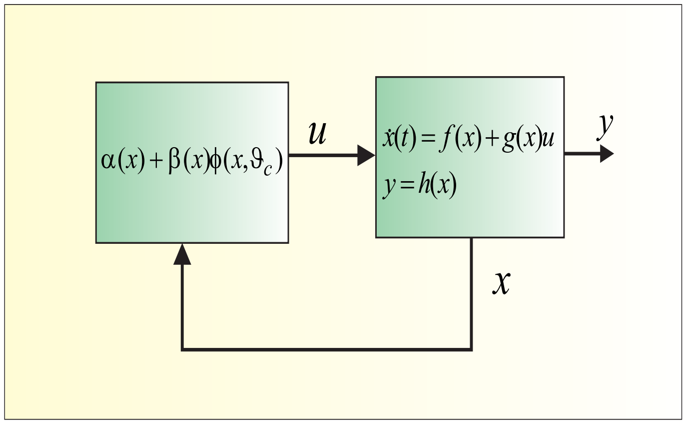


Fig. 3.3. Control scheme based on feedback linearization to drive states to zero.

If we desired drive the system output to a set point y_d then v is assigned with the same equation (3.17) plus the additional term $k_0 y_d$ as follows.

$$v = \underbrace{-k_{n-1}L_f^{(n-1)}h(x) - k_{n-2}L_f^{(n-2)}h(x) - .. + k_0(y_d - y)}_{\phi(x,\vartheta_c,e)} \tag{3.19}$$

being the error $e = (y_d - y)$. In the control scheme depicted in Figure 3.4 the error term is within the function $\phi(x, \vartheta_c, e)$.

Also we can use an integral control or other kind of linear controller to maintain the output at a non-zero set point despite unmeasured disturbances and parametric variations of the plant. In this case, the control input v can be assigned as follows:

$$\begin{aligned} v = {} & -k_n L_f^{(n-1)}h(x) - k_{n-1}L_f^{(n-2)}h(x) - .. + k_1(y_d - y) + \\ & k_0 \int_0^t (y_d - y)d\tau \Rightarrow \phi(x, \vartheta_c, e) \end{aligned} \tag{3.20}$$

where k_0 is an additional tunning parameter associated with the integral term. The control scheme based on state feedback linearization with linear controller is depicted in Figure 3.5.

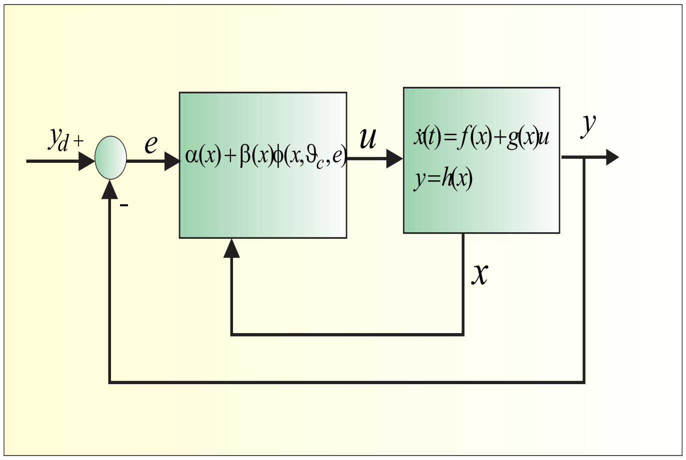


Fig. 3.4. Control scheme based on feedback linearization to drive the output to a set point y_d.

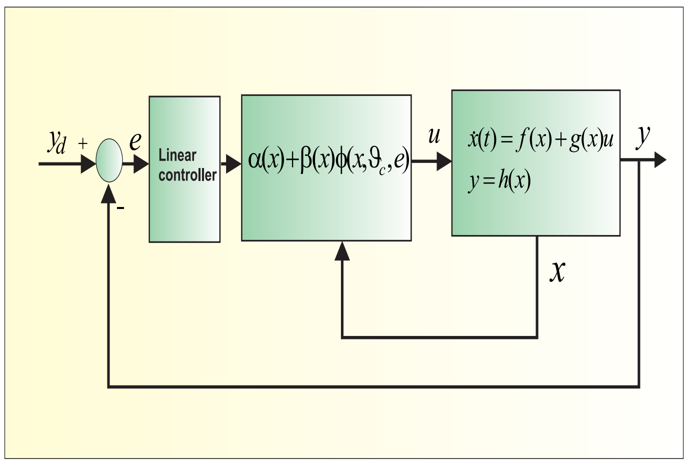


Fig. 3.5. Control scheme based on feedback linearization with linear controller to drive ouput to a set point y_d.

The control input v can be assigned also in error function as it is indicated in equation (3.21)

$$v = y_d^{(n)} - k_n(y^{(n-1)} - y_d^{(n-1)}) - k_{(n-1)}(y^{(n-2)} - y_d^{(n-2)}) .. - k_1(y - y_d) \tag{3.21}$$

Replacing equation (3.21) inside equation (3.15) the controller u will become:

$$u = \underbrace{\alpha(x) + \beta(x)y_d^{(n)}}_{feedforward} + \\ \underbrace{\beta(x)(-k_n(y^{(n-1)} - y_d^{(n-1)}) - k_{(n-1)}(y^{(n-2)} - y_d^{(n-2)}).. - k_1(y - y_d))}_{feedback} \tag{3.22}$$

As $y^{(n-1)} = L_f^{(n-1)}h(x)$ and $y^{(n-2)} = L_f^{(n-2)}h(x)$ the previous equation can be expressed as follows:

$$u = \underbrace{\alpha(x) + \beta(x)y_d^{(n)}}_{u^*} + \\ \underbrace{\beta(x)(-k_n(L_f^{(n-1)}h(x) - y_d^{(n-1)}) - k_{n-1}(L_f^{(n-2)}h(x) - y_d^{(n-2)}).. - k_1(y - y_d))}_{\phi(x,\vartheta_c,e)} \tag{3.23}$$

The control law in equation (3.23) can be viewed as a two-degree-of-freedom controller. The feedforward terms are the inputs that are required to track the trajectory y_d; the feedback terms are used in order to make that the tracking error decays exponentially to zero under system uncertainty.

The equation (3.23) can be manipulated to express the control input u only in function of the state variables, controller parameters and error as follows:

$$u = \beta(x)\,[\underbrace{(y_d^{(n)} + k_n y_d^{(n-1)} + k_{n-1}y_d^{(n-2)} + .. + k_1 y_d)}_{v} + \\ \underbrace{(-L_f^{(n)}h(x) - k_n L_f^{(n-1)}h(x) - k_{n-1}L_f^{(n-2)}h(x) - .. - k_1 y)]}_{\phi(x,\vartheta_c,e)} \tag{3.24}$$

In this case v is fixed as follows:

$$v = y_d^{(n)} + k_n y_d^{(n-1)} + k_{n-1}y_d^{(n-2)} + .. + k_1 y_d \tag{3.25}$$

If we desire to drive the system output to a set point y_d, we can set $v = k_0 \int_0^t (y_d - y)d\tau$. So, equation 3.24 is expressed as it is indicated in equation (3.26)

$$u = \beta(x)\,[\underbrace{(k_0 \int_0^t (y_d - y)d\tau) +}_{v} \underbrace{(-L_f^{(n)}h(x) - k_n L_f^{(n-1)}h(x) - k_{n-1}L_f^{(n-2)}h(x) - .. - k_1 y)]}_{\phi(x,\vartheta_c,e)} \tag{3.26}$$

An example can be seen in appendix.

A differentially flat system is represented in terms of the flat output and its derivatives as it was indicated in equation (3.3). The feedback controller based on differential flatness to track a set point y_d can be expressed in the following form

$$u = \psi(\underbrace{y, \dot{y}, ..., y^{(n-2)}, y^{(n-1)}}_{u^*}, \underbrace{y^{(n)}}_{v}) \tag{3.27}$$

where $v = y^n$ is the new control input and $u^* = \psi(y, \dot{y}, ..., y^{(n-2)}, y^{(n-1)})$ the nominal feedforward. In this thesis work, $v = \phi(x, \vartheta_c, e)$ is assigned as it is indicated in equation (3.19). Therefore, the feedback controller (3.27) can be expressed as:

$$u = u^* + \phi(x, \vartheta_c, e) \tag{3.28}$$

being $\vartheta_c = \{-k_{n-1}, -k_{n-2}, .., -k_0\}$. The feedback control scheme based on differenial flatness is depicted in Figure 3.6.

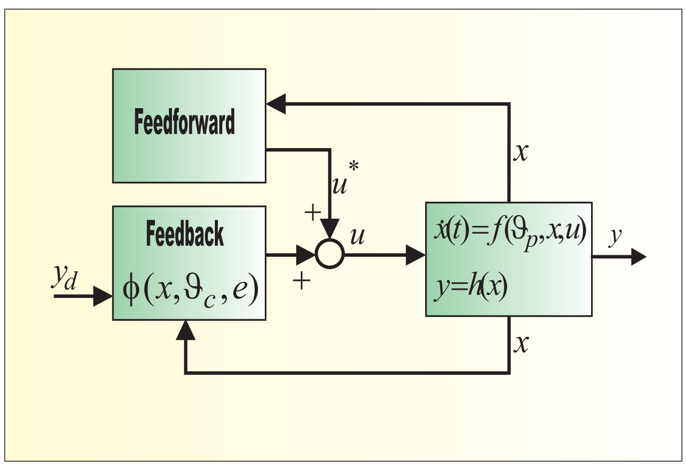


Fig. 3.6. Control scheme based on differenial flatness.

The nominal feedforward u^* can be expressed in terms of the states $u^* = \phi(x)$ so:

$$u = \psi(\underbrace{y, \dot{y}, ..., y^{(n-2)}, y^{(n-1)}}_{u^* = \phi(x)}, \underbrace{y^{(n)}}_{v=\phi(x, \vartheta_c, e)}) \tag{3.29}$$

As we may see, the controller based on differential flatness (3.29) uses the current state x and the desired output y_d, like the controller based on feedback linearization of equation (3.23). The difference is that the feedback linearizing controller inverts the coupling function $\beta(x)$. In **Van Nieuwstadt and Murray (1998)** we may observe more details. If $\beta(x)$ contains singularity points the controller will not be defined in those points; this leads to numerical problems even more if the uncertainty in the system is considerable. In contrast, the controller in equation (3.28) uses a scheduled gain that can be assigned by placing properly the closed-loop roots.

Flatness is a property of a system and does not imply that one intends to transform the system, via dynamic feedback and appropriate changes of coordinates, to a single linear system **Martin et al. (1997)**. Indeed, the power of flatness is precisely that this one does not convert nonlinear systems into linear ones. When a system is flat it is an indication that the nonlinear structure of the system is well characterized and one can exploit that structure in designing control algorithms for motion planning, trajectory generation, and stabilization.

Differential flatness can be seen as an equivalence problem between a set of undetermined ordinary differential equations and a set of linear ones **Chelouah (1997)**. As a matter of fact, relations (3.3) provide respectively the change of coordinates and the feedback control that transform (3.1) into a trivial form (i.e., the Brunovsky canonical form). This property can be illustrated by the following example:

Example 3.1.Consider the nonlinear model of a simple pendulum

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \tfrac{1}{MR^2}(-bx_2 - MRg\, sin(x_1) + u) \end{aligned} \tag{3.30}$$

The flat output can be the pendulum position $y = x_1$. Indeed,

$$\begin{aligned} x_1 &= y \\ x_2 &= \dot{x}_1 = \dot{y} \\ u &= MR^2\ddot{y} + b\dot{y} + MRg\, sin(y) \end{aligned} \tag{3.31}$$

Therefore $x = A(y, \dot{y})$ and $u = B(y, \dot{y}, \ddot{y})$. Setting $z_1 = y$, $z_2 = \dot{y}$ and $v = y^{(2)}$, the system in Brunovsky canonical form is

$$\begin{aligned} \dot{z}_1 &= \dot{y} = z_2 \\ \dot{z}_2 &= \ddot{y} = v \end{aligned} \tag{3.32}$$

and the state variable and the control input are transformed to

$$\begin{array}{l} x_1 = y = z_1 \\ x_2 = \dot{y} = z_2 \\ u = MR^2v + bz_2 + MRg\ sin(z_1) \end{array} \tag{3.33}$$

so $x = A(z_1, z_2)$, $u = B(z_1, z_2, v)$. As we appreciate from this example, the solutions of the nonlinear system can be expressed as functions of the solutions of a linear one in the Brunovsky canonical form and conversely, the solutions of the linear system can be expressed as functions of the solutions the nonlinear one. This property precisely defines the (absolute) equivalence between systems. Another examples can be consulted in **Chelouah (1997)**.

A variety of examples have been shown to be differentially flat, and controllers based on trajectory generation by interpolation and error feedback on the obtained trajectory have been developed. These examples include overhead cranes **Fliess et al. (1993)**; **Fliess et al. (1995b)**, cars with trailers **Rouchon et al. (1992)**; **Rouchon et al. (1993)**, conventional aircraft **Martin et al. (1994)**; **Martin (1996a)**, induction motors **Chelouah et al. (1996)**; **Martin and Rouchon (1996b)**, active magnetic bearings **Levine et al. (1996)**, chemical reactors **Rothfuss et al. (1996)**; **Rouchon (1996)** and some fed-batch bioreactors **Radhakrishnan et al. (2001)**; **Picó-Marco (2004)**.

In general, a computable test for checking if $\dot{x} = f(x, u)$, $x \in R^n$, $u \in R^m$ is flat, remains up to now an open problem. This means there are no systematic methods for constructing flat outputs. The main difficulty in checking flatness is that a flat output $y = h(x, u, ..., u^p)$ may depend on derivatives of u of an arbitrary order p. Where this order p admits an upper bound (in terms of n and m) that is now completely unknown **Martin et al. (1997)**.

3.2.1 Flatness in the context of parametric uncertainty

Literature about robustness issues is abundant; however, only the result presented in **Cazaurang (1997)** addresses the analysis of parametric robustness of controllers based on flatness. Hagenmeyer and Delaleau **Hagenmeyer and Delaleau (2003b)** demonstrated that the differentially flat systems are linearizable by a nominal feedforward if the initial condition is known. A methodology of robustness analysis with respect to parametric uncertainty for exact feedforward linearization, based on differential flatness, was presented by Hagenmeyer and Delaleau **Hagenmeyer and Delaleau (2003c)** for SISO flat systems.

The methodology of robustness analysis is based on a pointwise stability analysis of the linearized tracking error system under parametric uncertainties, in conjuntion with an argument for non-linear systems with slowly varying input.

In this thesis, the flat output and its derivatives are expressed in terms of specification parameters. Consider the uncertain SISO non-linear system

$$\dot{x}(t) = f(\vartheta_p, x(t), u(t)),\ x(0) = x_0 \tag{3.34}$$

with time $t \in R$, state $x(t) \in R^n$ and input $u(t) \in R$. The vector field $f : R^{n_p} \times R^n \times R \rightarrow R^n$ is smooth. The uncertainty of the plant parameters $\vartheta_p \in R^{n_p}$ are considered as intervals. The system (3.34) is said to be differentially flat **Hagenmeyer and Delaleau (2003c)** if for all ϑ_p exists a flat output $y(t) \in R$, such that

$$y = h(x, u, \dot{u}, ..., u^{(p)}) \tag{3.35}$$

$$x = \varphi(\vartheta_p, y, \dot{y}, ..., y^{(k)}) \tag{3.36}$$

$$u = \psi(\vartheta_p, y, \dot{y}, ..., y^{(k+1)}) \tag{3.37}$$

where h, φ, and ψ are smooth functions at least in an open set of $R^{n+(p+1)}$, $R^{n_p+(k+1)}$ and $R^{n_p+(k+2)}$, respectively. In the equations (3.36) and (3.37), for every given trajectory of the flat output $t \mapsto y(t)$, the evolution of all other variables of the system $t \mapsto x(t)$ and $t \mapsto u(t)$ is also determined without integration of the system of differential equations. Moreover, for a sufficiently smooth desired trajectory of the flat output $t \mapsto y^*(t)$, equation (3.37) can be used to design the corresponding feedforward $u^*(t)$ directly for the nominal system parameters ϑ_p. The trajectory y^* is called the nominal trajectory, while the trajectory u^* is called the nominal control. The family of nominal feedforward controllers is given by

$$u^* = \psi(\vartheta_p, y^*, \dot{y^*}, ..., y^{*(k+1)}) \tag{3.38}$$

meaning to say, for each admissible nominal trajectory $y^*(t)$, there is a nominal feedforward u^*.

From equation (3.35), the flat output y can be obtained from the feedback system, using current values of state variables x, control signal u and some of its derivatives $u^{(1)}, .., u^{(p)}$. On the other hand, the flat output can be constructed externally with polynomial functions. Regarding the objectives stated in this thesis, we want to maintain the output of the controlled system within a specified region, so, we propose an approach where the flat output and its derivatives are defined as a region of trajectories in time parametrized with interval parameters. Specifically, if we let that each admissible nominal trajectory $y^*(t)$ and its derivatives $\dot{y}^*(t), ..., y^{*(k+1)}$ depend on specification parameters ϑ_s then

$$\begin{aligned} y^* &= \gamma(t, \vartheta_s) \\ y^{*(1)} &= \gamma_1(t, \vartheta_s) \\ &. \\ &. \\ y^{*(k)} &= \gamma_k(t, \vartheta_s) \\ y^{*(k+1)} &= \gamma_{k+1}(t, \vartheta_s) \end{aligned} \tag{3.39}$$

The associated family of state variables and nominal feedforward controls will be given by:

$$\begin{aligned} x &= \varphi(\vartheta_p, \gamma(t, \vartheta_s), \gamma_1(t, \vartheta_s), ..., \gamma_k(t, \vartheta_s)) \\ x &= \varphi(t, \vartheta_p, \vartheta_s) \end{aligned} \tag{3.40}$$

$$\begin{aligned} u^* &= \psi(\vartheta_p, \gamma(t, \vartheta_s), \gamma_1(t, \vartheta_s), ..., \gamma_{k+1}(t, \vartheta_s)) \\ u^* &= \psi(t, \vartheta_p, \vartheta_s) \Rightarrow \psi(t, \vartheta_k) \end{aligned} \tag{3.41}$$

That is to say, that the family of controllers u^* will be given in terms of process parameters ϑ_p and specification parameters ϑ_s. Equation (3.41) shows clearly that the determination of a family of robust controllers, will be subject to guarantee the fulfillment of the specifications ϑ_s and the determination of the permissible maximum uncertainty in process parameters ϑ_p.

We emphasize that if we use a fixed point of specification parameters $\bar{\vartheta}_s$ and a nominal plant $\bar{\vartheta}_p$, then, the trajectories for the desired output, as well as its derivatives $y^*, .., y^{*(k+1)}$, state variables x and control signal u^* represent unique trajectories in time as it is indicated in Figure 3.7.

However, if ϑ_s and ϑ_p are intervals, then the output and its derivatives $y^*, .., y^{*(k+1)}$, state variables x and control signal u^*, represent regions of trajectories in time as it is shown in Figure 3.8.

Recently, a new approach for the robustness analysis of tracking controllers based on flatness and using classic interval methods was proposed by **Antritter et al. (2007)**. Admissible intervals for the uncertain parameters were explicitly determined. The application of the robustness analysis was demonstrated for a feedforward and feedback tracking controller for the Van der Vusse type continuous stirred tank reactor.

On the other hand, differential flatness has been used to solve dynamic optimization problems with constraints **Faiz and Agrawal (2001)**; **Radhakrishnan et al. (2001)**; **Oldenburg and Marquardt (2002)**. The problems were reformulated as constrained optimization problems with algebraic equations instead of differential equations; to achieve this, the original dynamic optimization problem:

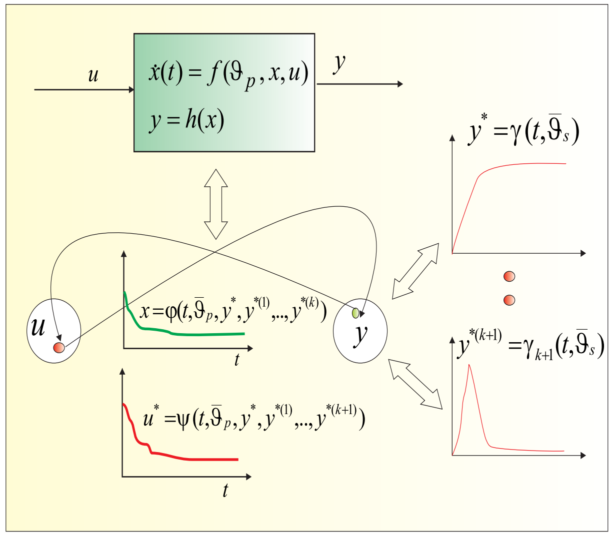


Fig. 3.7. Inversion of the dynamical system considering an single trajectory for the flat output.

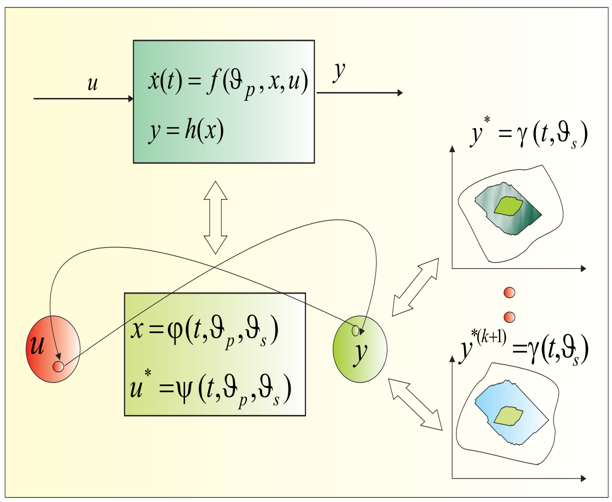


Fig. 3.8. Trajectories regions in time parametrized with interval parameters.

$$\min_{x_o, u(t)} J(x(t_f), u(t_f)) \quad subject\ to \begin{cases} \dot{x}(t) = f(x(t), u(t)),\ t \in [t_0, t_f] \\ 0 \leq c(x(t), u(t)),\ t \in [t_0, t_f] \\ 0 \leq c_f(x(t_f), u(t_f)), \\ 0 = x(t_0) - x_0 \end{cases} \tag{3.42}$$

is reformulated as the static optimization problem:

$$\min_{Y(t)} J(\varphi(Y(t_f)), \psi(Y(t_f))) \ subject\ to \begin{cases} 0 \leq c(\varphi(Y(t)), \psi(Y(t))), \\ t \in [t_0, t_f] \\ 0 \leq c_f(\varphi(Y(t_f)), \psi(Y(t_f))), \\ 0 = x_0 - \varphi(Y(t_0)) \end{cases} \tag{3.43}$$

where $Y = (y^*, \dot{y}^*, ..., y^{*(k+1)})$.

This thesis poses to solve the problem of robust control considering the optimization approach based on flatness. For the first time, the constraints are expressed in terms of set inclusion and the variables are quantified in this type of optimization problems. Thus, the problem will be formulated as the optimization of functions like, for instance:

$$Max\ \{\vartheta_p | \vartheta_p \in [\underline{\vartheta}_p, \overline{\vartheta}_p]\}$$

$$subject\ to \begin{cases} c_y(\varphi_x(t, \vartheta_p, \vartheta_s), \psi_u(t, \vartheta_p, \vartheta_s)) \subseteq \gamma_y(t), \quad t \in [t_0, t_f] \\ \varphi_x(t, \vartheta_p, \vartheta_s) \subseteq \gamma_x(t), \quad t \in [t_0, t_f] \\ \psi_u(t, \vartheta_p, \vartheta_s) \subseteq \gamma_u(t), \quad t \in [t_0, t_f] \\ \vartheta_p \in [\underline{\vartheta}_p, \overline{\vartheta}_p], \vartheta_s \in [\underline{\vartheta}_s, \overline{\vartheta}_s] \end{cases} \tag{3.44}$$

ϑ_s is a set specification parameters for the flat outputs. ϑ_p is a set of intervals for the plants, $\gamma_y(t)$ defines a region for flat outputs, $\gamma_x(t)$ defines bounding regions for the state variables. $\gamma_u(t)$ defines bounding regions for the control inputs. $\varphi_x(t, \vartheta_p, \vartheta_s)$ defines interval functions of the state variables and $\psi_u(t, \vartheta_p, \vartheta_s)$ the interval functions of the control input. $c_y(\varphi_x(t, \vartheta_p, \vartheta_s), \psi_u(t, \vartheta_p, \vartheta_s))$ defines general interval functions of the system outputs.

From equation (3.35), the system output can be set in several ways. Some of the forms can be:

1. The system output only depends on the state variables.
$$y = h(x) \tag{3.45}$$
2. The system output depends on the state variables and plant parameters.
$$y = h(x, \vartheta_p) \tag{3.46}$$
3. The system output depends on the state variables and control input.
$$y = h(x, u) \tag{3.47}$$
4. The system output depends on the state variables, plant parameters and control input.
$$y = h(x, u, \vartheta_p) \tag{3.48}$$
5. The system output depends on the state variables, control input and its derivatives.
$$y = h(x, u, \dot{u}, ..., u^{(p)}) \tag{3.49}$$
6. The system output depends on the state variables, control input, its derivatives and plant parameters.
$$y = h(x, u, \dot{u}, ..., u^{(p)}, \vartheta_p) \tag{3.50}$$

The general interval functions of the system output c_y for each of the previous cases can be reconstructed from output space as it is indicated in equations (3.51)-(3.56)

$$y = h(x) \Leftrightarrow y = c_y(t, \vartheta_s) \tag{3.51}$$

$$y = h(x, \vartheta_p) \Leftrightarrow y = c_y(t, \vartheta_s, \vartheta_p) \tag{3.52}$$

$$y = h(x, u) \Leftrightarrow y = c_y(t, \vartheta_s, \vartheta_k) \tag{3.53}$$

$$y = h(x, u, \vartheta_p) \Leftrightarrow y = c_y(t, \vartheta_s, \vartheta_k, \vartheta_p) \tag{3.54}$$

$$y = h(x, u, \dot{u}, ..., u^{(p)}) \Leftrightarrow y = c_y(t, \vartheta_s, \vartheta_k) \tag{3.55}$$

$$y = h(x, u, \dot{u}, ..., u^{(p)}, \vartheta_p) \Leftrightarrow y = c_y(t, \vartheta_s, \vartheta_k, \vartheta_p) \tag{3.56}$$

The constraints are formulated in general from a point of view of spaces interaction between flat outputs, states and controllers. That is, it is made a set inclusion relationship between the specifications space of the desired flat outputs trajectories compared with the flat output interval functions, the space-bound states compared with the interval functions of the states and the space-bounded controllers compared with controllers interval functions. The solution boxes set will be those that meet or satisfy all constraints for all instant of time within the specified interval.

To solve global optimization problems like the one above, two approaches can be applied to obtain ϑ_p **Hansen et al. (2004)**. In outer approximation, ϑ_p is approximated from the outside by a sequence of reduction of parameters and inclusion tests $\vartheta_{p1} \supset \vartheta_{p2} \supset \vartheta_{p3} ... \supset \vartheta_{p*}$. Being ϑ_{p*} the optimal solution. In inner approximation, ϑ_p is approximated from the inside by a sequence of expansion of parameters and inclusion tests $\vartheta_{p1} \subset \vartheta_{p2} \subset \vartheta_{p3} ... \subset \vartheta_{p*}$.

Since one desires to ensure compliance of the specifications and constraints with quantifiable parameters, the Modal Interval Analysis is the mathematical tool to deal with problems involving uncertainty and logical quantifiers (universal($\forall$), existential($\exists$)).

The main characteristic of the Modal Interval Arithmetic (MIA), is that an interval is not only seen as the bounding formed by two real numbers. The intervals are associated with quantified predicates (quantified constrained functions). The constrained functions can be true or false depending on the kind of quantifiers associated with variables. For instance $\forall(x \in [-2, 2]^{'})x > 0$ is false because for all x within the interval $[-2, 2]$ the constraint $x > 0$ is not met. However $\exists(x \in [-2, 2]^{'})x > 0$ is true because it exists within the interval $[-2, 2]$ a set of values to x that satisfy the constraint. Instead the system of classic intervals cannot to express the difference between universal $\forall(x \in [-2, 2]^{'})x > 0$ and existential $\exists(x \in [-2, 2]^{'})x > 0$ quantifiers **Gardeñes et al. (2001)**.

To explain in more detail problems that we will solve in this thesis, let us give the following example. With relation to the optimization approach from equation (3.44), given an interval function of the form $f(\vartheta_p, \vartheta_k) \subseteq \gamma(\vartheta_s)$ and a combination of quantifiers between specifications (ϑ_s), plants (ϑ_p) and controllers (ϑ_k) as it is indicated in Figure 3.9, we raised the following problems:

1. To determine the maximum admissible uncertainty ($\vartheta_p(\forall)$) by a nominal controller ($\vartheta_k(\exists)$) ensuring that some specifications are met ($\vartheta_s(\exists)$) and the constraints are satisfied. The considerations to solve the problem it is indicated in Figure 3.10. We consider the general case, where the system output $y = h(x, u, \vartheta_p)$ depends on the state variables, plant parameters and control input . The nominal controller is preset $\vartheta_k(\exists)$.

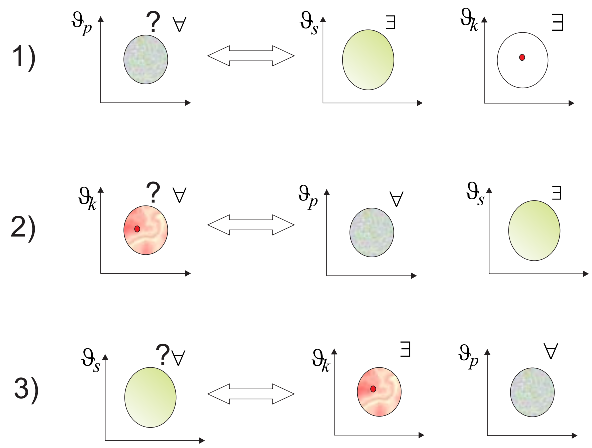


Fig. 3.9. Quantifiers in problems of inverse dynamic.

The free variable is ϑ_p. For the general case ϑ_p can be in all equations c_y, φ and ψ. So, a plant of the solution set is one that enforces the constraints $c_y \subseteq \gamma_y \wedge \varphi \subseteq \gamma_x \wedge \psi \subseteq \gamma_u$. Finally, we will find solutions inside specification parameters $\vartheta_s(\exists)$. The bounding regions γ_y, γ_x and γ_u are determined with a greater influence of the specification parameters ϑ_s over the functions γ_y, γ_x and γ_u since they are considered as intervals, while the parameters $\overline{\vartheta}_p$ and $\overline{\vartheta}_k$ are considered as specific points. A representation is indicated in Figure 3.11.

2. To determine the family of controllers $(\vartheta_k(\forall))$ that could ensure that some specification are met $(\vartheta_s(\exists))$ and the constraints are satisfied under parametric uncertainty in the plant $(\vartheta_p(\forall))$. The considerations to solve the problem is indicated in Figure 3.12. The free variable is ϑ_k. As in previous problem, ϑ_k can be in all equations c_y, φ and ψ. So, a controller of the solution set is one that enforces the constraints $c_y \subseteq \gamma_y \wedge \varphi \subseteq \gamma_x \wedge \psi \subseteq \gamma_u$. Each controller ϑ_k of the solution set will be robust under variations of plant parameters $\vartheta_p(\forall)$. The bounding regions γ_y, γ_x and γ_u are determined in a similar way as in the previous case.
3. To determine the achievable specifications $(\vartheta_s(\forall))$ by some nominal controllers $(\vartheta_k(\exists))$, satisfying the constraints under parametric uncertainty in the plant $(\vartheta_p(\forall))$.The considerations to solve the problem is indicated in Figure 3.13. The free variable is ϑ_s. As in previous problem, ϑ_s can be in all equations c_y, φ and ψ. So, a specification of the solution set that satisfies a nominal controller $\overline{\vartheta}_k$ is one that enforces the constraints $c_y \subseteq \gamma_y \wedge \varphi \subseteq \gamma_x \wedge \psi \subseteq \gamma_u$ under variations of plant parameters $\vartheta_p(\forall)$. The bounding regions γ_y, γ_x and γ_u are determined by using an auxiliary parameter ϑ_{ss} as it is indicated in Figure 3.14.

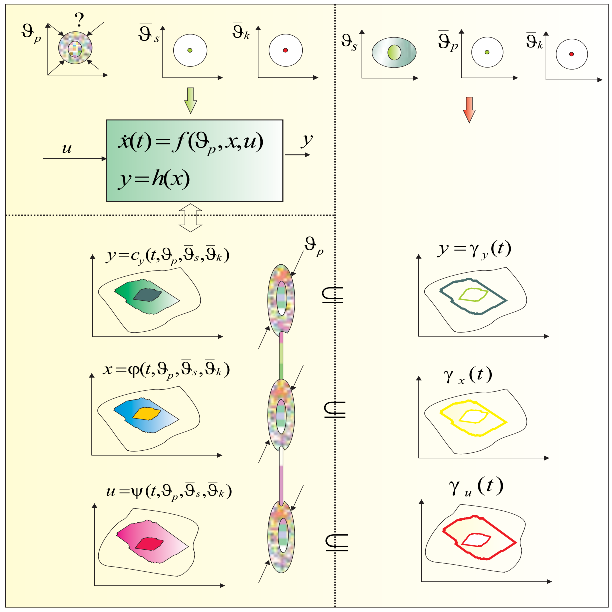


Fig. 3.10. Considerations to determine the maximum admissible uncertainty ($\vartheta_p(\forall)$) by a nominal controller ($\vartheta_k(\exists)$) ensuring that some specifications are met ($\vartheta_s(\exists)$) and the constraints are satisfied.

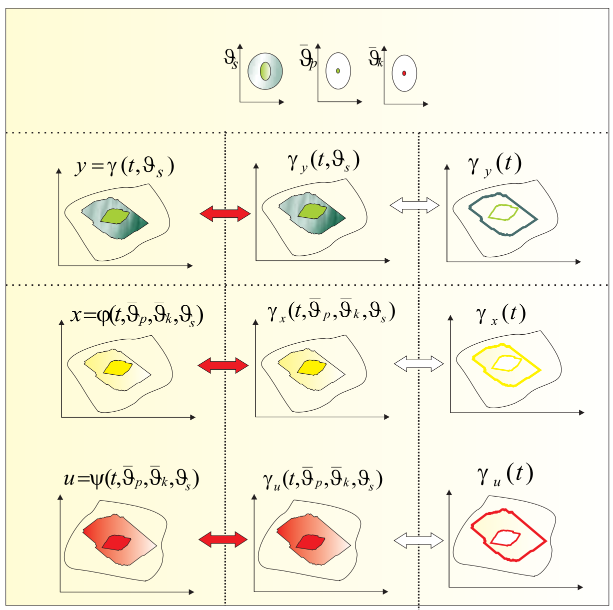


Fig. 3.11. Considerations to determine the bounding regions γ_y, γ_x and γ_u.

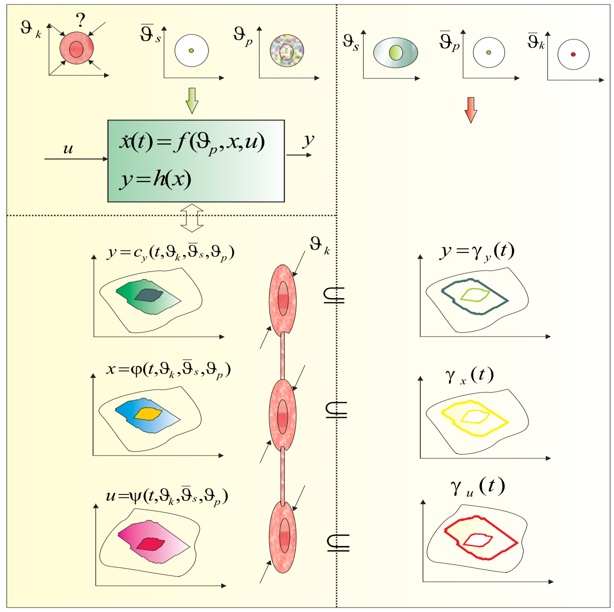


Fig. 3.12. Considerations to determine the family of controllers $(\vartheta_k(\forall))$ that could ensure that some specification are met $(\vartheta_s(\exists))$ and the constraints are satisfied under parametric uncertainty in the plant $(\vartheta_p(\forall))$.

This kind of inversion from specifications to controllers and from controllers to specifications with constraints and quantifiers cannot be solve of direct form with the classic interval analysis **Herrero (2006)**, **Bondia et al. (2006)**. The main reason is that with modal interval arithmetic we can search solutions of parameters in both sides of numeric relations (equality, inequality, inclusion) of interval functions. However, with the classic interval arithmetic we can search solutions only on one side of numeric relations of interval functions. When we try to solve problems that are associated with quantifiers or modes of selection, the classical interval arithmetic is limited because in its structure was not incorpored how to interpret the logical quantifiers. It was designed to perform arithmetic operations with intervals and outer approximations of the interval functions generally are computed.

In general, with classic interval arithmetic we can solve problems of constrains satisfaction of the form $f(x) > 0$, $f(x) < 0$ and $f(x) = 0$. For instance we can find all the equilibrium points of a nonlinear dynamic system of the form $\dot{x} = f(x)$. As in stable state $f(x) = 0$ we specified a initial box of parameter X_o and we find value sets to x such that the function $f(x)$ is near to zero. Other potentialities of the Modal Interval Arithmetic and limitations of the Classic Interval Arithmetic are presented in the next section.

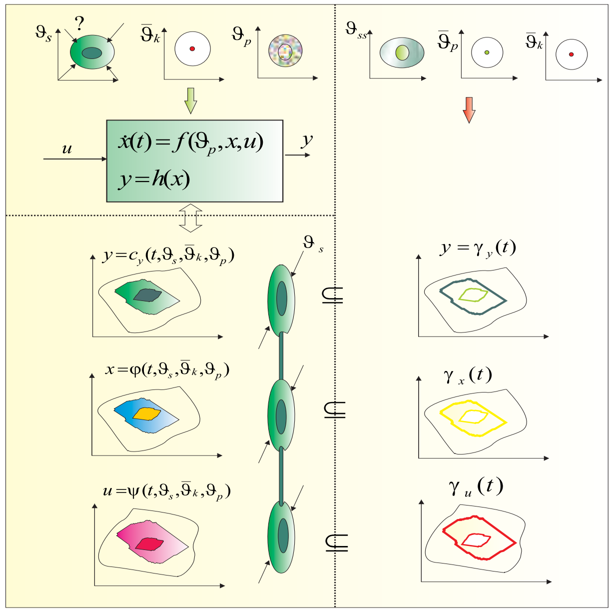


Fig. 3.13. Considerations to determine the achievable specifications ($\vartheta_s(\forall)$) by some nominal controllers ($\vartheta_k(\exists)$), satisfying the constraints under parametric uncertainty in the plant ($\vartheta_p(\forall)$).

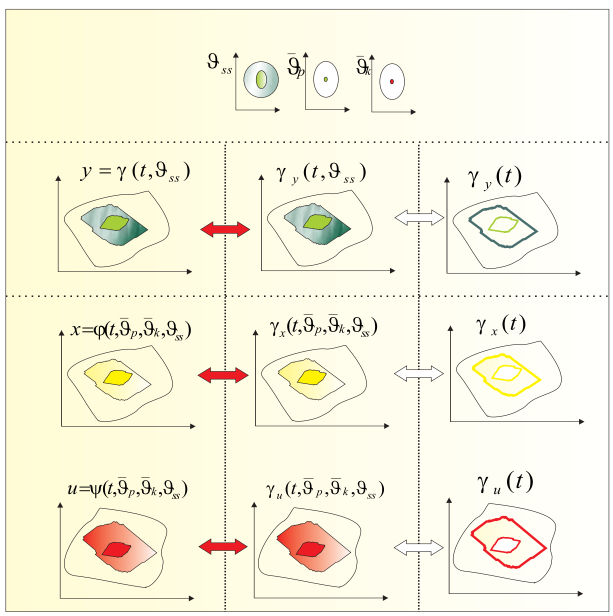


Fig. 3.14. Considerations to determine the bounding regions γ_y, γ_x and γ_u.

3.3 Modal Interval Analysis

3.3.1 Classic intervals and its limitations

The starting point for the interval analysis can be fixed in 1966 with the publication of Moore's book **Moore (1966)**. On this book was defined that an interval $[\underline{a}, \overline{a}]$ can be identified as the set of real numbers x such that $\underline{a} \leq x \leq \overline{a}$. The set of all intervals whose endings are real values is denoted by $I(R)$.

$$I(R) = \{[\underline{a}, \overline{a}] | \underline{a}, \overline{a} \in R, \underline{a} \leq \overline{a}\} \tag{3.57}$$

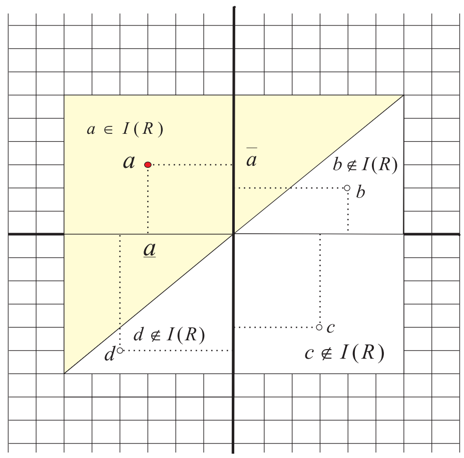


Fig. 3.15. Moore diagram.

In Moore's diagram, depicted in figure 3.15, the set of classic intervals are those that are within the shaded zone, or all those intervals that are on the left of the main diagonal. Some basic arithmetic operations $(+, -, *, /)$ between intervals are defined as:

$$\begin{aligned}
&[\underline{a}, \overline{a}] + [\underline{b}, \overline{b}] = [\underline{a} + \underline{b}, \overline{a} + \overline{b}] \\
&[\underline{a}, \overline{a}] - [\underline{b}, \overline{b}] = [\underline{a} - \overline{b}, \overline{a} - \underline{b}] \\
&[\underline{a}, \overline{a}] * [\underline{b}, \overline{b}] = [min(\underline{a} * \underline{b}, \underline{a} * \overline{b}, \overline{a} * \underline{b}, \overline{a} * \overline{b}), \\
&max(\underline{a} * \underline{b}, \underline{a} * \overline{b}, \overline{a} * \underline{b}, \overline{a} * \overline{b})] \\
&[\underline{a}, \overline{a}]/[\underline{b}, \overline{b}] = [\underline{a}, \overline{a}] * [1/\overline{b}, 1/\underline{b}] \ = [\underline{a}/\overline{b}, \overline{a}/\underline{b}] \ if \ 0 \notin [\underline{b}, \overline{b}] \ and \\
&\ if \underline{a} \geq 0, \overline{a} \geq 0, \underline{b} > 0, \overline{b} > 0 \\
&[\underline{a}, \overline{a}]/[\underline{b}, \overline{b}] = [\underline{a}/\underline{b}, \overline{a}/\underline{b}] \ if \ 0 \notin [\underline{b}, \overline{b}] \ and \\
&\ if \underline{a} < 0, \overline{a} \geq 0, \underline{b} > 0, \overline{b} > 0
\end{aligned} \tag{3.58}$$

If $a = [-2, 2]$ and $b = [3, 5]$ the basic arithmetic operations can be computed as:

$$\begin{aligned}
&a + b = [-2, 2] + [3, 5] = [1, 7] \\
&a - b = [-2, 2] - [3, 5] = [-7, -1] \\
&a * b = [-2, 2] * [3, 5] = [min(-2 * 3, -2 * 5, 2 * 3, 2 * 5), \\
&max(-2 * 3, -2 * 5, 2 * 3, 2 * 5)] = [-10, 10] \\
&a/b = [-2, 2] * (1/[3, 5]) = [-2, 2] * [1/5, 1/3] = [-2/3, 2/3]
\end{aligned} \tag{3.59}$$

If $x = [-2, 2]$ and $f = x - x$ then $f = [-2, 2] - [-2, 2] = [-4, 4]$. We can observe that the results for f is not zero. This is because in the Interval Analysis each variable is considered as an independent variable (two positions of memory) and not as a unique variable (one position of memory) as it is considered in real arithmetics. $[-4, 4]$ is an overbounded result which contains the real solution 0.

The same happens with the division $x/x = [1, 3]/[1, 3] = [1/3, 3]$ which is not equals 1. Classic intervals do not verify the distributive property. If $A, B, C \in I(R)$ one obtains a subset for $A * (B + C) \subseteq A * B + A * C$.

The simplest interval equations, as $A + X = B$ and $A * X = B$ can not always be solved. Even in the case that a solution exists, it may be not attainable through interval operations. For example the equation

$$[2, 5] + [x_1, x_2] = [5, 7] \Rightarrow [2, 5] + [3, 2] = [5, 7] \Rightarrow [x_1, x_2] = [3, 2] \notin I(R) \tag{3.60}$$

has not solution in $I(R)$, since the interval $[3, 2]$ cannot be represented as a classic interval (see fig. 3.15), since it does not fulfills the rule $\underline{a} \leq x \leq \overline{a}$. On the other hand, the equation

$$[2, 5] + [x_1, x_2] = [3, 7] \Rightarrow [2, 5] + [1, 2] = [3, 7] \Rightarrow [x_1, x_2] = [1, 2] \tag{3.61}$$

has solution in $I(R)$, but it is not achievable through interval operations since $[x_1, x_2] = [3, 7] - [2, 5] = [-2, 5] \neq [1, 2]$.

In the context of classic interval analysis, the so-called united extension R_f (the real range of the function) of a continuous function $f : R^n \to R$ at intervals $\boldsymbol{X}_1, ..., \boldsymbol{X}_n$, is defined as the range of values of the function.

$$R_f(\boldsymbol{X}_1, ..., \boldsymbol{X}_n) = \{f(x_1, ..., x_n) | x_1 \in \boldsymbol{X}_1, ..., x_n \in \boldsymbol{X}_n\} \tag{3.62}$$

The range will be given by:

$$[min\{f(x_1, ..., x_n) | x_i \in \boldsymbol{X}_i\}, max\{f(x_1, ..., x_n) | x_i \in \boldsymbol{X}_i\}] \tag{3.63}$$

On the other hand, if f is a real continuous function, its rational or natural extension is defined as a function $fR : I(R^n) \to I(R)$, obtained by replacing the variables $x_1, ..., x_n$ by their ranges $\boldsymbol{X}_1, ..., \boldsymbol{X}_n$, and the rational operations between the variables by the corresponding interval operations. Rational extensions fulfill the monotonous inclusion property which is essential in the context of the interval analysis. For instance if $\boldsymbol{X}_1 \subseteq \boldsymbol{Y}_1, ..., \boldsymbol{X}_n \subseteq \boldsymbol{Y}_n$, then $fR(\boldsymbol{X}_1, ..., \boldsymbol{X}_n) \subseteq fR(\boldsymbol{Y}_1, ..., \boldsymbol{Y}_n)$, if it is provided there are not divisions by intervals which contain zero. This property will allow the use of interval truncation for calculating with intervals, ensuring that the accurate result is inside of the calculated result.

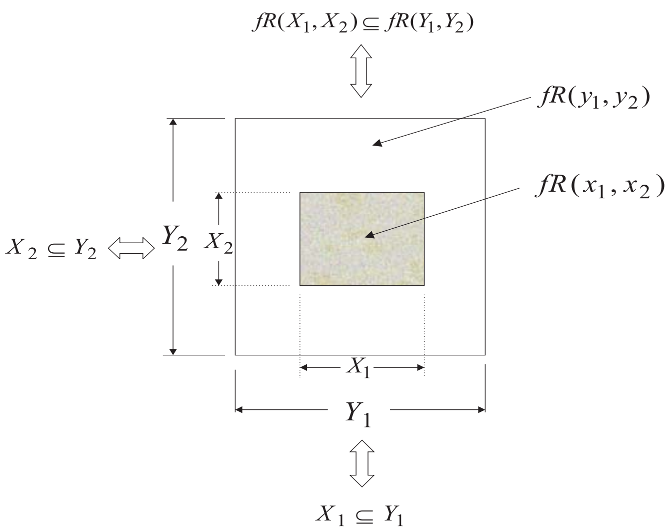


Fig. 3.16. Inclusion monotonicity.

If an interval $\boldsymbol{X}$ is partitioned in various subintervals $\boldsymbol{X}_1, \boldsymbol{X}_2, .., \boldsymbol{X}_n$, the union of the all rational extensions $fR(\boldsymbol{X}_1) \bigcup fR(\boldsymbol{X}_2) \bigcup, ..., \bigcup fR(\boldsymbol{X}_n) \subseteq fR(\boldsymbol{X})$ is a subset of the rational extension obtained with the interval $\boldsymbol{X}$. This property is very applied to reduce overestimation in the computation of interval extensions. A representation is indicated in Figure 3.17.

The relationship between the united extension and the rational extension is

$$R_f(\boldsymbol{X}_1, ..., \boldsymbol{X}_n) \subseteq fR(\boldsymbol{X}_1, ..., \boldsymbol{X}_n) \tag{3.64}$$

Thus $fR(\boldsymbol{X}_1, ..., \boldsymbol{X}_n)$ it is an overbound of $R_f(\boldsymbol{X}_1, ..., \boldsymbol{X}_n)$. For example, if the function $f(x) = \frac{x}{1+x}$ is defined for $\boldsymbol{X} = [2, 4]$ it obtains

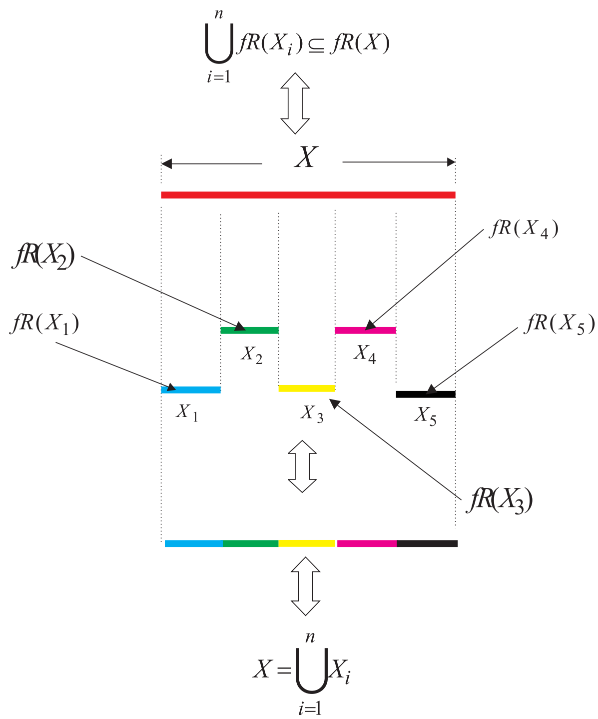


Fig. 3.17. Reduction of overestimation.

$$\begin{aligned} R_f([2,4]) &= \frac{1}{\frac{1}{x}+1} = \frac{1}{\frac{1}{[2,4]}+1} = \frac{1}{[\frac{5}{4},\frac{3}{2}]} = [\tfrac{2}{3},\tfrac{4}{5}] = [0.6, 0.8] \\ fR([2,4]) &= \frac{[2,4]}{[1,1]+[2,4]} = \frac{[2,4]}{[3,5]} = [\tfrac{2}{5},\tfrac{4}{3}] = [0.4, 1.3] \end{aligned} \tag{3.65}$$

Notice the united extension gives the real range of the function. Each instance of a variable is not considered as a different variable. That is, there is no problem of multiincidence. Yet, with the natural extension there is a problem of multiincidence. Each instance of a variable is considered as a different one. This, as already seen with basic arithmetic operations, leads to overbounding.

On the context from interval analysis, there are other interval extensions providing an extension closer to the united extension than the rational extension. These are centered form, average value and monotonous test form **Vehí (1998)**. These interval extensions are obtained considering different ways of expressing the rational function fR. Nevertheless, Hansen in his book **Hansen (2004)**, exposes: If a function is evaluated in its natural form with intervals, we can obtain a narrower result that can be obtained with Taylor's expansion. This represents an unsolved problem in interval analysis.

For instance, the function $f(x_1, x_2, x_3) = \frac{x_1}{x_2+2} + \frac{x_2}{x_3+2} + \frac{x_3}{x_1+2}$ if $\boldsymbol{X}_i = [-1, 1]$ $(i = 1, 2, 3)$ its natural extension is:

$$
\begin{array}{l}
f(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3) = \frac{\boldsymbol{X}_1}{\boldsymbol{X}_2+2} + \frac{\boldsymbol{X}_2}{\boldsymbol{X}_3+2} + \frac{\boldsymbol{X}_3}{\boldsymbol{X}_1+2} \\
f(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3) = \frac{[-1,1]}{[1,3]} + \frac{[-1,1]}{[1,3]} + \frac{[-1,1]}{[1,3]} \\
f(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3) = [-1,1] + [-1,1] + [-1,1] \\
f(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3) = [-3,3]
\end{array} \tag{3.66}
$$

Now, the Taylor's expansion for the function corresponds to:

$$
\begin{array}{l}
f(x_1, x_2, x_3) = f(x_1, x_2, x_3) + (\boldsymbol{X}_1 - x_1) g_1(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3) + \\
(\boldsymbol{X}_2 - x_2) g_2(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3) + (\boldsymbol{X}_3 - x_3) g_3(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3)
\end{array} \tag{3.67}
$$

where x_1, x_2, x_3 are points inside $\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3$ respectively. $g_i = \frac{df}{dx_i}$ $(i = 1, .., n)$ is the gradient of f. Thus, if $x_1 = 0$, $x_2 = 0$ and $x_3 = 0$, the function extension is:

$$
\begin{array}{l}
f(x_1, x_2, x_3) = f(x_1, x_2, x_3) + (\boldsymbol{X}_1 - x_1)(\frac{1}{\boldsymbol{X}_2+2} - \frac{\boldsymbol{X}_3}{(\boldsymbol{X}_1+2)^2}) + \\
(\boldsymbol{X}_2 - x_2)(\frac{1}{\boldsymbol{X}_3+2} - \frac{\boldsymbol{X}_1}{(\boldsymbol{X}_2+2)^2}) + (\boldsymbol{X}_3 - x_3)(\frac{1}{\boldsymbol{X}_1+2} - \frac{\boldsymbol{X}_2}{(\boldsymbol{X}_3+2)^2}) \\
f(x_1, x_2, x_3) = [-6, 6]
\end{array} \tag{3.68}
$$

We may see that the result obtained with natural extension $[-3, 3]$ is narrower that the one obtained with Taylor's expansion $[-6, 6]$. Hansen **Hansen (2004)** proposes to evaluate the components of g_i with some real points and some intervals instead of using only intervals for all the arguments to obtain a even more narrow result. Thus, if $g_1(\boldsymbol{X}_1, x_1, x_2)$, $g_2(\boldsymbol{X}_1, \boldsymbol{X}_2, x_3)$ and $g_3(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3)$ are used in Taylor's expansion, the function extension is reduced to:

$$
\begin{array}{l}
f(x_1, x_2, x_3) = f(x_1, x_2, x_3) + (\boldsymbol{X}_1 - x_1)(\frac{1}{x_2+2} - \frac{x_3}{(\boldsymbol{X}_1+2)^2}) + \\
(\boldsymbol{X}_2 - x_2)(\frac{1}{x_3+2} - \frac{\boldsymbol{X}_1}{(\boldsymbol{X}_2+2)^2}) + (\boldsymbol{X}_3 - x_3)(\frac{1}{\boldsymbol{X}_1+2} - \frac{\boldsymbol{X}_2}{(\boldsymbol{X}_3+2)^2}) \\
f(x_1, x_2, x_3) = [-4, 4]
\end{array} \tag{3.69}
$$

Rational and united extensions of a function f at intervals $\boldsymbol{X}_1, ..., \boldsymbol{X}_n$ have a single semantics,

$$
\begin{array}{l}
\forall(x_1 \in \boldsymbol{X}_1)...\forall(x_n \in \boldsymbol{X}_n)\exists(z \in R_f(\boldsymbol{X}_1, ..., \boldsymbol{X}_n))z = f(x_1, ..., x_n) \\
\forall(x_1 \in \boldsymbol{X}_1)...\forall(x_n \in \boldsymbol{X}_n)\exists(z \in fR(\boldsymbol{X}_1, ..., \boldsymbol{X}_n))z = f(x_1, ..., x_n)
\end{array} \tag{3.70}
$$

Now then, suppose we have a function $z = f(x_1, x_2)$. With classic interval arithmetics one can obtain an interval $\boldsymbol{Z}$ such that

$$
\forall(x_1 \in \boldsymbol{X}_1)\forall(x_2 \in \boldsymbol{X}_2)\exists(z \in \boldsymbol{Z})z = f(x_1, x_2) \tag{3.71}
$$

If x_1 is associated to specifications, x_2 for those of the plants, and z for those of the controllers. Therefore, with classic interval arithmetics one can answer the question: Give me a set of controllers such that for all plants and all specifications there can exist a valid controller within the set of controllers. Clearly, the solution to the problem of controller design is within the set of solutions provided. Yet, there are false solutions.

On the other hand, if x_2 is obtained one can answer the question: Give me a set of all plants such that for all specifications and the nominal controller, the equation $z = f(x_1, x_2)$ is fulfilled. That is, we would obtain the allowed process uncertainty. This is the problem solved in **Antritter et al. (2007)**.

with classic interval arithmetics one cannot solve equations with semantics such as:

$$\forall(x_2 \in \boldsymbol{X}_2)\forall(z \in \boldsymbol{Z})\exists(x_1 \in \boldsymbol{X}_1) z = f(x_1, x_2) \tag{3.72}$$

solving (3.72) would obtain the interval $\boldsymbol{Z}$ of all controllers such that for all plants in $\boldsymbol{X}_2$ there is some specification in $\boldsymbol{X}_1$ which is fulfilled.

Example 3.2.Given the function $z = x_1 u - x_2 v^2 sin(x_1) > 0$, we desire to find the set of possible values to z such that constrained function is greater than zero for all $x_1 \in [-10, 10]$, for all $x_2 \in [-10, 10]$, for all $u \in [-1, 1]$ and for all $v \in [-2, 2]$, to perform the arithmetic operations with the ends of the intervals the interpretation of the results is as follows:

$$\begin{aligned}&\forall(x_1 \in [-10, 10]')\forall(x_2 \in [-10, 10]')\\&\forall(u \in [-1, 1]')\forall(v \in [-2, 2]')\exists(z \in \boldsymbol{Z}')(z = x_1 u - x_2 v^2 sin(x_1) > 0)\end{aligned} \tag{3.73}$$

The interval extension obtained for the function $(x_1 u - x_2 v^2 sin(x_1))$ is $[-50, 50]$, then the set of values for z that satifies the constraint $x_1 u - x_2 v^2 sin(x_1) > 0$ can be expressed as $z = [\epsilon^+, 50]$. Being ϵ^+ any value positive close to zero. Then if $\epsilon^+ = 0.01$ the result is $z = [0.01, 50]$. This kind of semantic is solved with classic interval arithmetic.

In general with classical interval arithmetic we can solve different global optimization problems **Hansen (2004)** such as:

1. Unconstrained optimization. We have an unconstrained function and the optimization problem is posed as follows:

   $$Minimize(globally)\ f(\boldsymbol{x})$$

   Where f is a scalar function of a vector $\boldsymbol{x}$ of n components. We seek the solution set to $\boldsymbol{x}$ where the unconstrained function $f(\boldsymbol{x})$ has its global minimum.

2. Constrained optimization. In this kind of optimization problem we can have inequality and equality constraints, the optimization problem is raised as follows:

$$\mathit{Minimize}\ f(\boldsymbol{x}) \quad \mathit{subject\ to} \begin{cases} p_i(\boldsymbol{x}) \leq 0\ (i = 1, .., m) \\ q_i(\boldsymbol{x}) = 0\ (i = 1, .., r) \end{cases} \tag{3.74}$$

   We seek the solution set to $\boldsymbol{x}$ where the constrained function $f(\boldsymbol{x})$ has its global minimum.
3. Inequality constrained optimization. Here we only have inequality constraints, the optimization problem is as follows:

$$\mathit{Minimize}\ f(\boldsymbol{x}) \quad \mathit{subject\ to} \Big\{ p_i(\boldsymbol{x}) \leq 0\ (i = 1, .., m) \tag{3.75}$$

   we seek the solution set to $\boldsymbol{x}$ where the inequality constrained function $f(\boldsymbol{x})$ has its global minimum.
4. Equality constrained optimization. In this case we have equality constrains only. The problem is raised of the following form:

$$\mathit{Minimize}\ f(\boldsymbol{x}) \quad \mathit{subject\ to} \Big\{ q_i(\boldsymbol{x}) = 0\ (i = 1, .., r) \tag{3.76}$$

   we seek the solution set to $\boldsymbol{x}$ where the equality constrained function $f(\boldsymbol{x})$ has its global minimum.
5. Perturbed optimization. In this case, a vector of interval parameters $\boldsymbol{c}$ is considered in the constrained function thus as the interval vector $\boldsymbol{x}$ and the optimization problem is raised as follows:

$$\underbrace{\mathit{Minimize}}_{\boldsymbol{x},\boldsymbol{c}}\ f(\boldsymbol{x}, \boldsymbol{c}) \quad \mathit{subject\ to} \begin{cases} p_i(\boldsymbol{x}, \boldsymbol{c}) \leq 0\ (i = 1, .., m) \\ q_i(\boldsymbol{x}, \boldsymbol{c}) = 0\ (i = 1, .., r) \\ \boldsymbol{x} \in [\underline{x}, \overline{x}], \boldsymbol{c} \in [\underline{c}, \overline{c}] \end{cases} \tag{3.77}$$

   we seek the set of values to $\boldsymbol{c}$ and $\boldsymbol{x}$ where the perturbed constrained function $f(\boldsymbol{x}, \boldsymbol{c})$ has its global minimum.

3.3.2 Modal intervals

As we have seen in the previous section, classic interval arithmetics has some deficiencies. Modal Interval Analysis is an extension of the interval analysis that recovers some of the properties required by a numerical system. It simplifies the computation of interval functions. Moreover, it allows richer semantic interpretation of the results than classic intervals.

The development and theoretical foundation of the Interval Modal Analysis can be found in **Gardeñes et al. (1980)**; **Gardeñes et al. (1985)**; **Gardeñes et al. (2001)**; **Remei (2005)**.

Let us specify an interval $\boldsymbol{X}^{'}$ which states and delimits a real number x that verifies a given predicate $P(x)$. This necessarily requires a choice between the existential and universal quantifiers to build two logical expressions.

$$\begin{aligned}&\exists(x \in \boldsymbol{X}^{'})P(x) : exists\ an\ element\ x\ within\ \boldsymbol{X}^{'}\ that\ verifies\ P(x)\\&\forall(x \in \boldsymbol{X}^{'})P(x) : all\ element\ x\ of\ the\ set\ \boldsymbol{X}^{'}\ verifies\ P(x)\end{aligned} \tag{3.78}$$

To work with logical expressions such as (3.78) with existential and universal quantifiers, modal intervals are defined as tuples formed by a classic interval and a quantifier. Thus, the set of modal intervals will be represented by $I^{*}(R) := \{(\boldsymbol{X}^{'}, \{\exists, \forall\})|\boldsymbol{X}^{'} \in I(R)\}$. The modal coordinates of a modal interval are its point-set domain and its modality.

If the modality of a modal interval is existential $\boldsymbol{X} = (\boldsymbol{X}^{'}, \exists)$ then the modal interval is proper and if the modality is universal $\boldsymbol{X} = (\boldsymbol{X}^{'}, \forall)$ then the modal interval is improper. For example the interval $[2, 5]$ corresponding to a proper modal interval $\boldsymbol{X} = ([2, 5]^{'}, \exists)$. The interval $[2, 1]$ is perfectly valid, and corresponds to an improper modal interval $\boldsymbol{X} = ([1, 2]^{'}, \forall)$. In particular, an interval as $[3, 3]$ is denominated point-interval. Its modality can be considered like proper $\boldsymbol{X} = ([3, 3]^{'}, \exists)$ or improper $\boldsymbol{X} = ([3, 3]^{'}, \forall)$.

The canonical representation for a modal interval is:

$$\boldsymbol{X} = [a, b] := \begin{cases} ([a, b]^{'}, \exists)\ \underline{if}\ \ a \leq b \\ ([b, a]^{'}, \forall)\ \underline{if}\ \ a \geq b \end{cases} \tag{3.79}$$

For an interval $\boldsymbol{X} = [a, b]$, the operators Prop, Impr and Dual are defined by

$$\begin{aligned}&Dual([a, b]) = [b, a]\\&Prop([a, b]) = [min\{a, b\}, max\{a, b\}] \in (\boldsymbol{X}^{'}, \exists)\\&Impr([a, b]) = [max\{a, b\}, min\{a, b\}] \in (\boldsymbol{X}^{'}, \forall)\end{aligned} \tag{3.80}$$

The extern (OutR) and internal (InnR) rounding of an interval is defined as:

$$\begin{aligned}&OutR([a, b]) = [Left(a), Right(b)]\\&InnR([a, b]) = [Right(a), Left(b)]\end{aligned} \tag{3.81}$$

Remark: In order to avoid confusions between classical and modal intervals, a classical interval of bounds a and b are represented by $[a, b]^{'}$, instead of the standard notation $[a, b]$.

3.3.3 f^* and f^{**} semantic extensions

In the theory of the modal intervals, we can compute two extensions for a function. These are the f^* and f^{**} semantic extensions. For instance if $\boldsymbol{X} = (\boldsymbol{X}_p, \boldsymbol{X}_i)$ is the proper and improper components vector, the f^* and f^{**} extensions are expressed as:

$$\begin{aligned}
&f^*(\boldsymbol{X}) := \vee(x_p \in \boldsymbol{X}_p^{'}) \wedge (x_i \in \boldsymbol{X}_i^{'})[f(x_p, x_i), f(x_p, x_i)] \\
&= [min(x_p \in \boldsymbol{X}_p^{'}) max(x_i \in \boldsymbol{X}_i^{'}) f(x_p, x_i), \\
&max(x_p \in \boldsymbol{X}_p^{'}) min(x_i \in \boldsymbol{X}_i^{'}) f(x_p, x_i)], \\
&f^{**}(\boldsymbol{X}) := \wedge(x_i \in \boldsymbol{X}_i^{'}) \vee (x_p \in \boldsymbol{X}_p^{'})[f(x_p, x_i), f(x_p, x_i)] \\
&= [max(x_i \in \boldsymbol{X}_i^{'}) min(x_p \in \boldsymbol{X}_p^{'}) f(x_p, x_i), \\
&min(x_i \in \boldsymbol{X}_i^{'}) max(x_p \in \boldsymbol{X}_p^{'}) f(x_p, x_i)].
\end{aligned} \tag{3.82}$$

where $\vee$ and $\wedge$ are Join and Meet operators respectively **Herrero et al. (2005)**. These operators are explained in the following example.

Example 3.3.Given a family of intervals $\boldsymbol{A}(i)$ with values $\boldsymbol{A}(1) = [2, 5]$, $\boldsymbol{A}(2) = [6, 2]$, $\boldsymbol{A}(3) = [1, -5]$, $\boldsymbol{A}(4) = [-1, 6]$, $\boldsymbol{A}(5) = [-3, 2]$ and $\boldsymbol{A}(6) = [4, -2]$ the Join $\vee$ and Meet $\wedge$ operations are computed as follows: The Join $\vee$ operator obtains a minimun value from all the minimun values of the family of intervals and then, it obtains a maximum value from all the maximum values of the family. The Meet $\wedge$ operator obtains a maximum value from all the minimun values of the family of intervals and then, it obtains a minimun value from all the maximum values of the family. The mentioned above can be expressed of the following form:

$$\begin{aligned}
&\vee(i, I)\boldsymbol{A}(i) = [min\{2, 6, 1, -1, -3, 4\}, max\{5, 2, -5, 6, -2\}] = [-3, 6] \\
&\wedge(i, I)\boldsymbol{A}(i) = [max\{2, 6, 1, -1, -3, 4\}, min\{5, 2, -5, 6, -2\}] = [6, -5]
\end{aligned} \tag{3.83}$$

as we have seen in Figure 3.18, the result is a box that contains all members of the family of intervals. The range computed by the Meet operator is the lower value (minimun) from box and the range computed by the Join operator is the upper value (maximum) from box.

In the case that $\boldsymbol{X}_i = 0$ then $f^* = f^{**}$

$$\begin{aligned}
&f^*(\boldsymbol{X}) = f^{**}(\boldsymbol{X}) \\
&= \vee(x \in \boldsymbol{X}^{'})[f(x), f(x)] = [min(x \in \boldsymbol{X}^{'}) f(x), max(x \in \boldsymbol{X}^{'}) f(x)]
\end{aligned} \tag{3.84}$$

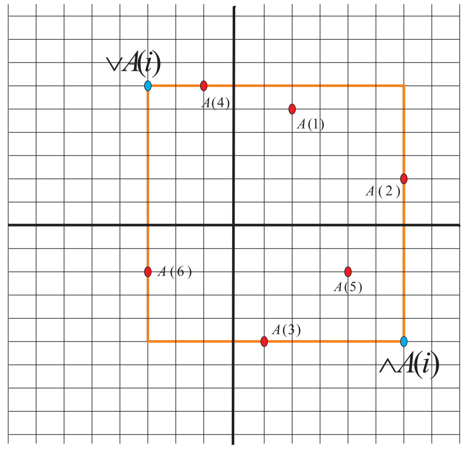


Fig. 3.18. Meet and join operations for a family of intervals.

In this case f^* is equal to the united extension (real extension of the function) of the classical intervals. In Figure 3.19 we make the observation that when carrying out the calculation of f^*, one first computes the Meet operation, and then the Join operation, and vice versa for the calculation of f^{**}. First one computes the Join operation, and then the Meet operation (see Figure 3.20).

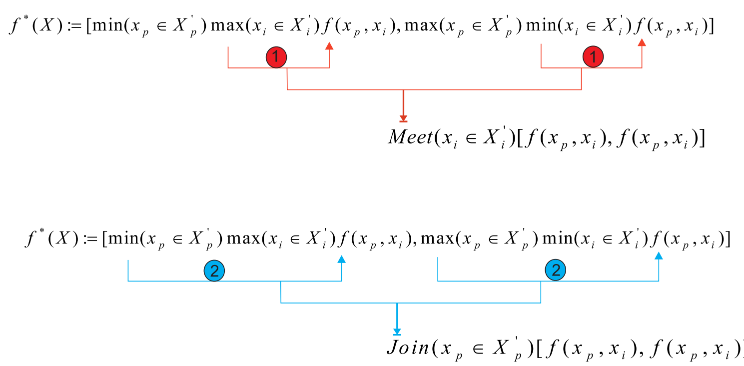


Fig. 3.19. Lattice meet and join operations in f^*.

Example 3.4.Given the function $f(x_1, x_2) = x_1^2 + x_2^2$ obtain the f^* extension for $\boldsymbol{X}_1 = [-1, 1]$ (proper) and $\boldsymbol{X}_2 = [1, -1]$) (improper).

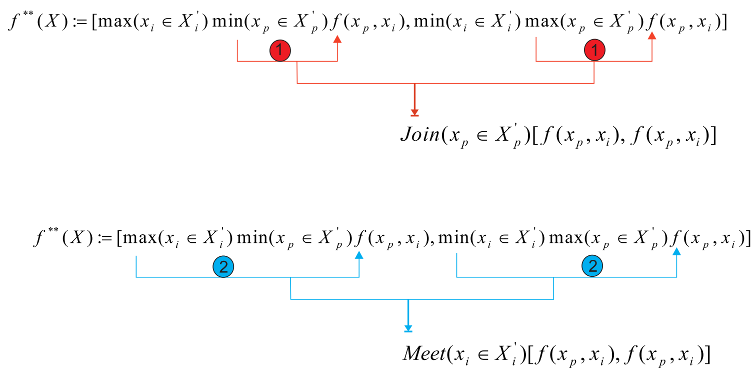


Fig. 3.20. Lattice meet and join operations in f^{**}.

1. Let us obtain $\wedge(x_2 \in [-1,1]^{'})[x_1^2 + x_2^2, x_1^2 + x_2^2]$. As the maximum value for the function $(x_1^2 + x_2^2)$ for $\forall(x_2 \in [-1,1]^{'})$ occurs when $x_2 = -1$ or $x_2 = 1$ then

$$max(x_1^2 + x_2^2) = x_1^2 + (-1)^2 = x_1^2 + 1 \tag{3.85}$$

and the $min(x_1^2 + x_2^2)$ for $\forall(x_2 \in [-1,1]^{'})$ occurs when $x_2 = 0$ then

$$min(x_1^2 + x_2^2) = x_1^2 \tag{3.86}$$

Let us stress that all the values within the interval x_2 were considered to calculate the minimum of the function. Therefore $\wedge(x_2 \in [-1,1]^{'})[x_1^2 + x_2^2, x_1^2 + x_2^2] = [x_1^2 + 1, x_1^2]$. Notice that the proper component x_1 was not considered in the first step.

2. Let us obtain $\vee(x_1 \in [-1,1]^{'})[x_1^2 + 1, x_1^2]$. As the minimun value for the function $min(x_1^2 + 1)$ for $\forall(x_1 \in [-1,1]^{'})$ occurs when $x_1 = 0$ then

$$min(x_1^2 + 1) = 1 \tag{3.87}$$

and $max(x_1^2)$ for $\forall(x_1 \in [-1,1]^{'})$ occurs when $x_1 = -1$ or $x_1 = 1$ then

$$max(x_1^2) = (-1)^2 = 1 \tag{3.88}$$

Therefore $f^*(x_1, x_2) = f^*([-1,1],[1,-1]) = [1,1]$. The process is similar for the calculation of f^{**}. In previous example we saw that f^* cannot be obtained if we perform inteval operations only with the extremes of the intervals. It is necesary to consider values within intervals to compute the maximum and minimum ranges of the functions.

The f^{**} extension can be computed from f^* extension using the Dual operation. Thus, $f^{**}(\boldsymbol{X}) = Dual(f^*(Dual(\boldsymbol{X})))$.

The most interesting feature of modal intervals is the semantic interpretation of the results. The f^* and f^{**} extensions can be interpreted with the $*$ and $**$ semantic theorems respectively. These theorems are as follows:

THEOREM 3.1. *(f^* **Semantic theorem**). Given a modal interval vector $\boldsymbol{A} \in I^*(R^n)$, a function $f : R^n \to R$ continuous on $\boldsymbol{A}'$, and a modal interval $F(\boldsymbol{A}) \in I^*(R)$ then,*

$$f^*(\boldsymbol{A}) \subseteq F(\boldsymbol{A}) \Leftrightarrow \forall(a_p \in \boldsymbol{A'}_p)Q(z \in F(\boldsymbol{A}))\exists(a_i \in \boldsymbol{A}_i^{'})(z = f(a_p, a_i)) \tag{3.89}$$

◇

That is, for any partition of parameters $\boldsymbol{A} = (\boldsymbol{A}_p, \boldsymbol{A}_i)$ in proper $\boldsymbol{A}_p$ and improper $\boldsymbol{A}_i$ ones, any superset $F(\boldsymbol{A})$ of f^* admits that semantics. Also we make the clarification that the theorem defines that the type of quantifier will be existential ($\exists$) for the improper parameters and universal ($\forall$) for the proper parameters. Q is the type of quantifier that is assigned to $\boldsymbol{Z}$ according to the obtained result.

If the result is proper then the quantifier to $\boldsymbol{Z}$ is existential ($\exists$) and if it is improper the quantifier is universal ($\forall$).

$F(\boldsymbol{A})$ is an extension of f function obtained with natural extension, centered forms or the study of function monotonicity for instance. Therefore, the interpretation reads "For all elements belonging to the proper intervals, there exists at least one element in the improper ones such that, either $\forall$" or $\exists$ (depending on Q) $\boldsymbol{Z}$ in the superset of $f^*(\boldsymbol{A})$, the equation $\boldsymbol{Z} = f(a_p, a_i)$ is fulfilled.

Example 3.5.Given a proper interval $\boldsymbol{a} = [10, 20]$ and an improper interval $\boldsymbol{b} = [20, 15]$. Let us consider the function $z = a + b$. Obtain the z^* semantic extension for the function.

$$z^* = [\underbrace{min}_{a}\ \underbrace{max}_{b}(a + b), \underbrace{max}_{a}\ \underbrace{min}_{b}(a + b)] \tag{3.90}$$

1. Let us obtain $\wedge(b \in [15, 20]^{'})[a + b, a + b]$. As the maximum value for the function $(a + b)$ for $\forall(b \in [15, 20]^{'})$ occurs when $b = 20$ then

$$max(a + b) = a + 20 \tag{3.91}$$

and the $min(a + b)$ for $\forall(b \in [15, 20]^{'})$ occurs when $b = 15$ then

$$min(a+b) = a + 15 \tag{3.92}$$

So, $\wedge(b \in [15,20]^{'})[a+b, a+b] = [a+20, a+15]$.

2. Let us obtain $\vee(a \in [10,20]^{'})[a+20, a+15]$. As the minimun value for the function $min(a+20)$ for $\forall(a \in [10,20]^{'})$ occurs when $a = 10$ then

$$min(a+20) = 30 \tag{3.93}$$

and $max(a+15)$ for $\forall(a \in [10,20]^{'})$ occurs when $a = 20$ then

$$max(a+15) = 35 \tag{3.94}$$

Therefore $z^*(a,b) = f^*([10,20],[20,15]) = [30,35]$. Which is proper. Therefore, the semantic interpretation is

$$\forall(a \in [10,20]^{'})\exists(b \in [15,20]^{'})\exists(z \in [30,35]^{'})(a+b=z) \tag{3.95}$$

THEOREM 3.2. *(f^{**} **Semantic theorem**). Given a modal interval vector $\boldsymbol{A} \in I^*(R^n)$, a function $f : R^n \rightarrow R$ continuous on $\boldsymbol{A}^{'}$, and $F(\boldsymbol{A}) \in I^*(R)$ then,*

$$f^{**}(\boldsymbol{A}) \supseteq F(\boldsymbol{A}) \Leftrightarrow \forall(a_i \in \boldsymbol{A'}_i)Q(z \in Dual(F(\boldsymbol{A})))\exists(a_p \in \boldsymbol{A}^{'}_p)(z = f(a_p, a_i)) \tag{3.96}$$

We can see that the f^{**} semantic theorem can be obtained from f^*, through the complement of f^*. In this case, the improper components are universally quantified and proper components of form existential. The modality of z is also exchanged with the dual operator.

◇

Example 3.6.Given the function $f = a + b$ and $\boldsymbol{a} = [1,2]$ and $\boldsymbol{b} = [5,7]$ proper intervals, to perform the arithmetic operation we obtain:

$$\boldsymbol{f} = [1,2] + [5,7] = [6,9] \tag{3.97}$$

in the context of Classic Interval Arithmetic the semantic interpretation corresponds to:

$$\forall(a \in [1,2]')\forall(b \in [5,7]')\exists(f \in [6,9]')(f = a + b) \quad (3.98)$$

That is, for all $a \in [1,2]$ and for all $b \in [5,7]$ exist values within the interval $f \in [6,9]$ that satisfies the function $f = a + b$.

in addition to this one, in the context of Modal Interval Analysis the f^{**} semantic interpretation is as follows:

$$\forall(f \in [6,9]')\exists(a \in [1,2]')\exists(b \in [5,7]')(a + b = f) \quad (3.99)$$

as the intervals $\boldsymbol{a}$ and $\boldsymbol{b}$ are propers, then $\boldsymbol{a}$ and $\boldsymbol{b}$ are existentially quantified according to the f^{**} semantic theorem. The result computed to $\boldsymbol{f}$ is $[6,9]$ and corresponds with the existential quantifier according to the f^{*} semantic theorem but its modality is exchanged with the dual operator to be interpreted correctly with the f^{**} semantic theorem. So, $\boldsymbol{f}$ is universally quantified. That is, exist values within $a \in [1,2]$ and $b \in [5,7]$ such that any value within the interval $f \in [6,9]$ can be achieved.

Unfortunately, the computation of the f^{*} and f^{**} extensions is in general, a difficult challenge. Therefore the usual procedure is to find overbounded computations of f^{*} and underbounded computations of f^{**} which maintain the semantic interpretations **Gardeñes et al. (1985)**; **Armengol (1999)**. A representation between the $f^{*}(\boldsymbol{A})$ and $f^{**}(\boldsymbol{A})$ extensions and $F(\boldsymbol{A})$ can be seen in Figure 3.21.

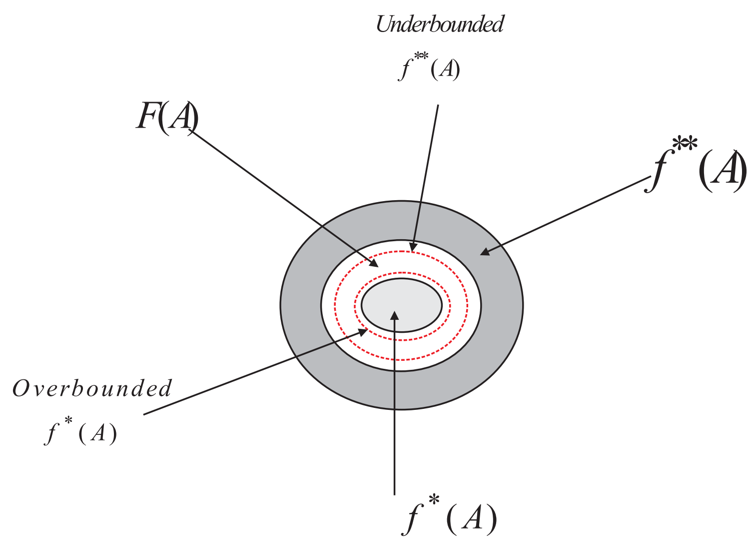


Fig. 3.21. $f^{*}(\boldsymbol{A}) \subseteq F(\boldsymbol{A}) \subseteq f^{**}(\boldsymbol{A})$.

Another form to compute the f^* extension is with the FSTAR algorithm **Herrero et al.(2005)**. This algorithm obtains an overbounded and underbounded covering to f^*. The basic algorithm consists in the following steps.

The parameters vector is grouped in proper ($\boldsymbol{u}$) and improper ($\boldsymbol{v}$) components. The proper ($\boldsymbol{u}$) and improper ($\boldsymbol{v}$) components are divided in cells and strips. For each cell, an inner and outer aproximation is computed through tests of monotonicity of the interval function. The Meet operator is applied to compute an inner and outer aproximation for each strip from inner and outer aproximations of cells.

$$\begin{aligned} Out(Strip_1) &= \wedge(Out(Cell_{11}), Out(Cell_{12})) \\ Inn(Strip_1) &= \wedge(Inn(Cell_{11}), Inn(Cell_{12})) \end{aligned} \tag{3.100}$$

Finally, the Join operator is applied to compute the inner and outer aproximations to f^* from inner and outer aproximations of previous strips. In equation 3.101 the operations are indicated. See in appendix a basic example.

$$\begin{aligned} Outer(f^*) &= \vee(Out(Strip_1), .., Out(Strip_j)) \\ Inner(f^*) &= \vee(Inn(Strip_1), .., Inn(Strip_j)) \end{aligned} \tag{3.101}$$

3.3.4 Applications of the Modal Interval Analysis

In the literature regarding to these approaches, there are different contributions and applications for the robust control of dynamic systems using tools based on modal intervals. Vehí and Sainz **Vehí and Sainz (1999)** proposed necessary and sufficient conditions for robust stability, Armengol **Armengol (1999)** applied the Modal Interval Analysis to the simulation of the dynamic behavior of the systems with uncertain parameters. An algorithm based on Modal Interval Analysis was developed to obtain error-bounded envelopes applied to failures diagnosis. Inner and outer approximations to the range of the functions at each time point were computed. The algorithm applied theorems of partial and optimal coercion to the monotonic variables in order to compute the range of the functions.

In 1997, Malan **Malan et al. (1997)** carried out the robust analysis and design of control systems using classic interval arithmetic, and in 1998 Vehí **Vehí (1998)** applied for the first time the Modal Interval Analysis to the field of control engineering, developing a modal interval formulation of the problem of robust control for linear systems with parametric uncertainty. Different semantic theorems of the modal interval theory and the results of optimality to rational functions were applied to obtain necessary and sufficient conditions for robust stability. A feedback control system indicated in Figure 3.22 was considered, with an uncertain system $G(s, \mathbf{q})$ of the form indicated in equation (3.102)

$$G(s,\mathbf{q}) = \frac{\alpha_o(\mathbf{q})+\alpha_1(\mathbf{q})s+...+\alpha_m(\mathbf{q})s^m}{\beta_o(\mathbf{q})+\beta_1(\mathbf{q})s+...+\beta_n(\mathbf{q})s^n} \tag{3.102}$$

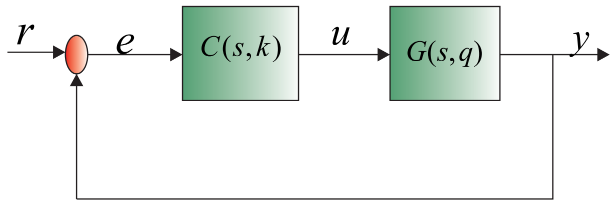


Fig. 3.22. Feedback uncertain system.

with coefficients α_i and β_j depending arbitrarily of a structured disturbance characterized by the vector of parameters $\mathbf{q} = [q_1 \ q_2 \ ... \ q_l]^T$ where every parameter is bounded within an uncertainty domain as follows:

$$\mathbf{Q} = \left\{ \mathbf{q} = [q_1 \ q_2 \ ... \ q_l]^T | q_i \in [\underline{q}_i, \overline{q}_i], i = 1, ..., l \right\} \tag{3.103}$$

$C(s,\mathbf{k})$, is a feedback controller that depends on the vector of parameters $\mathbf{k}$.

$$\mathbf{k} = [k_1 \ k_2 \ .. \ k_l]^T \tag{3.104}$$

the problem of design was searching and evaluating some fixed values to the parameters $\mathbf{k}^o = [k_1^o \ k_2^o \ .. \ k_l^o]$ of the feedback controller such that the closed-loop system satisfies some specifications of robustness.

In **Vehí (1998)** it was showed that the performance specifications of controlled system can be expressed in terms of a closed-loop characteristic polynomial $p(s,\mathbf{q},\mathbf{k})$ or in terms of domain frequency. In both cases they are reduced to a set of inequalities to satisfy. In the following equation an inequalities is indicated

$$f_i(\alpha,\mathbf{q},\mathbf{k}) > 0, \ \forall \alpha \in \boldsymbol{A}, \ \forall \boldsymbol{q} \in \boldsymbol{Q}, \ \forall \boldsymbol{k} \in \boldsymbol{K} \tag{3.105}$$

where $\boldsymbol{A}$ is the variation interval of the generalized frequency α, and $\boldsymbol{K}$ can be a nominal controller $\boldsymbol{k}^o$ or a certain domain in the parameters space of the controller depending on the considered problem. With this formulation different problems from the robust control were raised:

1. Verification of the specifications fulfillment. Given the uncertain system $G(s,\boldsymbol{q})$ and variation domain of the plant parameters $\boldsymbol{Q}$, verify if the designed controller $C(s,\boldsymbol{k}^o)$ fulfills the robustness specifications:

$$f_i(\alpha, \boldsymbol{Q}, \boldsymbol{k}^o) > 0 \tag{3.106}$$

A representation of the problem is indicated in Figure 3.23. As we see the problem is related to find the achievable specifications by a nominal controller under parametric uncertainty.

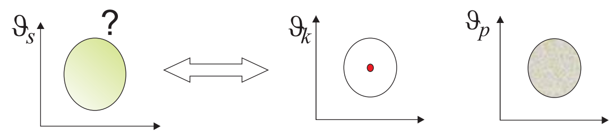


Fig. 3.23. Verification of the fulfillment of specifications.

2. Calculation of the robustness margin: Given a nominal plant, $G(s, q^o)$, to prove the maximum domain $\boldsymbol{Q}^*$ so that the designed controller fulfills the robustness specifications:

$$f_i(\alpha, \boldsymbol{Q}^*, \boldsymbol{k}^o) > 0 \tag{3.107}$$

A representation of the problem is indicated in Figure 3.24. In this problem the admissible maximum uncertainty by a nominal controller is obtained.

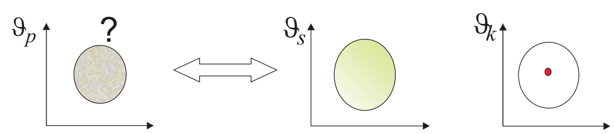


Fig. 3.24. Calculation of the robustness margin.

3. Design of a robust controller: Assuming certain structures of the controller, and variation domain of the plant parameters $\boldsymbol{Q}$, prove a fixed controller $C(s, \boldsymbol{k}^o)$ so that the closed-loop controlled system fulfills the robustness specifications:

$$f_i(\alpha, \boldsymbol{Q}, \boldsymbol{k}^o) > 0 \tag{3.108}$$

A representation of the problem is indicated in Figure 3.25. This problem is similar to the first one.

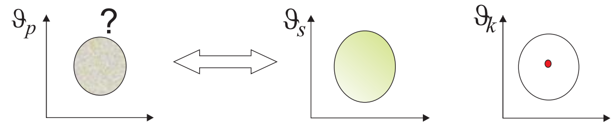


Fig. 3.25. Design of a robust controller.

4. Obtaining a set $\boldsymbol{K}$ of robust controllers: Given an uncertain plant $G(s, \boldsymbol{q})$, the variation domain of the plant parameters, $\boldsymbol{Q}$, and the controller structure, find the set of robust controllers $\boldsymbol{K}$ so that $C(s, \boldsymbol{K})$ fulfills the robustness specifications:

$$f_i(\alpha, \boldsymbol{Q}, \boldsymbol{K}) > 0 \tag{3.109}$$

the last problem corresponds with finding the controllers space under parametric uncertainty.

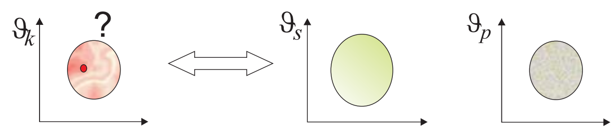


Fig. 3.26. Obtaining a set ϑ_k of robust controllers.

In **Vehí et al. (2000)** the robustness analisys of predictive controllers via modal intervals was studied.

In **Bondia et al. (2006)**, the Modal Interval Arithmetic was applied to the robust controller design under fuzzy pole-placement specifications. The design problem was raised as a fuzzy set inclusion problem between a set of closed-loop fuzzy specifications and a closed-loop characteristic polynomial. The closed-loop controller parameters were found so that closed-loop image of the fuzzy plant are including within the closed-loop fuzzy specifications.

In **Sainz et al. (2008)** a new approach to the solution of continuous unconstrained and constrained minimax problems over real was introduced, using tools based on modal intervals. For instance, if $z = f(x_1, .., x_n)$ is a continuous function of R^n to R defined in a n-dimensional interval domain $\boldsymbol{X} = \boldsymbol{U} \times \boldsymbol{V}$,

- the unconstrained minimax problem consists of finding a point $x^*_{minimax} \in \boldsymbol{U} \times \boldsymbol{V}$ such that

$$f(x^*_{minimax}) = \underbrace{min}_{u \in U} \underbrace{max}_{v \in V} f(\boldsymbol{x}) \tag{3.110}$$

together with the minimax value $f(x^*_{minimax})$.

- the constrained minimax problem consists of finding $x^*_{minimax}$ such that

$$f(x^*_{minimax}) = \underbrace{min}_{u \in U} \underbrace{max}_{v \in V} f(\boldsymbol{x}), \tag{3.111}$$

subject to the constraints

$$g_r(x^*_{minimax}) \leq 0 \; (r = 1, ..., m), \tag{3.112}$$

where g_r are continuous functions defined in $\boldsymbol{X}$.

3.4 Constraints Satisfaction Problems

Constraint Satisfaction Problems (CSP) emerged in the field of artificial intelligence. In the Tsang's Book **Tsang (1993)** the foundations of Constraint Satisfaction Problems and solution techniques can be consulted. Three techniques to solve CSP are mentioned: problem reduction, search and solution synthesis.

Problem reduction is a class of techniques to transform a continuous real CSP into problems which are easier to solve, reducing the size of the domains of the variables and constraints in the problems.

In solutions search, the basic operation is to assign a value from the parameters space to the variables and to verify if all the constraints are satisfied. If all the constraints are satisfied, the value is a solution. If a constraints is violated, a new value is assigned to the variables and one evaluates the constraints again.

Solution synthesis, are algorithms which explore multiple branches simultaneously to search solutions.

Shary **Shary (2002)**, defined a numerical Constraint Satisfaction Problem as a triple $\mathbf{CSP} = (\mathbf{x}, \mathbf{D}, \mathbf{C}(\mathbf{x}))$ where

1. a set of numeric variables $\mathbf{x} = x_1, ..., x_n$,
2. a set of domains $\mathbf{D} = D_1, ..., D_n$ where D_i, a set of intervals, is the domain associated with the variable x_i,
3. a set of constraints $C(\mathbf{x}) = C_1(\mathbf{x}), ..., C_m(\mathbf{x})$ where a constraint $C_i(\mathbf{x})$ is determined by any numerical relation (equation, inequality, inclusion, etc.) linking a set of variables under consideration.

A solution to a numeric constraint satisfaction problem is an instantiation of the variables of $\mathbf{x}$ for which both inclusion in the associated domains and all the constrains of $\mathbf{C}(\mathbf{x})$ are satisfied.

A Set Inversion Via Interval Analysis (SIVIA) Algorithm was introduced by Jaulin and Walter **Jaulin and Walter (2004)**, is well known as a paradigm of interval analysis, is also suited for approximating solution sets of the form (3.113) by means of sub pavings (sets of non overlapping boxes).

$$\Sigma = \{\boldsymbol{x} \in \boldsymbol{X}^{'} | f(\boldsymbol{x}) \leq 0\} \tag{3.113}$$

where f is a continuous function from R^n to R^m. The SIVIA algorithm combines branch-and-bound techniques with the following two rules to determine if a box $\boldsymbol{x}$ is contained in the solution set Σ or if $\boldsymbol{x}$ does not intersect with Σ.

$$Rule\,1: \ \forall(\boldsymbol{x} \in \boldsymbol{X}^{'}) f(\boldsymbol{x}) \leq 0 \Longleftrightarrow \boldsymbol{X}^{'} \subseteq \Sigma \tag{3.114}$$

The rule 1 is used to prove that a box $\boldsymbol{X}^{'}$ is contained in the solution set.

$$Rule\,2: \ \forall(\boldsymbol{x} \in \boldsymbol{X}^{'}) \neg(f(\boldsymbol{x}) \leq 0) \Longleftrightarrow \boldsymbol{X}^{'} \subseteq \overline{\Sigma} \tag{3.115}$$

The rule 2, is used to prove that a box $\boldsymbol{X}^{'}$ does not belong to the solution set. $\overline{\Sigma}$ is defined as a complementary of the solution set. The SIVIA algorithm is able to solve a subset of QCSPs, that is to say, CSPs. An example where we can apply the SIVIA algorithm is given below.

Example 3.7.Given a constrained function $x_1u - x_2v^2sin(x_1) > 0$ and intervals for $x_1 \in [-10, 10]$, $x_2 \in [-10, 10]$, $v \in [-2, 2]$ and $u \in [-1, 1]$ we desire to find value sets for $\boldsymbol{x} = \{x_1, x_2, v, u\}$, such that the constrained function $x_1u - x_2v^2sin(x_1) > 0$ is greater than zero. The basic approach of the SIVIA algorithm is to divide the parameters space x_1, x_2, v and u and prove for each partition if the constraint is verified. If the constraint is verified then the partition is stored in a solution set, in other cases the partition is devided and the process is repeated until certain precision is reached on the division.

The solution set, can be expressed as follows:

$$\Sigma = \{\boldsymbol{x} \in \boldsymbol{X}^{'} | x_1u - x_2v^2sin(x_1) > 0\} \tag{3.116}$$

Being $\boldsymbol{X}^{'} = \{\boldsymbol{X}_1^{'}, \boldsymbol{X}_2^{'}, \boldsymbol{V}^{'}, \boldsymbol{U}^{'}\}$ the set of domains of variables $\boldsymbol{x} = \{x_1, x_2, v, u\}$. We found the results indicated in Figures 3.27 and 3.28. In Figure 3.27 x_1 and x_2 is plotted with v and in Figure 3.28 x_1 and x_2 is plotted with u.

3.5 Quantified Constraints Satisfaction Problems

A Quantified Constraint (QC) **Herrero et al. (2005)**, is an algebraic expression over the real numbers which contains quantifiers $(\exists, \forall)$, predicate

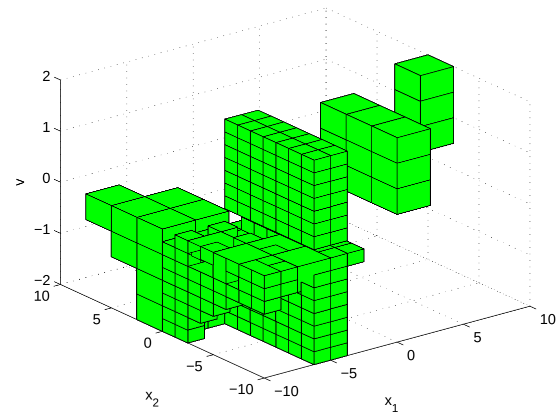


Fig. 3.27. Solution set x_1, x_2, v from Constraint Satisfaction Problem (CSP) considering intervals $u \in [-1, 1]$, $v \in [-2, 2]$, $x_1 \in [-10, 10]$, $x_2 \in [-10, 10]$ and the constrained function $(x_1 u - x_2 v^2 sin(x_1) > 0)$.

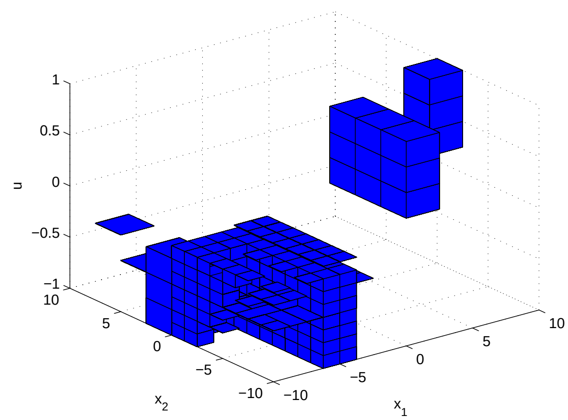


Fig. 3.28. Solution set x_1, x_2, u from Constraint Satisfaction Problem (CSP) considering intervals $u \in [-1, 1]$, $v \in [-2, 2]$, $x_1 \in [-10, 10]$ and $x_2 \in [-10, 10]$ and constrained function $(x_1 u - x_2 v^2 sin(x_1) > 0)$.

symbols ($e.g., =, <, \leq$), function symbols ($e.g., +, -, x, sin, exp$), constants and variables $x = x_1, ..., x_n$ ranging over real domains $\mathbf{D} = \boldsymbol{D}_1, ..., \boldsymbol{D}_n$.

An example of a quantified constraint is the following one,

$$\forall(x \in R)(x^4 + px^2 + qx + r \geq 0), \tag{3.117}$$

where x is an universally quantified variable $\forall$ and p, q and r are free variables.

In a Constraint Satisfaction Problem (CSP) all variables are existentially quantified. CSP is a particular case of a QCSP that can be used to model problems containing uncertainty. The uncertainty is indicated with the universally quantified variables. Universal variables are used to model actions or events which are uncertain or are not in our control. In a QCSP we try to find solution sets, searching values in the existential variables for all possible sequences of instantiations for the universal variables so that all the constraints in the problem are satisfied **Gent et al. 2008**.

Supposing that the constraints **C(x,p)** depend on some parameters $p_1, p_2, ..., p_l$ in which it is only known that they belong to some intervals $P_1, P_2, ..., P_l$. Moreover, these parameters have an associated quantifier $Q \in \forall, \exists$. Taking into account the dual character of interval uncertainty, the most general definition of the set of solutions such Quantified Constraint Satisfaction Problem (**QCSP**) will have the form

$$\sum = \{\boldsymbol{x} \in D | (Q_1 p_{\sigma 1} \in \boldsymbol{P}_{\sigma 1}) ... (Q_l p_{\sigma l} \in \boldsymbol{P}_{\sigma l}) C(\boldsymbol{x}, \boldsymbol{p})\}, \tag{3.118}$$

where

- each Q_i is logical quantifier $\forall$ or $\exists$,
- $\mathbf{p} = \{p_1, p_2, ..., p_l\}$ is the set of parameters of the constraints system considered.
- $\mathbf{P} = \{P_1, P_2, ..., P_l\}$ is a set of intervals containing the possible values of $\mathbf{p}$,
- $\sigma = (\sigma_1, \sigma_2, ..., \sigma_l)$ is a permutation of the numbers $1, ..., l$.

The sets of the form (3.118) will be referred as quantified solution sets to the Quantified Constraints Satisfaction Problem.

When the set of solutions to solve are of the form (3.118) the SIVIA algorithm is not able to solve this kind of problems in a direct way . The problem of characterizing the sets of the form (3.118) will be referred as quantified set inversion (QSI) **Herrero et al. (2005)**. Let us develop two examples to explain the QSI algorithm.

Example 3.8.Given the constrained function $x_1 u - x_2 v^2 sin(x_1) > 0$, we desire to find the set of values for x_1 and x_2 for all u within interval $u \in [-1, 1]$ such that exists a set of values within $v \in [-2, 2]$ and the constrained function is greater than zero. Quantified constrained function can be expressed as follows:

$$\forall(u \in [-1,1]^{'})\exists(v \in [-2,2]^{'})(x_1 u - x_2 v^2 sin(x_1) > 0) \qquad (3.119)$$

In this kind of QCSPs, when a parameter is universally quantified means that we can use all the range of the parameter in the algorithm to determine solution sets. If the parameter is existentially quantified means that we will search solution sets within interval such that the constraints are satisfied. We also search solution sets within the free variables (x_1, x_2) such that the constraints are satisfied. To apply the above approach to the QSI algorithm, we found solution sets to x_1 and x_2 and solution regions to v as it is indicated in Figures 3.29 and 3.30 respectively.

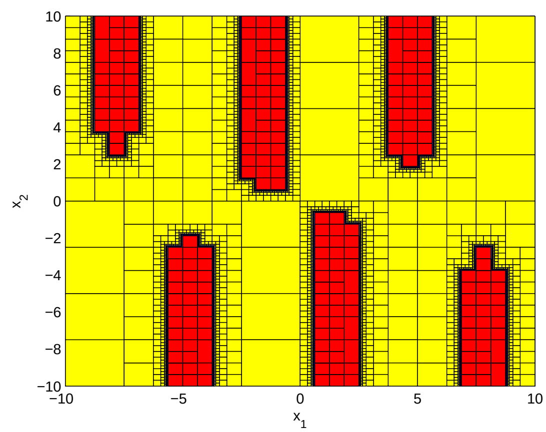


Fig. 3.29. Solution set x_1, x_2 from Quantified Constraint Satisfaction Problem (QCSP) $\forall(u \in [-1,1]^{'})\exists(v \in [-2,2]^{'})(x_1 u - x_2 v^2 sin(x_1) > 0)$. The red boxes represent solution set, yellow boxes are outside of the solution set and black boxes are undefined.

Example 3.9.From constrained function $x_1 u - x_2 v^2 sin(x_1) > 0$, we desire to find the set of values for x_1 and x_2 for all u within interval $u \in [-1,1]$ and for all $v \in [-2,2]$ such the constrained function is greater than zero. Quantified constrained function can be expressed as follows:

$$\forall(u \in [-1,1]^{'})\forall(v \in [-2,2]^{'})(x_1 u - x_2 v^2 sin(x_1) > 0) \qquad (3.120)$$

to treat of solve the previous problem, we do not find solutions. If the modality of a parameter is changed from existential to universal as it is the

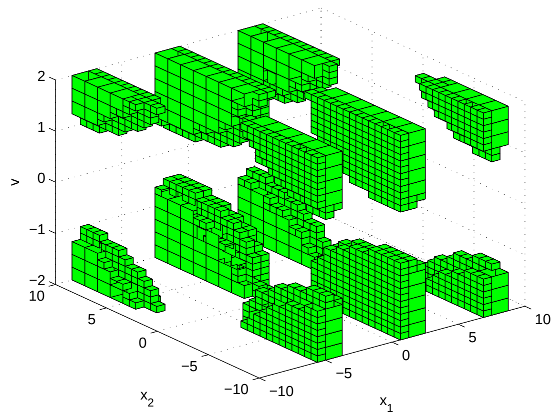


Fig. 3.30. Solution set x_1, x_2, v from Quantified Constraint Satisfaction Problem (QCSP)$\forall(u \in [-1,1]^{'})\exists(v \in [-2,2]^{'})(x_1 u - x_2 v^2 sin(x_1) > 0)$.

case of v , then the problem is more restrictive. In Figure 3.30 we can see that within v there are holes and it is not possible to satisfy the constraints for all the range of v and u and for all the set of possible values of x_1 and x_2 within the intervals $x_1 \in [-10, 10]$ and $x_2 \in [-10, 10]$ respectively.

In particular, quantified solution sets where universal quantifiers are constrained to precede existential quantifiers are called ∀∃- solution set **Shary (2002)**;**Goldsztejn and Chabert (2006)**.

Recently, a technique to solve QCSPs where the existentially quantified variables precede to the universally quantified variables was proposed by Goldsztejn **Goldsztejn et al. (2009)**. The approach consists in finding all the possible values for the existentially quantified variables that satisfy the constraints. A point of the universally quantified variables was used in the constraints and the QCSP is transformed to a classical CSP equivalent and the constraints are expressed solely with the existentially quantified variables.

Formaly, the form of QCSPs where existential quantifiers precede to the universal quantifiers is expressed as follows:

$$\exists(x \in \boldsymbol{x}^{'}), \forall(y \in \boldsymbol{y}^{'}), c_1(x,y)\wedge, ..., \wedge c_p(x,y) \tag{3.121}$$

where $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_m)$ denote vectors of variables, $\boldsymbol{x}^{'} = (\boldsymbol{x}_1^{'}, ..., \boldsymbol{x}_n^{'})$ and $\boldsymbol{y}^{'} = (\boldsymbol{y}_1^{'}, ..., \boldsymbol{y}_m^{'})$ represent vectors of intervals over continuous domains and the constraints c_i are inequalities of the form $f_i(x,y) \leq 0$. We want to find solutions within the existential variables that satisfy the constraints. The solution set of (3.121) is defined as follows:

$$\sum\nolimits_{\exists\forall} := \{x \in \boldsymbol{x} : \forall(y \in \boldsymbol{y}^{'})\ c_i(x,y)\wedge, ..., \wedge c_p(x,y)\} \tag{3.122}$$

Example 3.10.Let us consider the following QCSP

$$\exists(x \in \boldsymbol{x}^{'}), \forall(y \in \boldsymbol{y}^{'}), f(x,y) \leq 0 \tag{3.123}$$

being $f(x,y) = 5y - x - y^3$, $\boldsymbol{y}^{'} = [0,1]$ and $\boldsymbol{x}^{'} = [0,15]$, determine the values of x that satisfy the constraint. We are going to consider a point of y for instance $y = [1,1]$. If we replace this point in the function $f(x,y)$,

$$\begin{aligned} f(x,y) = 5 * [1,1] - x - [1,1] * [1,1] * [1,1] \leq 0 \\ f(x,y) = [5,5] - x - [1,1] \leq 0 \end{aligned} \tag{3.124}$$

The QCSP is transformed to a CSP as follows:

$$\exists(x \in \boldsymbol{x}^{'}), f(x) \leq 0 \tag{3.125}$$

being $f(x) = 5 - x - 1$. In the constraint $f(x) \leq 0$ we can see that the solution set of x that satisfies the constraint is $x = [4,15]$.

Shary **Shary (2002)** solves static control problems raised as QCSP's.

Consider a linear interval system of the following form

$$F(a,x) = b \tag{3.126}$$

being $x \in R^n$ a real vector, $a \in R^l$ the system inputs and $b \in R^m$ the system outputs. The inputs to the system are divided in two groups

- perturbations $a_1, ..., a_r$, which vary within intervals $\mathbf{a}_1, ..., \mathbf{a}_r$ independently of our will, and
- controls $a_{r+1}, ..., a_l$ which we can choose from intervals $\mathbf{a}_{r+1}, ..., \mathbf{a}_l$.

The set of all the system outputs are divided in:

- the components $b_1, b_2, ..., b_s$ that we must be able to transform to any values from prescribed attainability intervals $\mathbf{b}_1, ..., \mathbf{b}_s$, and
- the components $b_{s+1}, ..., b_m$ that must certainly fall into some intervals $\mathbf{b}_{s+1}, ..., \mathbf{b}_m$

The outputs of the first type are considered as controlled outputs while the outputs of the second type as stabilized outputs. In Figure 3.31 the structural scheme is depicted.

With respect to the system inputs, the logical quantifiers are assigned taking into account the following considerations

1. The inputs that are not under our control, being external uncontrolled disturbances, correspond to the quantifier ($\forall$), and

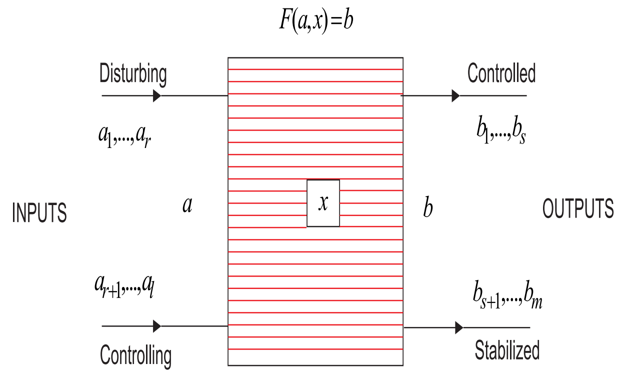


Fig. 3.31. Structural scheme of a static control system.

2. The inputs that we are able to vary within prescribed intervals or are under our control, correspond to the quantifier (∃).

With respect to the system outputs

1. Stabilization regions of the system in which it is required to ensure its functioning irrespective of values of the disturbances, correspond to the quantifier (∃), and
2. Attainability sets of the system whose every element is covered as the result of an appropriate choice of the controlled factors, correspond to the quantifier (∀).

From linear interval system (3.126), find the set of all states x from its inputs a and outputs b can be expressed as follows. Given the control inputs inside the intervals $a_{r+1} \in \mathbf{a}'_{r+1}, ..., a_l \in \mathbf{a}'_l$. Consider any perturbations $a_1 \in \mathbf{a}'_1, ..., a_r \in \mathbf{a}'_r$ such that for any a priori given output values $b_1 \in \mathbf{b}'_1, ..., b_s \in \mathbf{b}'_s$ the response of the system $F(a, x)$ would be exactly equal to $b_1, ..., b_s$ in the controlled outputs and would be inside $\mathtt{b}'_{s+1}, ..., \mathtt{b}'_m$ in stabilized outputs. Under this setting the problem of finding the region of the state space where all these constraints apply can be expressed, using universal (∀) and existential (∃) quantifiers. The first order predicate calculus corresponds to:

for any $a_1 \in \mathbf{a}'_1, ..., a_r \in \mathbf{a}'_r$ and for any $b_1 \in \mathbf{b}'_1, ..., b_s \in \mathbf{b}'_s$ there exist $a_{r+1} \in \mathbf{a}'_{r+1}, ..., a_l \in \mathbf{a}'_l$ such that $F_1(a, x), ..., F_s(a, x)$ are equal to $b_1, ..., b_s$ and $F_{s+1}(a, x), ..., F_m(a, x)$ are inside $\mathbf{b}'_{s+1}, ..., \mathbf{b}'_m$.

This can be equivalently rewritten with the following predicate (logical formula):

$$\forall(a_1 \in \mathbf{a}'_1)...\forall(a_r \in \mathbf{a}'_r)\forall(b_1 \in \mathbf{b}'_1)...\forall(b_s \in \mathbf{b}'_s)$$
$$\exists(a_{r+1} \in \mathbf{a}'_{r+1})...\exists(a_l \in \mathbf{a}'_l)\exists(b_{s+1} \in \mathbf{b}'_{s+1})...\exists(b_m \in \mathbf{b}'_m)(F(a, x) = b)$$

(3.127)

That is to say, the set of all states x satisfying the constraints of the problem is described as follows

$$\sum_{\forall\exists} := \{x \in R^n | \forall(a_1 \in \mathbf{a}_1^{'})...\forall(a_r \in \mathbf{a}_r^{'})\forall(b_1 \in \mathbf{b}_1^{'})...\forall(b_s \in \mathbf{b}_s^{'})$$
$$\exists(a_{r+1} \in \mathbf{a}_{r+1}^{'})...\exists(a_l \in \mathbf{a}_l^{'})\exists(b_{s+1} \in \mathbf{b}_{s+1}^{'})...\exists(b_m \in \mathbf{b}_m^{'})(F(a,x) = b)\} \quad (3.128)$$

An example is developed as follows:

Example 3.11.Consider the interval linear system (3.129) **Shary (2002)**.

$$\begin{array}{l} a_{11}x_1 + a_{12}x_2 \subseteq b_1 \\ a_{21}x_1 + a_{22}x_2 \subseteq b_2 \end{array} \quad (3.129)$$

In this example we included the inclusion operator $\subseteq$. Consider the intervals indicated in the following table

Parameters	Intervals
$\boldsymbol{a}_{11}$	$[2, 4]$
$\boldsymbol{a}_{12}$	$[-2, 1]$
$\boldsymbol{a}_{21}$	$[-1, 2]$
$\boldsymbol{a}_{22}$	$[2, 4]$
$\boldsymbol{b}_1$	$[-2, 2]$
$\boldsymbol{b}_2$	$[-2, 2]$
$\boldsymbol{x}_1$	$[-1, 1]$
$\boldsymbol{x}_2$	$[-1, 1]$

A Quantified Constraints Satisfaction Problem can be raised as follows:

1. Find $\Sigma_{\forall\exists}$-solution sets to free variables x_1 and x_2 from interval linear system (3.129) such that for any a_{11}, a_{12} and a_{21} within the intervals $\boldsymbol{a}_{11}^{'}$, $\boldsymbol{a}_{12}^{'}$ and $\boldsymbol{a}_{21}^{'}$, there exist some values to a_{22}, b_1 and b_2 within the intervals $\boldsymbol{a}_{22}^{'}$, $\boldsymbol{b}_1^{'}$ and $\boldsymbol{b}_2^{'}$ and that constraints are satisfied. So, the kind of quantifiers assigned to the parameters correspond to the indicated in Table 3.1.
   The previous QCSP can be expressed of the following form:

$$\begin{array}{rl} \Sigma_{\forall\exists} = \{x_1 \times x_2 | & \forall(a_{11} \in \boldsymbol{a}_{11}^{'})\forall(a_{12} \in \boldsymbol{a}_{12}^{'})\forall(a_{21} \in \boldsymbol{a}_{21}^{'}) \\ & \exists(a_{22} \in \boldsymbol{a}_{22}^{'})\exists(b_1 \in \boldsymbol{b}_1^{'})\exists(b_2 \in \boldsymbol{b}_2^{'}) \\ & a_{11}x_1 + a_{12}x_2 \subseteq b_1 \wedge \\ & a_{21}x_1 + a_{22}x_2 \subseteq b_2\} \end{array} \quad (3.130)$$

   In Figure (3.32) the results are indicated.

Parameters	quantification
a_{11}	$\forall$
a_{12}	$\forall$
a_{21}	$\forall$
a_{22}	$\exists$
b_1	$\exists$
b_2	$\exists$

Table 3.1. Quantifiers assigned to the parameters to the problem one

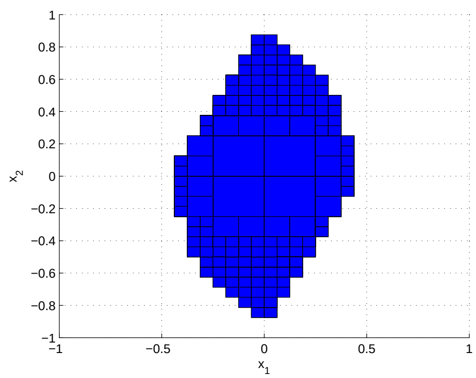


Fig. 3.32. $\Sigma_{\forall\exists}$-solution sets, obtained with quantifiers of the Table (3.1).

2. Now, let us exchange the kind of quantification to a_{12} from universal ($\forall$) to existential ($\exists$) keeping unchanged the kind of quantification of the others parameter as it is indicated in Table 3.2.

Parameters	quantification
a_{11}	$\forall$
a_{12}	$\exists$
a_{21}	$\forall$
a_{22}	$\exists$
b_1	$\exists$
b_2	$\exists$

Table 3.2. Quantifiers assigned to the parameters to the problem two.

The solution set for this case is expressed as:

$$\begin{aligned}\Sigma_{\forall\exists} = \{x_1 \times x_2 | &\forall(a_{11} \in \boldsymbol{a}_{11}^{'})\forall(a_{21} \in \boldsymbol{a}_{21}^{'})\exists(a_{12} \in \boldsymbol{a}_{12}^{'}) \\ &\exists(a_{22} \in \boldsymbol{a}_{22}^{'})\exists(b_1 \in \boldsymbol{b}_1^{'})\exists(b_2 \in \boldsymbol{b}_2^{'}) \\ &a_{11}x_1 + a_{12}x_2 \subseteq b_1 \wedge \\ &a_{21}x_1 + a_{22}x_2 \subseteq b_2\}\end{aligned} \tag{3.131}$$

When we exchange the kind of quantification from universal ($\forall$) to existential ($\exists$) as is the case of the parameter a_{12} we find solution sets more extended. In Figure (3.33) we observe that solution sets includes the paving obtained with quantifications from Table 3.1.

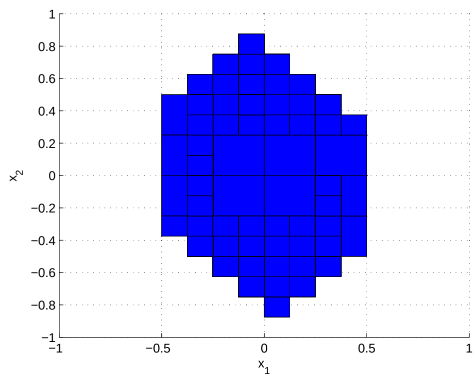


Fig. 3.33. $\Sigma_{\forall\exists}$-solution, obtained with quantifiers of the Table 3.2.

An example of (QCSP) directly related to the thesis is solving with Quantified Set Inversion Algorithms (QSIA) an inclusion function $f(\vartheta_p, \vartheta_k) \subseteq \gamma(\vartheta_s)$ being ϑ_k a set of parameters representing a family of controllers, ϑ_p a family of plants and ϑ_s a set of desired specifications. We are interested in obtaining solution sets referred to controllers, attainable specifications and maximum allowable uncertainty by the controllers. The problems are:

1. To find the maximum admissible uncertainty ($\vartheta_p(\forall)$) by a nominal controller ($\vartheta_k(\exists)$) ensuring that some specifications are met ($\vartheta_s(\exists)$) and the constraints are satisfied. The previous problem is expressed as follows:

$$\forall(\vartheta_p \in \boldsymbol{\vartheta}_p^{'})\exists(\vartheta_k \in \boldsymbol{\vartheta}_k^{'})\exists(\vartheta_s \in \boldsymbol{\vartheta}_s^{'})(f(\vartheta_k, \vartheta_p) \subseteq \gamma(\vartheta_s)) \tag{3.132}$$

2. To determine the family of controllers ($\vartheta_k(\forall)$) that could ensure that some specification are met ($\vartheta_s(\exists)$) and the constraints are satisfied under parametric uncertainty in the plant ($\vartheta_p(\forall)$). The semantic is as follows:

$$\forall(\vartheta_k \in \boldsymbol{\vartheta}_k^{'})\forall(\vartheta_p \in \boldsymbol{\vartheta}_p^{'})\exists(\vartheta_s \in \boldsymbol{\vartheta}_s^{'})(f(\vartheta_k, \vartheta_p) \subseteq \gamma(\vartheta_s)) \tag{3.133}$$

3. To determine the achievable specifications ($\vartheta_s(\forall)$) by a nominal controller ($\vartheta_k(\exists)$), satisfying the constraints under parametric uncertainty in the plant ($\vartheta_p(\forall)$). The semantic corresponds to:

$$\forall(\vartheta_s \in \boldsymbol{\vartheta}_s^{'})\forall(\vartheta_p \in \boldsymbol{\vartheta}_p^{'})\exists(\vartheta_k \in \boldsymbol{\vartheta}_k^{'})(f(\vartheta_k, \vartheta_p) \subseteq \gamma(\vartheta_s)) \tag{3.134}$$

3.6 Conclusions

This Chapter has presented the most important properties of the flatness theory, flatness-based approach on dynamic optimization and flatness in the context of parametric uncertainty. It has been determined that the state variables and control signals are expressed in terms of specification parameters and process parameters when the admissible nominal trajectories of the flat outputs and its derivatives are expressed in terms of the specification parameters. The problems of robust control, have been proposed as Quantified Constraints Satisfaction Problems. Constraints are expressed by using set inclusion operators and quantified variables.

4 Approach to robust possibilistic control of nonlinear flat systems

In this Chapter we present basic aspects of possibility levels as well as the approach of robust possibilistic control expressed as set inclusion problems. A formal presentation on how to propose Quantified Constraints Satisfaction Problems for nonlinear flat systems is given.

4.1 Levels of possibility

Possibility theory emerged as a natural tool for modeling and dealing with uncertainty related to express knowledge in a natural language and represented by fuzzy propositions **Zadeh (1999)**; **Dubois et al. (2003)**. In possibility theory, the available information is represented by means of possibility distributions. On an universe of discource U with elements $u \in U$, some fuzzy subsets A_i can be represented as it is indicated in Figure 4.1. Where $\mu_{A_i},\ i = 1, 2, 3, 4$ are member functions for each fuzzy subset.

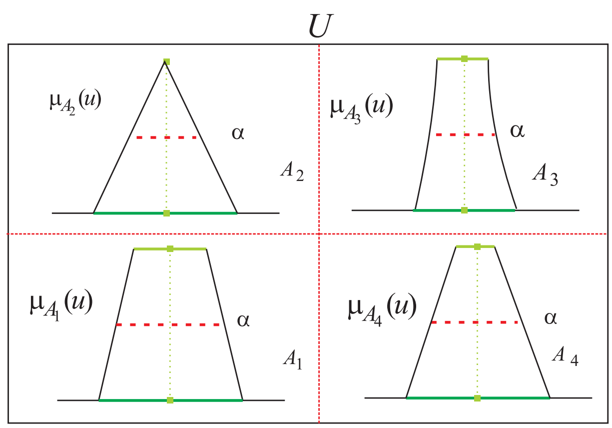


Fig. 4.1. Fuzzy subsets.

A fuzzy subset A_i can be assigned to a variable X, this denomination can be written as:

$$X = \mu_{A_i}(u) \tag{4.1}$$

where $\mu_{A_i}(u)$ is interpreted as the degree to which the fuzzy subset A_i is satisfied.

In 1978 Zadeh **Zadeh (1978)** proposed to represent pieces of information by means of possibility distributions. Elementary propositions such "X is A_i" is a fuzzy predicate and X is the variable (ranging on a domain U). The main role of possibility distributions is to discard states of affairs inconsistent with the available knowledge. Indeed, $\pi_x(u) = 0$ means that the assignment $X = u$ is totally excluded if the statement "X is A_i" is taken for granted.

If μ_{A_i} denotes the membership function of the fuzzy set A_i on U, and π_x denotes a possibility distribution on U, Zadeh proposed the equality $\pi_x = \mu_{A_i}$ expressing the statement "X is A_i" and induces to a possibility distribution that can be equated with μ_{A_i}.

The definition of $\pi_x(u)$ implies that the degree of possibility may be any number in the interval $[0, 1]$ rather than just 0 or 1. In this connection, it should be noted that the existence of possibility intermediate degrees is implicit where commonly encountered propositions as "There is a slight possibility that Marilyn is very rich", "It is quite possible that Jean-Paul will be promoted", "It is almost impossible to find a needle in a haystack", etc.

The possibility distribution functions can be expressed by means of fuzzy sets. In fuzzy sets, one assigns a so-called membership functions $\mu_{A_i}(u)$, taking values in $[0, 1]$ to each element u of a given domain U.

A membership function can take any arbitrary form. A representation that combines simplicity and expressiveness is the trapezoidal one.

If the domain U corresponds to the space of values of a given parameter, the membership function assigns grades of possibility to the different values that the parameter can potentially take.

Formally, if the discourse universe, U, it is discrete and finite, and the membership function is denoted as $\mu_{A_i} = \mu_{\tilde{A}}$, then the fuzzy set $A_i = \tilde{A}$ it is represented as the following form **Bondia (2002)**

$$\tilde{A} = \textstyle\sum_{i=1}^{m} \mu_{\tilde{A}}(u_i)/u_i \tag{4.2}$$

The height of a fuzzy set $\tilde{A}$ is defined as the maximum value of the membership function $\mu_{\tilde{A}}$, ie:

$$hgt(\tilde{A}) := \underbrace{sup}_{u \in U} \mu_{\tilde{A}}(u) \tag{4.3}$$

The core of a fuzzy set $\tilde{A}$ is the set of U elements with membership function values equal to the unit, ie

$$core(\tilde{A}) := \{u \in U | \mu_{\tilde{A}}(u) = 1\} \tag{4.4}$$

The support of a fuzzy set $\tilde{A}$ is the set of U elements with membership function values non-zero, ie

$$supp(\tilde{A}) := \{u \in U | \mu_{\tilde{A}}(u) > 0\} \tag{4.5}$$

Example 4.1.A fuzzy set $\tilde{A}$ with membership function $\pi_{\tilde{A}}(u) = \mu_{\tilde{A}}(u)$ of trapezoidal type is indicated in Figure 4.2

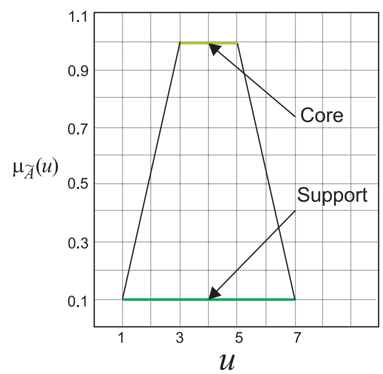


Fig. 4.2. Trapezoidal fuzzy set.

Some discrete elements of the membership function are listed in the following table,

u	$\pi_{\tilde{A}}(u)$	$\mu_{\tilde{A}}(u)$
1	0.1	0.1
2	0.53	0.53
3	1	1
4	1	1
5	1	1
6	0.53	0.53
7	0.1	0.1

In Figure 4.2, it is verified that $core(\tilde{A}) := \{u \in U | \mu_{\tilde{A}}(u) = 1\}$ for $\{3 \leq u \leq 5\}$ and $supp(\tilde{A}) := \{u \in U | \mu_{\tilde{A}}(u) > 0\}$ for $\{1 \leq u \leq 7\}$.

A fuzzy set $\tilde{A}$ can be represented by the set of all its cuts **Terano et al. (1992)**; **Bondia (2002)**, as a stack of closed intervals of the form:

$$\tilde{A} = \bigcup_{\alpha \in (0,1]} \alpha[A_\alpha^-, A_\alpha^+] = \bigcup_{\alpha \in (0,1]} \alpha[\tilde{A}]_\alpha \tag{4.6}$$

where $\alpha[A_\alpha^-, A_\alpha^+]$ represents the fuzzy set whose support is the α-cut $[A_\alpha^-, A_\alpha^+]$, denoted as $[\tilde{A}]_\alpha$ and the level of constant membership equal to α.

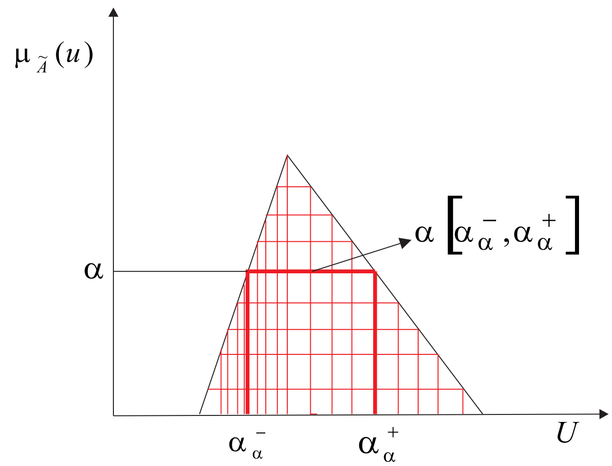


Fig. 4.3. α-cuts.

4.2 The robust possibilistic control approach

The main objetive of robust control is to develop feedback control laws that are robust against plant model uncertainties and changes in dynamic conditions. A system is robustly stable when the closed-loop is stable for any chosen plant within the specified uncertainty set, and a system has robust performance if the closed-loop system satisfies the performance specifications for any plant model within the specified uncertainty description.

The robust possibilistic control for fuzzy plants was introduced by Bondia **Bondia et al. 2005**. A fuzzy set in a linear plant space was used to associate a possibility to each member in the family of plants. A controller was designed with the purpose of achieving a fuzzy specification set. The core of the fuzzy specifications set was denominated as the hard specifications to be achieved by the more possible plants, while the support was denominated as the soft specifications representating the minimum specifications to be achieved for all the plants in the family.

Concerning the design problem, given a fuzzy plant $\tilde{P}$ and a set of fuzzy specifications $\tilde{S}$, the family of controllers $\tilde{K}$ was obtained by solving a fuzzy set inclusion problem of the form

$$J(\tilde{P}, \tilde{K}) \subseteq \tilde{S} \tag{4.7}$$

being $J(\tilde{P}, \tilde{K})$ a fuzzy set of the different performances achieved by the family of controllers K in terms of the possibility of the different plants in the family.

The inclusion in (4.7) corresponds to the fuzzy sets inclusion ($\tilde{A} \subseteq \tilde{B} \leftrightarrow \mu_{\tilde{A}(x)} \leq \mu_{\tilde{B}}(x), \forall x$).

In terms of α-cuts denoting as $\tilde{P}_\alpha$ and $\tilde{S}_\alpha$ to the appropriate α-cuts, the equation (4.7) corresponds to:

$$J(\tilde{P}_\alpha, \tilde{K}_\alpha) \subseteq \tilde{S}_\alpha \tag{4.8}$$

In this form, the family of controllers applies a set of fuzzy plants and specifications, getting good performance for more possible plants(core), and guaranteeing a minimum performance for any possible plant.

The solution of the equation (4.7) requires a discretization at the membership level α. In many practical cases the definition of only two levels can be perfectly feasible considering only the worst and best plant family cases and a set of hard and soft specifications.

In this thesis the inclusion relations are between specification spaces, states, controllers and plants as it was specified in equation (3.44). In order to guarantee the fulfillment of specifications we will quantify the variables involved in the process. The technique to carry out these quantifications and the formal proposal of the different problems from the robust control point of view will be presented in the following sections.

4.3 Quantified set inversion

One way of solving a QCSP is through the characterization of its solution set by means of a Quantified Sets Inversion Algorithm **Herrero et al. (2005)**.

We will focus in proposing and solving Quantified Constraints Satisfaction Problems considering interval functions f^* that contain inclusion relations of the form

$$f^*(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{V}) \subseteq \boldsymbol{Z} \tag{4.9}$$

where f^* is an interval function, $\boldsymbol{X}, \boldsymbol{U}$ are proper components and $\boldsymbol{V}$ improper components. $\boldsymbol{Z}$ is a given interval. We are going to find all the solution sets $\boldsymbol{X}$ such that the constraints are fulfilled taking into account different types quantification of the variables.

That is to say, to find solution sets related to the $*$- semantic:

$$\forall(\boldsymbol{x} \in \boldsymbol{X}^{'})\forall(\boldsymbol{u} \in \boldsymbol{U}^{'})\exists(\boldsymbol{v} \in \boldsymbol{V}^{'})f^{*}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}) \subseteq \boldsymbol{Z} \tag{4.10}$$

The solution set $\boldsymbol{X}$ is represented in equation(4.11)

$$Rule\ 1:\ \Sigma_{\forall\exists} = \{\boldsymbol{x} \in \boldsymbol{X}^{'} | \forall(\boldsymbol{u} \in \boldsymbol{U}^{'})\exists(\boldsymbol{v} \in \boldsymbol{V}^{'})f^{*}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}) \subseteq \boldsymbol{Z}\} \tag{4.11}$$

when the interval inclusion is fulfilled by an outer approximation of the $*$- semantic extension as it was explained in equation (3.101), then we can specify: $f^{*}(\boldsymbol{X},\boldsymbol{U},\boldsymbol{V}) \Rightarrow Outer(f^{*}(\boldsymbol{X},\boldsymbol{U},\boldsymbol{V}))$. So, $\Sigma_{\forall\exists}$ corresponds to the following expression:

$$\Sigma_{\forall\exists} = \{\boldsymbol{x} \in \boldsymbol{X}^{'} | \forall(\boldsymbol{u} \in \boldsymbol{U}^{'})\exists(\boldsymbol{v} \in \boldsymbol{V}^{'})(Outer(f^{*}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}))) \subseteq \boldsymbol{Z}\} \tag{4.12}$$

We will evaluate set of boxes $\boldsymbol{X}$ to verify that they are not inside solution sets. If solution sets from equation (4.11) exist, and if we use a solution box of the resulting set, then the kind of quantification for the box $\boldsymbol{X}$ can be existential ($\exists$) and the corresponding semantics is:

$$\forall(\boldsymbol{u} \in \boldsymbol{U}^{'})\exists(\boldsymbol{x} \in \boldsymbol{X}^{'})\exists(\boldsymbol{v} \in \boldsymbol{V}^{'})f^{*}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}) \subseteq \boldsymbol{Z} \tag{4.13}$$

Now, let us to obtain the set complement of the expression (4.13). We can verify with semantics (4.14) whether a box $\boldsymbol{X}$ belongs or not to the group of non solution boxes.

$$\neg(\forall(\boldsymbol{u} \in \boldsymbol{U}^{'})\exists(\boldsymbol{x} \in \boldsymbol{X}^{'})\exists(\boldsymbol{v} \in \boldsymbol{V}^{'})f^{*}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}) \subseteq \boldsymbol{Z}) \tag{4.14}$$

applying the negation to the quantifiers of the components $\boldsymbol{U},\boldsymbol{X},\boldsymbol{V}$ as well as to the inclusion relation $\subseteq$ in equation (4.14), the resultant semantic is indicated in equation (4.15)

$$\forall(\boldsymbol{x} \in \boldsymbol{X}^{'})\forall(\boldsymbol{v} \in \boldsymbol{V}^{'})\exists(\boldsymbol{u} \in \boldsymbol{U}^{'})(f^{*}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}) \nsubseteq \boldsymbol{Z}) \tag{4.15}$$

Thus, the set of boxes that satisfy equation (4.15) are grouped in non solution boxes

$$Rule\ 2:\ \overline{\Sigma} = \{\boldsymbol{x} \in \boldsymbol{X}^{'}) | \forall(\boldsymbol{v} \in \boldsymbol{V}^{'})\exists(\boldsymbol{u} \in \boldsymbol{U}^{'})(f^{*}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}) \nsubseteq \boldsymbol{Z})\} \tag{4.16}$$

If an inner approximation of the $*$- semantic extension of the continuous function f^{*} is considered, as it was explained in equation (3.101), then we can specify: $f^*(\boldsymbol{X},\boldsymbol{U},\boldsymbol{V}) \Rightarrow Inner(f^*(\boldsymbol{X},\boldsymbol{U},\boldsymbol{V}))$. So, $\overline{\Sigma}$ corresponds to the following expression:

$$\overline{\Sigma} = \{\boldsymbol{x} \in \boldsymbol{X}^{'})|\forall(\boldsymbol{v} \in \boldsymbol{V}^{'})\exists(\boldsymbol{u} \in \boldsymbol{U}^{'})(Inner(f^*(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v})) \nsubseteq \boldsymbol{Z})\} \tag{4.17}$$

Finally, if none of these rules are met, the box $\boldsymbol{X}$ is undefined

4.3.1 Quantified Sets Inversion Algorithm

The general QSIA **Herrero et al. (2005)** to solve the QCSPs explained above is described.

Algorithm 1. Quantified Sets Inversion Algorithm.

QSI-1(In:$\boldsymbol{C}$,$\boldsymbol{X}_o$,ϵ,Out:Σ,$\Delta\Sigma$)
Initialization: Stack $=\boldsymbol{X}_o$; $\Sigma := 0$; $\Delta\Sigma := 0$
Repeat
Unstack $\boldsymbol{X}$;
if Width($\boldsymbol{X}$)$\leq \epsilon$, **then**
$\Delta\Sigma := \Delta\Sigma \cup \boldsymbol{X}$;
else
if(Rule 1 is satisfied) **then**
$\Sigma := \Sigma \cup \boldsymbol{X}$;
else
if(Rule 2 is satisfied)**then**
has no solutions;
else
Bisect $\boldsymbol{X}$ and Stack resulting Boxes;
Until stack=0;

where

- ϵ: QSI stops the bisecting procedure over $\boldsymbol{X}$ when this precision is reached,
- Σ: subpaving that represents an inner approximation of the solution set,
- $\Delta\Sigma$: subpaving represents all the undefined boxes.

The algorithm (1) has five arguments: a set of constraints $\boldsymbol{C}$, the initial interval search space $\boldsymbol{X} = \boldsymbol{X}_o$, a parameter ϵ to stop the bisecting procedure, Σ and $\Delta\Sigma$ are output arguments to contain solution sets and undefined sets respectively. The algorithm begins with an initial box X_o. The first step is to verify if $Width(X) \leq \epsilon$ is true. If $Width(X) \leq \epsilon$ is true, then $\boldsymbol{X}$ will be an undefined box. If the rule 1 is satisfied, $\boldsymbol{X}$ is a solution box. If the rule 1 is not satisfied, the rule 2 is tested, if the rule 2 is satisfied, $\boldsymbol{X}$ is a non-solution box, if the rules 1 and 2 are not satisfied, $\boldsymbol{X}$ is bisected and the boxes are stored in the stack. The following box $\boldsymbol{X}$ is read from stack and rules are reevaluated until $stack = 0$.

4.4 Set inversion applied to nonlinear flat systems

The problem to solve is related to the design of robust controllers for nonlinear flat systems. Specifically, considering an uncertain SISO nonlinear flat system **Hagenmeyer and Delaleau (2003b)**

$$\dot{x}(t) = f(\vartheta_p, x(t), u(t)),\ x(0) = x_0 \tag{4.18}$$

with time $t \in R$, state $x(t) \in R^n$, parameters $\vartheta_p \in R^{n_p}$ and input $u(t) \in R$. The vector field $f : R^{n_p} \times R^n \times R \Rightarrow R^n$ is smooth. The uncertainty of the process parameters $\vartheta_p \in R^p$ are considered as intervals, but not exactly known. It is desired to find the family of possible controllers, in order to guarantee the satisfaction of specifications **Vehí (1998)**; **Bondia et al. (2005)**; **Bondia et al. (2006)**.

Let us consider the optimization approach based on flatness formulated in equation (3.44) and rewritten in (4.19)

$$Max\ \{\vartheta_p | \vartheta_p \in [\underline{\vartheta}_p, \overline{\vartheta}_p]\}$$

$$subject\ to \begin{cases} c_y(\varphi_x(t,\vartheta_p,\vartheta_s), \psi_u(t,\vartheta_p,\vartheta_s)) \subseteq \gamma_y(t), & t \in [t_0, t_f] \\ \varphi_x(t,\vartheta_p,\vartheta_s) \subseteq \gamma_x(t), & t \in [t_0, t_f] \\ \psi_u(t,\vartheta_p,\vartheta_s) \subseteq \gamma_u(t), & t \in [t_0, t_f] \\ \vartheta_p \in [\underline{\vartheta}_p, \overline{\vartheta}_p], \vartheta_s \in [\underline{\vartheta}_s, \overline{\vartheta}_s] \end{cases} \tag{4.19}$$

ϑ_s is a set specification parameters for the flat outputs. ϑ_p is a set of intervals for the plants, $\gamma_y(t)$ defines a region for the flat output, $\gamma_x(t)$ defines bounding regions for the state variables. $\gamma_u(t)$ defines bounding regions for the control input. $\varphi_x(t,\vartheta_p,\vartheta_s)$ defines interval functions of the state variables and $\psi_u(t,\vartheta_p,\vartheta_s)$ interval function of the control input.$c_y(\varphi_x(t,\vartheta_p,\vartheta_s)$, $\psi_u(t,\vartheta_p,\vartheta_s))$ defines general interval functions of the system output.

As it was explicated in Chapter 3, the bounding regions for the flat output γ_y, states γ_x and control input γ_u can be obtained from specification parameters ϑ_s and nominal plant $\overline{\vartheta}_p$. These bounding regions can be seen as different objects where we desired maintain the trajectories of the process output, states and controllers respectively. In other words, we obtain a transformation from hard and soft specifications ϑ_s to bounding regions in time γ_y, γ_x and γ_u. A representation is indicated in Figure 4.4.

Grouping the terms of the left side of inclusion relations from equation (4.19) of the following form:

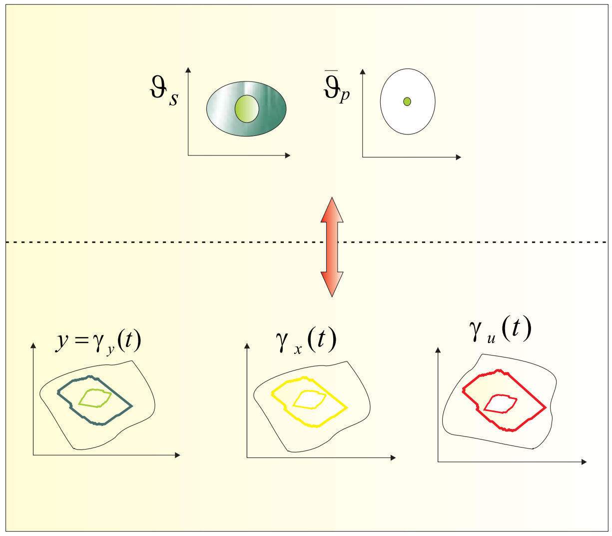


Fig. 4.4. Bounding regions for the flat output γ_y, states γ_x and control input γ_u obtained from specification parameters ϑ_s and nominal plant $\overline{\vartheta}_p$.

$$C(t,\vartheta_p,\vartheta_s) = \begin{bmatrix} c_y(\varphi_x(t,\vartheta_p,\vartheta_s),\psi_u(t,\vartheta_p,\vartheta_s)) \\ \varphi_x(t,\vartheta_p,\vartheta_s) \\ \psi_u(t,\vartheta_p,\vartheta_s) \end{bmatrix} \tag{4.20}$$

and the terms of the right side of the inclusion relations as follows:

$$E_s = \begin{bmatrix} \gamma_y(t) \\ \gamma_x(t) \\ \gamma_u(t) \end{bmatrix} \tag{4.21}$$

Being E_s a set of functions to meet: trajectories in the space of the flat outputs, bounding of the states and control signals, nominal specifications, etc. **Rauh et al. (2005)** and $C(t,\vartheta_p,\vartheta_s)$ the set of interval functions for the system output, states and controllers. If $\vartheta_s = \{\vartheta_s, \vartheta_o\}$ being ϑ_o a set of fixed parameters for the flat outputs then, the set of constraints from equation (4.19) can be expressed as a unique expression of set inclusion of the following form.

$$C(t,\vartheta_p,\vartheta_s,\vartheta_o) \subseteq E_s \tag{4.22}$$

4.4.1 Determining hard and soft uncertainty of the plant

Given hard and soft specifications $y = \gamma_{\underline{y}}(t,\vartheta_s,\bar{\vartheta}_o)$, one nominal plant $\bar{\vartheta}_p$ and a nominal controller $u = \psi(t,\bar{\vartheta}_s,\bar{\vartheta}_o,\bar{\vartheta}_p) \Rightarrow \psi(t,\bar{\vartheta}_k)$, determine two regions of uncertainty of the plant (hard and soft plants) ϑ_p such that some specifications are fulfilled and the constraints are satisfied. A representation of this supposition is indicated in Figure 4.5.

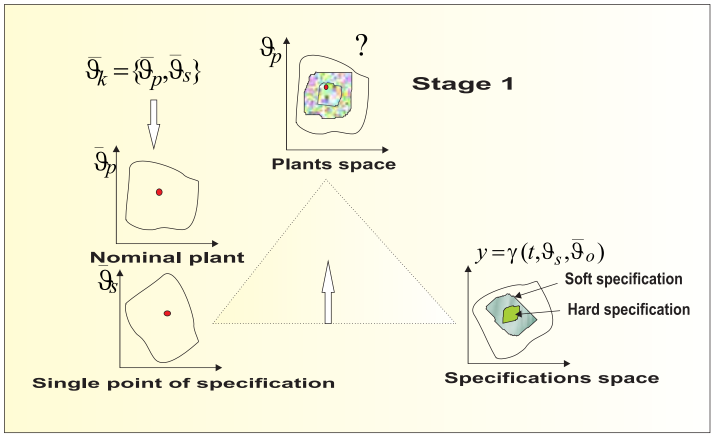


Fig. 4.5. Determination of hard and soft addmisible uncertainty by a nominal controller $u = \psi(t, \bar{\vartheta}_s, \bar{\vartheta}_o, \bar{\vartheta}_p)$, given hard and soft specifications $y = \gamma_y(t, \vartheta_s, \bar{\vartheta}_o)$ and a nominal plant $\bar{\vartheta}_p$.

The problem can be posed as a Quantified Constraints Satisfaction Problem of the form $QCSP = (\vartheta_p, D, C(t, \vartheta_p, \vartheta_s, \vartheta_o))$. where D is a domain associated with the variables. The solution of this kind of problem corresponds to determine $\Sigma_{\forall\exists}$-solution sets (i.e., all the occurrences of the existential quantifier $\exists$ are preceded by the occurrences of the universal quantifier $\forall$) of all the parameters of the process ϑ_p such that the constraints $C(t, \vartheta_p, \vartheta_s, \vartheta_o)$ are satisfied **Herrero et al. (2005)**. If the system output $y = h(x, \vartheta_p)$ depends on state variables and plant parameters, the solution set is defined by

$$\Sigma_{\forall\exists} = \{\vartheta_p \in R | \forall(\vartheta_o \in \boldsymbol{\vartheta}'_o)\forall(t \in \boldsymbol{t}')\exists(\vartheta_s \in \boldsymbol{\vartheta}'_s)C(t, \vartheta_p, \vartheta_s, \vartheta_o) \subseteq E_s\} \tag{4.23}$$

Being ϑ_p sets of the process parameters, $\boldsymbol{\vartheta}'_o$ additional parameter domain sets corresponding to the flat outputs, $\boldsymbol{\vartheta}'_s$ desired specification domain sets corresponding to the flat outputs. All the constraints will be verified within an interval of time $\boldsymbol{t}'$.

From general equation (4.23), the specific implementation to obtain the maximum admissible uncertainty ϑ_p by a nominal controller $u = \psi(t, \bar{\vartheta}_s, \bar{\vartheta}_o, \bar{\vartheta}_p)$ is indicated in equation 4.24.

$$\begin{aligned}\Sigma_{\forall\exists} = \{\vartheta_p \in R | &\forall(\vartheta_o \in \boldsymbol{\vartheta}'_o)\forall(t \in \boldsymbol{t}')\exists(\vartheta_s \in \boldsymbol{\vartheta}'_s) \\ &(c_y(t, \bar{\vartheta}_s, \bar{\vartheta}_o, \vartheta_p) \subseteq \gamma_y(t) \wedge \\ &\varphi_x(t, \bar{\vartheta}_s, \bar{\vartheta}_o, \vartheta_p) \subseteq \gamma_x(t) \wedge \\ &\psi_u(t, \bar{\vartheta}_s, \bar{\vartheta}_o, \vartheta_p) \subseteq \gamma_u(t))\}\end{aligned} \tag{4.24}$$

A certain region within the output space $\gamma_y(t)$, obtained when evaluating the QCSP (4.24), is equivalent to certain interval within the specification parameters ϑ_s. A representation is indicated in Figure 4.6.

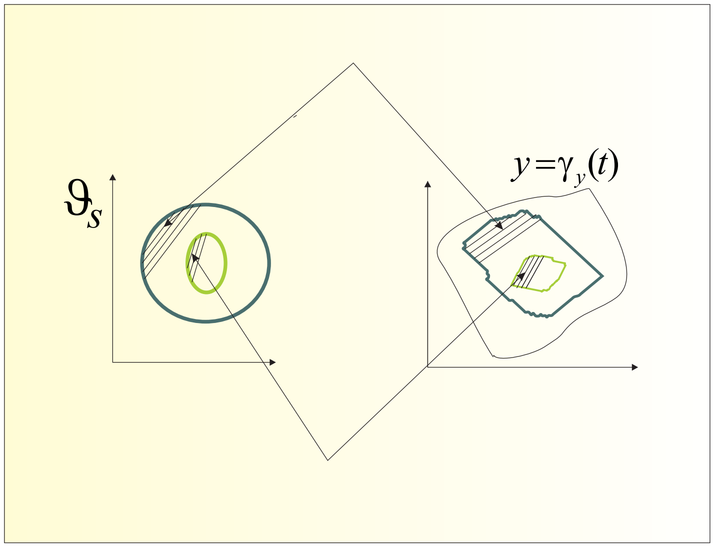


Fig. 4.6. Relation between regions of trajectories within the output space $\gamma_y(t)$ and specification parameters ϑ_s.

The process to obtain ϑ_p can initiate obtaining the space of soft uncertainty from the soft specifications. The bounding regions of the flat outputs, state variables and controllers are obtained using the values of the nominal plant and controller. Then one can repeat the process to obtain the hard uncertainty to fulfill with the hard specifications. Another form to obtain the hard and soft uncertainty is solving (4.23) in a single step simultaneously from the hard and soft specifications.

A Quantified Sets Inversion Algorithm verifies if a box ϑ_p is solution, non solution or undefined. Boxes that satisfy the constraints are grouped in a solution set. Those one that do not satisfy the constraints in a non solution set, and those that partially satisfy the constraints in an undefined set. To start, a set of initial boxes is specified to initiate the search of solution sets. Then an interval division technique (branch-and-bound) is applied evaluating the three following rules:

Rule 1:

$$\forall(\vartheta_p \in \boldsymbol{\vartheta}_p^{'})\forall(\vartheta_o \in \boldsymbol{\vartheta}_o^{'})\forall(t \in \boldsymbol{t}^{'})\exists(\vartheta_s \in \boldsymbol{\vartheta}_s^{'})C(t,\vartheta_p,\vartheta_s,\vartheta_o) \subseteq E_s \Leftrightarrow \vartheta_p \subseteq \Sigma \tag{4.25}$$

This quantified constraint is used to prove that a box ϑ_p is contained in the solution set. This cannot be easily proved by means of classical interval computations. The quantified constraint corresponding to Rule 1, can be checked through the following reasoning:

$$\begin{aligned}&Outer(C^*(\vartheta_p,\vartheta_o,\vartheta_s,t)) \subseteq E_s \Rightarrow C^*(\vartheta_p,\vartheta_o,\vartheta_s,t) \subseteq E_s\\&\Leftrightarrow \forall(\vartheta_p \in \boldsymbol{\vartheta}_p^{'})\forall(\vartheta_o \in \boldsymbol{\vartheta}_o^{'})\forall(t \in \boldsymbol{t}^{'})\exists(\vartheta_s \in \boldsymbol{\vartheta}_s^{'})C(\vartheta_p,\vartheta_s,\vartheta_o,t) \subseteq E_s\\&\Leftrightarrow \vartheta_p \subseteq \Sigma\end{aligned} \tag{4.26}$$

$Outer(C^*(\vartheta_p,\vartheta_o,\vartheta_s,t))$ is an outer approximation of the $*$- semantic extension of the continuous function C.

from the $*$- semantic theorem, concretely from equation (3.89), it follows as:

$$f^*(\boldsymbol{A}) \subseteq F(\boldsymbol{A}) \Leftrightarrow \forall(a_p \in \boldsymbol{A'}_p)Q(z,F(\boldsymbol{A}))\exists(a_i \in \boldsymbol{A}_i^{'})(z = f(a_p,a_i)) \tag{4.27}$$

In this case, we can see that the universal quantifiers ($\forall$) correspond to the proper components and the existential ones ($\exists$) to the improper components. Therefore $\vartheta_p,\vartheta_o,t$ are proper intervals and ϑ_s is an improper interval.

In order to prove the second rule and verify that a box has no interaction with the solution set, the following implication is used:
Rule 2:

$$\neg(\forall(\vartheta_o \in \boldsymbol{\vartheta}_o^{'})\forall(t \in \boldsymbol{t}^{'})\exists(\vartheta_s \in \boldsymbol{\vartheta}_s^{'})\exists(\vartheta_p \in \boldsymbol{\vartheta}_p^{'})C(t,\vartheta_p,\vartheta_s,\vartheta_o)) \Leftrightarrow \vartheta_p \subseteq \bar{\Sigma} \tag{4.28}$$

where $\bar{\Sigma}$ is the complementary set of Σ defined by

$$\bar{\Sigma} = \{\vartheta_p \in R | \forall(\vartheta_s \in \boldsymbol{\vartheta}_s^{'})\exists(\vartheta_o \in \boldsymbol{\vartheta}_o^{'})\exists(t \in \boldsymbol{t}^{'})\neg(C(t,\vartheta_p,\vartheta_s,\vartheta_o) \subseteq E_s)\} \tag{4.29}$$

The parameters that do not fulfill the specifications are grouped in a non solution set, see Figure 4.7.

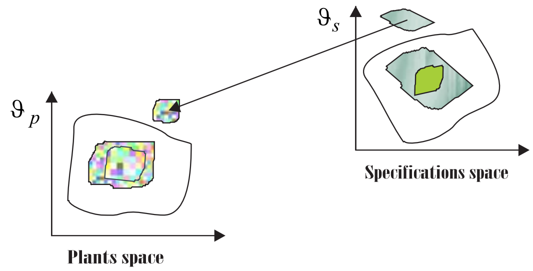


Fig. 4.7. Non solution set of plants.

This quantified constraint is, analogously, implied by the following interval exclusion:

$$\begin{aligned}
&Inner(C^*(\vartheta_p, \vartheta_o, \vartheta_s, t)) \nsubseteq E_s \Rightarrow C^*(\vartheta_p, \vartheta_o, \vartheta_s, t) \nsubseteq E_s \\
&\Leftrightarrow \neg(\forall(\vartheta_o \in \boldsymbol{\vartheta}_o^{'})\forall(t \in \boldsymbol{t}^{'})\exists(\vartheta_s \in \boldsymbol{\vartheta}_s^{'})\exists(\vartheta_p \in \boldsymbol{\vartheta}_p^{'})C(t, \vartheta_p, \vartheta_s, \vartheta_o) \subseteq E_s) \\
&\Leftrightarrow \forall(\vartheta_s \in \boldsymbol{\vartheta}_s^{'})\forall(\vartheta_p \in \boldsymbol{\vartheta}_p^{'})\exists(\vartheta_o \in \boldsymbol{\vartheta}_o^{'})\exists(t \in \boldsymbol{t}^{'})\neg(C(t, \vartheta_p, \vartheta_s, \vartheta_o) \subseteq E_s) \\
&\Rightarrow \vartheta_p \subseteq \bar{\Sigma}
\end{aligned} \tag{4.30}$$

where ϑ_o, t are proper intervals, ϑ_p, ϑ_s are improper ones. $Inner(C^*(\vartheta_p, \vartheta_o, \vartheta_s, t)) \nsubseteq E_s$ is an inner approximation of the $*$-semantic extension of the continuous function C. Finally, if none of these rules are accomplished, the box ϑ_p is undefined.

Rule 3: otherwise, ϑ_p is undefined.

If the specifications are partially satisfied then the set of plants is grouped in a set of indefinite plants, as it is ilustrated in Figure 4.8.

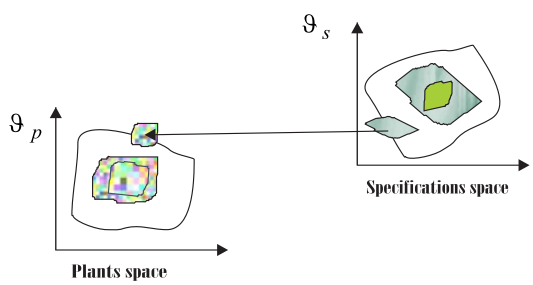


Fig. 4.8. Undefined solution set of plants.

When the constraints are of the form $C(t, \vartheta_p, \vartheta_s, \vartheta_o) \subseteq E_s$, with C being a continuous function from R^n to R^m and each existentially quantified variable appears in only one function component, the problem is reduced to m different problems, one for each component function. Then, the solution set may be obtained as:

$$\Sigma = \Sigma_1 \cap ... \cap \Sigma_m \tag{4.31}$$

on the other hand, if the system output $y = h(x, u, \vartheta_p)$ depends on state variables, plant parameters and control input, the solution set is defined by:

$$\begin{aligned}\Sigma_{\forall\exists} = \{&\vartheta_p \in R | \forall(\vartheta_o \in \boldsymbol{\vartheta}_o^{'})\forall(t \in \boldsymbol{t}^{'})\\ &\exists(\vartheta_k \in \boldsymbol{\vartheta}_k^{'})\exists(\vartheta_s \in \boldsymbol{\vartheta}_s^{'})C(t, \vartheta_k, \vartheta_p, \vartheta_s, \vartheta_o) \subseteq E_s\}\end{aligned} \tag{4.32}$$

From general equation (4.32), the specific implementation to obtain the maximum admissible uncertainty ϑ_p by a nominal controller $\bar{\vartheta}_k$ is indicated in equation 4.33.

$$\begin{aligned}\Sigma_{\forall\exists} = \{&\vartheta_p \in R | \forall(\vartheta_o \in \boldsymbol{\vartheta}_o^{'})\forall(t \in \boldsymbol{t}^{'})\exists(\vartheta_k \in \boldsymbol{\vartheta}_k^{'})\\ &\exists(\vartheta_s \in \boldsymbol{\vartheta}_s^{'})(c_{\underline{y}}(t, \bar{\vartheta}_k, \bar{\vartheta}_s, \bar{\vartheta}_o, \vartheta_p) \subseteq \gamma_y(t) \wedge\\ &\varphi_x(t, \bar{\vartheta}_k, \bar{\vartheta}_s, \bar{\vartheta}_o, \vartheta_p) \subseteq \gamma_x(t) \wedge\\ &\psi_u(t, \bar{\vartheta}_k, \bar{\vartheta}_s, \bar{\vartheta}_o, \vartheta_p) \subseteq \gamma_u(t))\}\end{aligned} \tag{4.33}$$

A representation is indicated in Figure 4.9.

4.4.2 Determining hard and soft controllers

Given hard and soft specifications $\gamma_y(t, \vartheta_s, \bar{\vartheta}_o)$ and a family of plants ϑ_p determine hard and soft controllers $u = \psi(t, \vartheta_s, \bar{\vartheta}_o, \vartheta_p) \Rightarrow \psi(t, \vartheta_k)$ to guarantee the satisfaction of some hard and soft specifications for all the plants in the family and the constraints are satisfied. In Figure 4.10 we depicted a representation.

Hard and soft controllers to guarantee the satisfaction of the hard and soft specifications can be obtained from two forms. The first corresponds to the direct form using the equation (3.41). This equation is rewritten in (4.34)

$$u^* = \psi(t, \vartheta_p, \vartheta_s, \vartheta_o) \tag{4.34}$$

Equation (4.34) can be used when the system ouput corresponds explicitely with some state variables. Then one replaces in (4.34) the hard or soft uncertainty in ϑ_p and a point for ϑ_s and ϑ_o. Therefore the set of hard and soft plants defines the hard and soft controllers.

The second form corresponds to solve the following QCSP

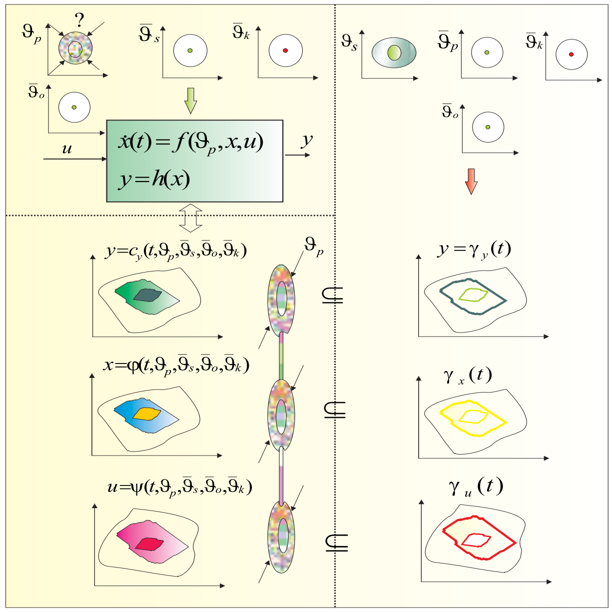


Fig. 4.9. Considerations to determine the maximum admissible uncertainty ($\vartheta_p(\forall)$) by a nominal controller ($\vartheta_k(\exists)$) ensuring that some specifications are met ($\vartheta_s(\exists)$) and the constraints are satisfied.

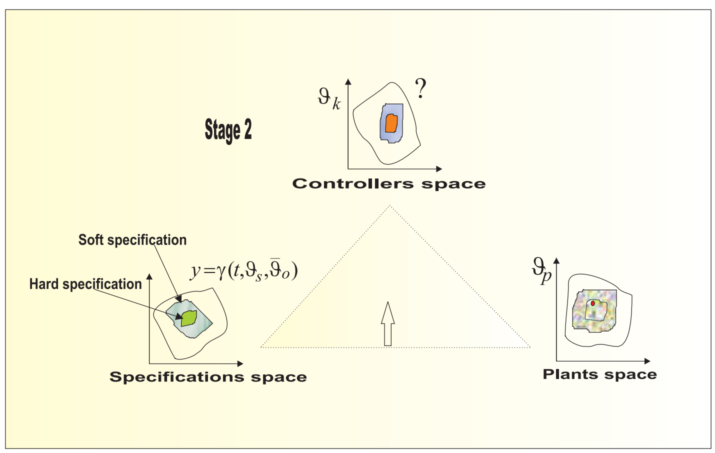


Fig. 4.10. Determination of hard and soft controllers $u = \psi(t, \vartheta_k)$ to guarantee the satisfaction of some hard and soft specifications $y = \gamma(t, \vartheta_s, \bar{\vartheta}_o)$ for all the plants in the family ϑ_p.

$$\begin{aligned}\Sigma_{\forall\exists} = \{&\vartheta_k \in R | \forall(\vartheta_o \in \boldsymbol{\vartheta}'_o)\forall(t \in \boldsymbol{t}')\forall(\vartheta_p \in \boldsymbol{\vartheta}'_p)\\ &\exists(\vartheta_s \in \boldsymbol{\vartheta}'_s) C(t, \vartheta_k, \vartheta_p, \vartheta_s, \vartheta_o) \subseteq E_s\}\end{aligned} \tag{4.35}$$

This case corresponds when the system output depends on the state variables, plant parameters and control input $y = h(x, u, \vartheta_p)$. In other words, the general interval function of the system output $c_y(t, \vartheta_s, \vartheta_k, \vartheta_p)$ depends on specification parameters ϑ_s, parameters of the controller ϑ_k and plant parameters ϑ_p as it was explicated in Chapter 3 (see equation (3.54)).

From general equation (4.35), the specific implementation to obtain the hard and soft controllers ($\vartheta_k(\forall)$) that could ensure that some specification are met ($\vartheta_s(\exists)$) and the constraints are satisfied under parametric uncertainty in the plant ($\vartheta_p(\forall)$) is indicated in equation 4.36. A representation is indicated in Figure 4.11.

$$\begin{aligned}\Sigma_{\forall\exists} = \{&\vartheta_k \in R | \forall(\vartheta_o \in \boldsymbol{\vartheta}'_o)\forall(t \in \boldsymbol{t}')\forall(\vartheta_p \in \boldsymbol{\vartheta}'_p)\exists(\vartheta_s \in \boldsymbol{\vartheta}'_s)\\ &(c_y(t, \vartheta_k, \bar{\vartheta}_s, \bar{\vartheta}_o, \vartheta_p) \subseteq \gamma_y(t) \wedge\\ &\varphi_x(t, \vartheta_k, \bar{\vartheta}_s, \bar{\vartheta}_o, \vartheta_p) \subseteq \gamma_x(t) \wedge\\ &\psi_u(t, \vartheta_k, \bar{\vartheta}_s, \bar{\vartheta}_o, \vartheta_p) \subseteq \gamma_u(t))\}\end{aligned} \tag{4.36}$$

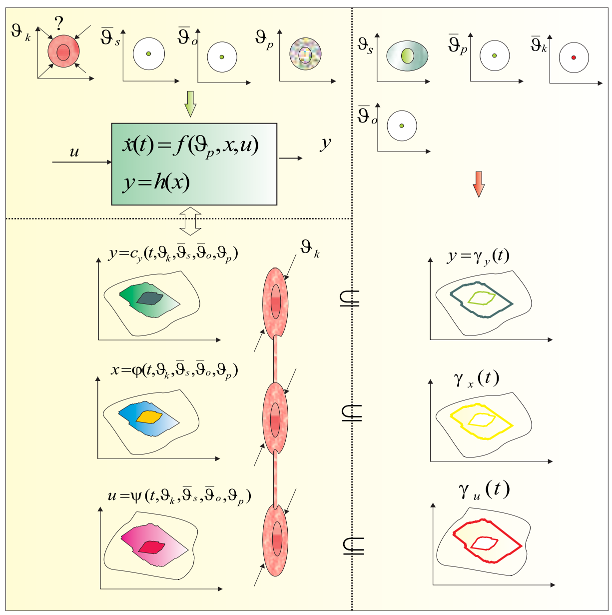


Fig. 4.11. Considerations to determine the hard and soft controllers ($\vartheta_k(\forall)$) that could ensure that some specification are met ($\vartheta_s(\exists)$) and the constraints are satisfied under parametric uncertainty in the plant ($\vartheta_p(\forall)$).

With a specific point for $\overline{\vartheta}_k$, $\overline{\vartheta}_s$, $\overline{\vartheta}_o$ and $\overline{\vartheta}_p$ we obtain a single trajectory for c_y, φ and ψ as it is indicated in Figure 4.12. With hard or soft uncertainty in the plant ϑ_p and a specific controller $\overline{\vartheta}_k$, we obtain a first expansion of the space to c_y, φ and ψ. The hard or soft uncertainty in the plant $\vartheta_p(\forall)$ determined in the previous problem can be considered. Thus, with controllers ϑ_k of the solution set and the preset hard and soft uncertainty ϑ_p, the space to c_y, φ and ψ are expanding while the inclusion relation $c_y \subseteq \gamma_y \wedge \varphi \subseteq \gamma_x \wedge \psi \subseteq \gamma_u$ is fulfilled.

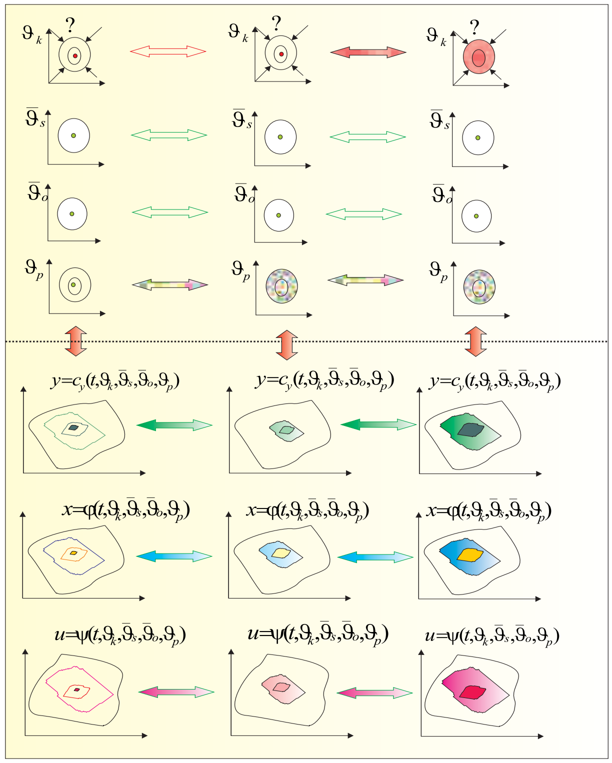


Fig. 4.12. With controllers ϑ_k of the solution set and the preset hard and soft uncertainty ϑ_p, the space to c_y, φ and ψ are expanding while the inclusion relation $c_y \subseteq \gamma_y \wedge \varphi \subseteq \gamma_x \wedge \psi \subseteq \gamma_u$ is fulfilled.

The procedure to evaluate the equation (4.36) is similar as it was described for the hard and soft uncertainty. First, one obtain the soft controllers from soft specifications considering a family of plants and then the hard controllers from hard specifications.

4.4.3 Determining hard and soft specifications

Given hard and soft plants ϑ_p and hard and soft controllers $u = \psi(t, \bar{\vartheta}_s, \bar{\vartheta}_o, \vartheta_p)$, determine attainable hard and soft specifications $y = \gamma_y(t, \vartheta_s, \bar{\vartheta}_o)$ by some hard and soft controllers for all the hard and soft plants ϑ_p and the constraints are satisfied. In Figure 4.13 a representation can be viewed.

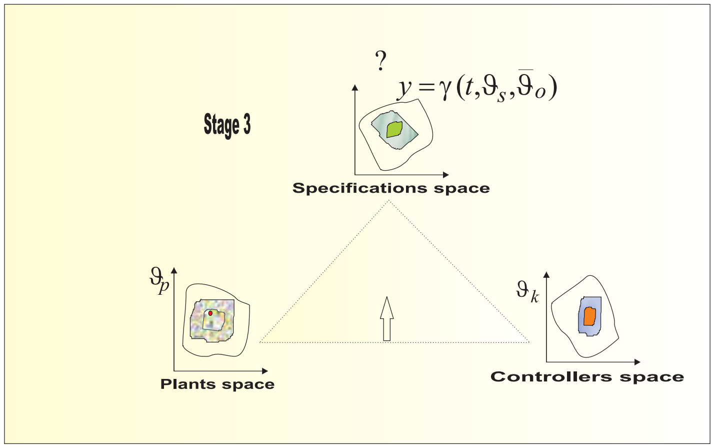


Fig. 4.13. Determination of hard and soft attainable specifications $y = \gamma_y(t, \vartheta_s, \bar{\vartheta}_o)$ by some hard and soft controllers $u = \psi(t, \bar{\vartheta}_s, \bar{\vartheta}_o, \vartheta_p)$ for all the hard and soft plants ϑ_p.

If the system output $y = h(x, \vartheta_p)$ depends on state variables and plant parameters, the solution set corresponds to:

$$\begin{aligned}\Sigma_{\forall\exists} = \{&\vartheta_s \in R | \forall(\vartheta_o \in \boldsymbol{\vartheta}'_o)\forall(t \in \boldsymbol{t}')\exists(\vartheta_p \in \boldsymbol{\vartheta}'_p)\\ &\cdot C(t, \vartheta_p, \vartheta_s, \vartheta_o) \subseteq E_s\}\end{aligned} \tag{4.37}$$

A nominal controller $u = \psi(t, \bar{\vartheta}_s, \bar{\vartheta}_o, \overline{\vartheta}_p)$ is preset. The specific implementation is indicated in the following equation:

$$\begin{aligned}\Sigma_{\forall\exists} = \{&\vartheta_s \in R | \forall(\vartheta_o \in \boldsymbol{\vartheta}'_o)\forall(t \in \boldsymbol{t}')\exists(\vartheta_p \in \boldsymbol{\vartheta}'_p)\\ &(c_y(t, \vartheta_s, \bar{\vartheta}_o, \bar{\vartheta}_p) \subseteq \gamma_y(t) \wedge\\ &\varphi_x(t, \vartheta_s, \bar{\vartheta}_o, \bar{\vartheta}_p) \subseteq \gamma_x(t) \wedge\\ &\psi_u(t, \vartheta_s, \bar{\vartheta}_o, \bar{\vartheta}_p) \subseteq \gamma_u(t))\}\end{aligned} \tag{4.38}$$

The three rules are the following:
Rule 1:

$$\forall(\vartheta_s \in \boldsymbol{\vartheta}_s^{'})\forall(\vartheta_o \in \boldsymbol{\vartheta}_o^{'})\forall(t \in \boldsymbol{t}^{'})\exists(\vartheta_p \in \boldsymbol{\vartheta}_p^{'}) \\ \cdot C(t, \vartheta_p, \vartheta_s, \vartheta_o) \subseteq E_s \Leftrightarrow \vartheta_s \subseteq \Sigma \tag{4.39}$$

Rule 2:

$$\neg(\forall(\vartheta_o \in \boldsymbol{\vartheta}_o^{'})\forall(t \in \boldsymbol{t}^{'})\exists(\vartheta_p \in \boldsymbol{\vartheta}_p^{'}) \\ \cdot\exists(\vartheta_s \in \boldsymbol{\vartheta}_s^{'})C(t, \vartheta_p, \vartheta_s, \vartheta_o) \subseteq E_s) \Leftrightarrow \vartheta_s \subseteq \bar{\Sigma} \tag{4.40}$$

Rule 3: otherwise, ϑ_s is undefined.

We are going to select a controller of the resulting family and will try to find the attainable specifications by the selected controller. The specifications space is subdivided in small boxes and the constraints are verified.

In Figure 4.14 we represent a single controller and a possible specifications space that can reach. In other words, one hopes that some controllers of the core can reach all the range of specifications.

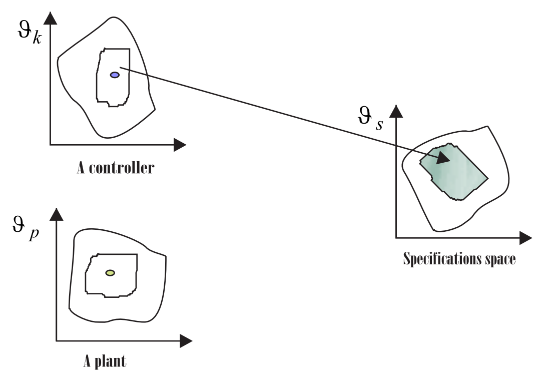


Fig. 4.14. Attainable specifications by a controller.

Generalizing, we could evaluate the attainable specifications by other controllers of the solution set. It is possible that the controllers that are located in the ends of the solutions space can reach only part of the specifications as it is indicated in Figures 4.15 and 4.16.

Also we will evaluate the controllers that are located outside the core of solutions to verify the specifications space that can reach as it is indicated in Figure 4.17.

When we want to analyze the behaviour of a system within a specified region (hard specification) with the consideration that could operate beyond its region (soft specification), we can find the family of controllers that could satisfy certain region and a tolerance region. A nominal controller may be able to meet the hard and soft specifications, as it is indicate in Figure 4.18.

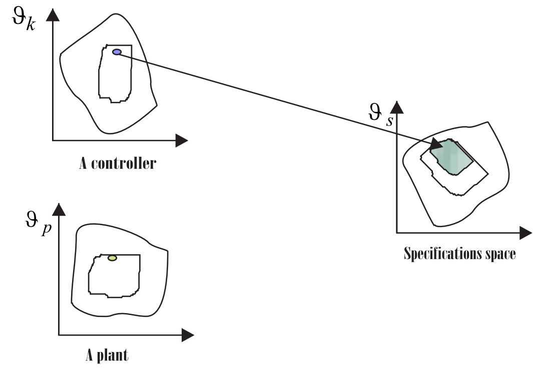


Fig. 4.15. Attainable specifications by a controller on the left end of the solutions space.

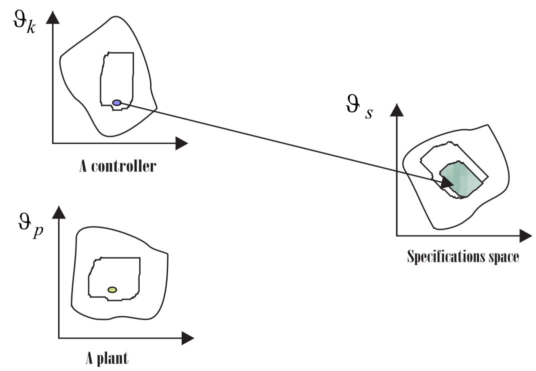


Fig. 4.16. Attainable specifications by a controller on the right end of the solutions space.

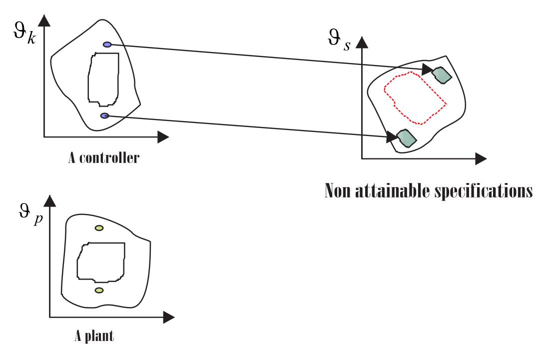


Fig. 4.17. Unattainable specifications by some controllers that are outside of the solutions space.

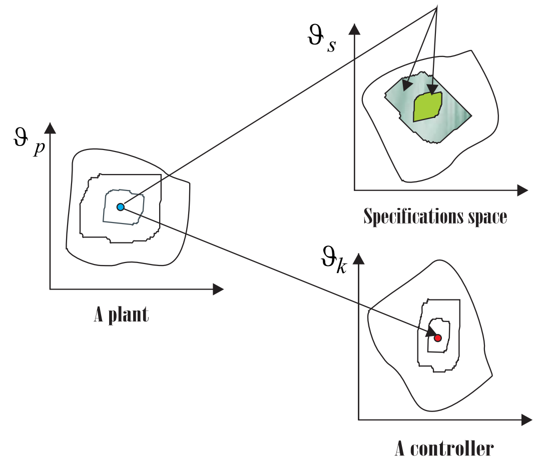


Fig. 4.18. A nominal controller can satisfy hard and soft specifications.

The second form corresponds to solve the following QCSP

$$\begin{aligned} \varSigma_{\forall\exists} = \{&\vartheta_s \in R | \forall(\vartheta_o \in \boldsymbol{\vartheta}'_o)\forall(t \in \boldsymbol{t}')\forall(\vartheta_p \in \boldsymbol{\vartheta}'_p) \\ &\exists(\vartheta_k \in \boldsymbol{\vartheta}'_k) C(t, \vartheta_s, \vartheta_o, \vartheta_k, \vartheta_p) \subseteq E_s\} \end{aligned} \tag{4.41}$$

This case corresponds when the system output depends on the state variables, plant parameters and control input $y = h(x, u, \vartheta_p)$. See equation (3.54) in Chapter 3.

From general equation (4.41), the specific implementation to determine the achievable specifications $\vartheta_s(\forall)$ by some nominal controller $(\vartheta_k(\exists))$ and the constraints are satisfied under parametric uncertainty in the plant $\vartheta_p(\forall)$ is indicated in equation (4.42).

$$\begin{aligned} \varSigma_{\forall\exists} = \{&\vartheta_s \in R | \forall(\vartheta_o \in \boldsymbol{\vartheta}'_o)\forall(t \in \boldsymbol{t}')\forall(\vartheta_p \in \boldsymbol{\vartheta}'_p)\exists(\vartheta_k \in \boldsymbol{\vartheta}'_k) \\ &(c_y(t, \vartheta_s, \bar{\vartheta}_o, \bar{\vartheta}_k, \vartheta_p) \subseteq \gamma_y(t) \wedge \\ &\varphi_x(t, \vartheta_s, \bar{\vartheta}_o, \bar{\vartheta}_k, \vartheta_p) \subseteq \gamma_x(t) \wedge \\ &\psi_u(t, \vartheta_s, \bar{\vartheta}_o, \bar{\vartheta}_k, \vartheta_p) \subseteq \gamma_u(t))\} \end{aligned} \tag{4.42}$$

A representation is indicated in Figure 4.19.

4.4.4 Controller design based on differential flatness to trajectory tracking

After obtaining the family of robust controllers that meet hard and soft specifications in a guaranteed manner under parametric uncertainty in the process, one of them is chosen. The selected controller will be used as a feedforward

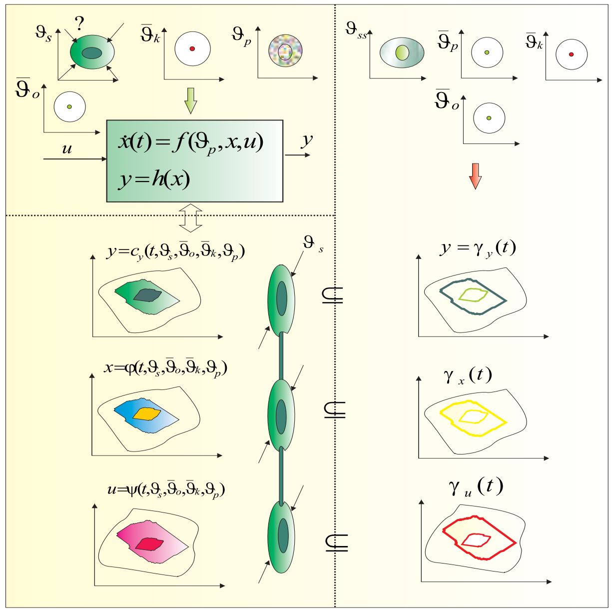


Fig. 4.19. Considerations to determine the achievable specifications ($\vartheta_s(\forall)$) by some nominal controllers ($\vartheta_k(\exists)$), satisfying the constraints under parametric uncertainty in the plant ($\vartheta_p(\forall)$).

in the control scheme as it was explained in Section 3.2. Flatness allows one to solve trajectory tracking problems in many ways. Because we want to maintain the desired output within a specified region, design parameters of the feedback controller are determined from viewpoint of the input-output behaviour of the feedback system **Zoran et. al (2003)**. That is, a feedback controller will ensure that the output of the system will stay within some desired specifications under parametric uncertainty of the plant.

All the controllers that fulfill the specifications are grouped in sets of feasible controllers **Bondia et al. 2004**; **Andújar et al. 2004**.

The problem to solve is related to obtaining hard and soft controllers in order to guarantee the satisfaction of hard and soft closed-loop specifications. This problem can be raised on similar way as in the previous section.

Without loss of generality, let us consider a control scheme based on differential flatness for the SISO case with tracking reference. Assume also that the flat output corresponds to some state variable as it is indicated in Figure 4.20

The terms α and β are obtained with an approach based on differential flatness and the feedwordard corresponds to w.

Consider the closed-loop nonlinear system (4.43)

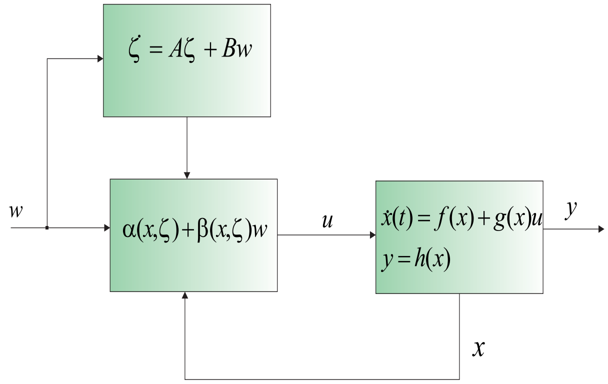


Fig. 4.20. Control scheme based on differential flatness.

$$\dot{x}(t) = f_{cl}(\vartheta_p, \vartheta_c, x(t)) \tag{4.43}$$

It is desired to find the hard and soft ∀∃-solution set of controller parameters ϑ_c for which robust performance holds, i.e.,

$$\begin{aligned} \Sigma_{\vartheta_c} = \{&\vartheta_c \in R | \forall(t \in \boldsymbol{t}') \forall(\vartheta_p \in \boldsymbol{\vartheta}'_p) \exists(\theta_q \in \boldsymbol{\theta}'_q) \\ &\cdot \mu_{cl}(t, \vartheta_p, \vartheta_c) \subseteq M(t, \theta_q)\} \end{aligned} \tag{4.44}$$

where $\mu_{cl}(t, \vartheta_p, \vartheta_c)$ is a closed-loop function and $M(t, \theta_q)$ is a desired reference model with a set of parameters θ_q and a set of domains $\boldsymbol{\theta}'_q$.

The constraint has the form $C_{cl}(t, \vartheta_c, \vartheta_p, \theta_q) := \{\mu_{cl}(t, \vartheta_p, \vartheta_c) \subseteq M(t, \theta_q)\}$. The three rules to solve the problem are the following:

Rule 1:

$$\forall(\vartheta_c \in \boldsymbol{\vartheta}'_c) \forall(t \in \boldsymbol{t}') \forall(\vartheta_p \in \boldsymbol{\vartheta}'_p) \exists(\theta_q \in \boldsymbol{\theta}'_q) C_{cl}(t, \vartheta_c, \vartheta_p, \theta_q) \Leftrightarrow \vartheta_c \subseteq \Sigma \tag{4.45}$$

Rule 2:

$$\neg(\forall(t \in \boldsymbol{t}') \forall(\vartheta_p \in \boldsymbol{\vartheta}'_p) \exists(\theta_q \in \boldsymbol{\theta}'_q) \exists(\vartheta_c \in \boldsymbol{\vartheta}'_c) C_{cl}(t, \vartheta_c, \vartheta_p, \theta_q)) \Leftrightarrow \vartheta_c \subseteq \bar{\Sigma} \tag{4.46}$$

Rule 3: otherwise, ϑ_c is undefined.

4.5 Method design

Using the formulation presented in this thesis, robust controllers can be designed by using the following method:

1. Given some desired specifications, apply the Quantified Set Inversion Algorithm to find the family of robust controllers that fulfill the specifications in a guaranteed way under parametric uncertainty in the process.
2. Design a state feedback control law based on differential flatness to drive the dynamic system.
3. Select a feedforward controller of the resulting family from step one and find the set of parameters (paving) of the state feedback control such that the specifications are met.
4. Select the located parameters in the center of paving from previous step to obtain robust performance of the controlled system under parametric uncertainty of the plant.

4.6 Advantages and limitations of the proposed technique

The robust control approach proposed in this thesis contains several elements that have not been treated and considered in the literature. Some of them are the following: 1) The variables are quantified: This allows us to specify the feedforward controller to apply to the feedback control system, the admissible uncertainty by the feedforward controller, the free variables that we wish to solve and obtain solution set within the specification parameters. 2) The bounding regions for the state variables, inputs and outputs are obtained by properly exploiting the flatness property. This reduces the execution time in the solutions search. On the other hand, the control engineer can set limits on the control input, states and outputs without changing the proposed approach. 3) The constraints are formulated in terms of set inclusion. An advantage obtained when we use the inclusion operators in the constraints is that the specification parameters defined as existentials, they do not need to be partitioned. It is suffice to evaluate whether a trajectory is included within the bounding regions. So the search of solutions within the specification parameters or bounding regions is straightforward. 4) The proposed design approach is based on sets. The advantage of the proposed approach regarding other based on single-trajectories is the robustness of the controllers obtained. A limitation of the proposed technique is that the approach is only applicable to nonlinear systems that are flat.

4.7 Conclusions

It has been shown how to find a family of possible controllers with a Quantified Sets Inversion Algorithm, in order to guarantee the satisfaction of specifications, as well as the three main rules to find the solution set referred to the attainable specifications by family of controllers. A new approach of robust possibilistic control for nonlinear flat systems has been developed. It has raised how to determine a family of robust controllers that meet of guaranteed manner a set of specifications. It has developed a method to tune parameters of the feedback controller to ensure the robust performance of the system.

5 Applications

In this Chapter three main applications are developed. The first example is applied to a linear system (DC motor) since all the controllable linear systems are flat and the method is easier to explain, the second and third example is applied to nonlinear flat systems (simple pendulum and fed-batch bioreactor).

5.1 Applications to linear systems

5.1.1 Dynamic model of a DC motor

The dynamic model of a DC motor can be consulted in the work of Weearasooriya and El-sharkawi **Weearasooriya and El-sharkawi (1991)**

$$\begin{aligned} \dot{x}_1 &= -\tfrac{R_a}{L_a}x_1 - \tfrac{K}{L_a}x_2 + \tfrac{1}{L_a}u \\ \dot{x}_2 &= \tfrac{K}{J}x_1 - \tfrac{D}{J}x_2 - \tfrac{T_L}{J} \end{aligned} \tag{5.1}$$

where the parameters are: x_2 rotor speed (rad/s), u input voltage (V), x_1 armature current (Amp), T_L load torque (Nm), J rotor inertia (Nm^2), K torque and back emf constant (NmA^{-1}), D damping constant (Nms), R_a armature resistance (Ω) and L_a armature inductance (H). The same model was used by Sira-Ramírez and Agrawal **Sira-Ramírez and Agrawal (2004)** with a constant load torque $T_l = 0$. In this work, the load torque is considered as a linear function of the rotor speed, that is :$T_l = p_1 x_2$. The parameterization of all system variables on function of the flat output $y = x_2$ and a finite number of its derivatives are expressed in equations (5.2),(5.3) and (5.4)

$$\begin{aligned} x_2 &= y \\ T_l &= p_1 y \end{aligned} \tag{5.2}$$

$$x_1 = \tfrac{1}{K}(J\dot{y} + Dy + p_1 y) \tag{5.3}$$

$$u = \frac{L_a}{K}(J\dddot{y} + D\ddot{y} + p_1\dot{y}) + \frac{R_a}{K}(J\dot{y} + Dy + p_1 y) + Ky$$
$$u = \frac{JL_a}{K}\ddot{y} + (\frac{L_a D + L_a p_1 + R_a J}{K})\dot{y} + (\frac{R_a D}{K} + p_1 + K)y \tag{5.4}$$

5.1.2 Simulation of the system considering a region of flat output

The trajectory planning in time of the flat output can be constructed with a polynomial function(Bézier polynomial). For example, a linear Bézier curve between two points y_o, y_1 can be obtained with the following expression:

$$y = y_o + t(y_1 - y_o) \Rightarrow y_o(1-t) + ty_1 \tag{5.5}$$

where $t \in [0,1]$. With this polynomial function we may obtain a straight line between points y_o, y_1. If instead of a straight line between the two points, we would desire to planify a parabolic segment between y_o and y_1, then we can use a cuadratic or cubic bézier polynomial such as:

$$y = (1-t)^2 y_o + 2(1-t)ty_1 + t^2 y_1 \Rightarrow cuadratic$$
$$y = (1-t)^3 y_o + 3(1-t)^2 ty_1 + 3(1-t)t^2 y_1 \Rightarrow cubic \tag{5.6}$$

These functions are obtained of the first equation. Another form of trajectory planning of the flat output is by means of an exponential function of the type

$$y = y_o + (y_1 - y_o)(1 - e^{(-t/\tau)}) \tag{5.7}$$

with the time constant τ, we adjust the response speed of the planified trajectory. In general, the task of desired trajectory planning of flat output can be made in many ways. If we only have a set of points around any desired trajectory, for instance, an aproximation function can be used together with a neural network to determine the aproximation function parameter values. So, we consider polynomial functions with defined derivatives, smooth and monotones. All the derivatives of the functions will be made respect to the time. Thus, all the resulting equations and its parameters will be used in the field of the modal interval arithmetic. We choose a Bézier polynomial (5.8) to construct the flat output $y = x_2$ (rotor speed) of the form:

$$y = y_o + (y_1 - y_o)B(\tau) \tag{5.8}$$

being $B(\tau)$ a polynomial function in time, τ is a point inside t.

$$\begin{array}{ll} B(\tau) & = \tau^5(252 - 1050\tau + 1800\tau^2 - 1575\tau^3 + 700\tau^4 - 126\tau^5) \\ \tau & = \frac{t - t_o}{t_1 - t_o} \end{array} \tag{5.9}$$

We will make some notations. $\vartheta_s = \{\boldsymbol{y}_o = [\underline{y}_o, \overline{y}_o], \boldsymbol{y}_1 = [\underline{y}_1, \overline{y}_1]\}$ contain the specification parameters. $\boldsymbol{y}_o$ and $\boldsymbol{y}_1$ represent the initial and final range of the rotor speed that we wished to define. $\overline{\vartheta}_s = \{y_o = (\overline{y}_o - \underline{y}_o)/2, y_1 = (\overline{y}_1 - \underline{y}_1)/2\}$ is a specification point. $\vartheta_p = \{\vartheta_{p1}, \vartheta_{p2}\}$, $\vartheta_{p1} = \{\boldsymbol{L}_a, \boldsymbol{R}_a, \boldsymbol{D}\}$, $\vartheta_{p2} = \{\boldsymbol{K}, \boldsymbol{J}, \boldsymbol{p}_1\}$ are uncertain intervals of the plant. $\overline{\vartheta}_p = \{\overline{\vartheta}_{p1}, \overline{\vartheta}_{p2}\}$, $\overline{\vartheta}_{p1} = \{L_a, R_a, D\}$, $\overline{\vartheta}_{p2} = \{K, J, p_1\}$ is a nominal plant.

The regions for the flat output can be computed with ϑ_s and t from equation (5.8). The equation is denoted as:

$$\boldsymbol{y} = \boldsymbol{y}_o + (\boldsymbol{y}_1 - \boldsymbol{y}_o)B(\tau) \Rightarrow \gamma_y(t, \vartheta_s) \Rightarrow \gamma_y(t) \tag{5.10}$$

If we consider the hard $\mathbf{y}_o = [0.2, 10]$, $\mathbf{y}_1 = [140, 150]$ and soft $\mathbf{y}_o = [0.1, 12]$, $\mathbf{y}_1 = [138, 152]$ specifications for $t_o = 0$, $t_1 = 1$ and $t := \{t \in R | 0 \leq t \leq 1\}$. The space of flat output will be bounded in time. For instance, in Figure (5.1) the space of flat output for the hard and soft specification is indicated.

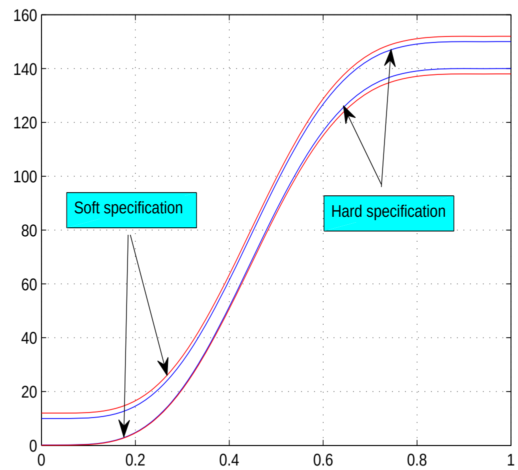


Fig. 5.1. Space of flat output for the hard and soft specification.

However, to reconstruct x_1, u and T_l from equations (5.3), and (5.4), the first and second derivative $(\dot{y}, \ddot{y})$ of the flat output are required. Both are described in equation (5.11).

$$
\begin{aligned}
\dot{y} \;\; &= (y_1 - y_o)\dot{B} \\
\ddot{y} \;\; &= (y_1 - y_o)\ddot{B} \\
\dot{B}(\tau) &= \tau^4(5(252) - (6)(1050)\tau + 7(1800)\tau^2 \\
&\quad - 8(1575)\tau^3 + 9(700)\tau^4 - 10(126)\tau^5) \\
\ddot{B}(\tau) &= \tau^3(4(5)(252) - (5)(6)(1050)\tau + 6(7)(1800)\tau^2 \\
&\quad - 7(8)(1575)\tau^3 + 8(9)(700)\tau^4 - 9(10)(126)\tau^5)
\end{aligned}
\tag{5.11}
$$

Again using the intervals of hard $\mathbf{y}_o = [0.2, 10]$, $\mathbf{y}_1 = [140, 150]$ and soft $\mathbf{y}_o = [0.1, 12]$, $\mathbf{y}_1 = [138, 152]$ specifications and nominal plant $K = 3.4775$, $D = 0.03475$, $J = 0.068$, $L_a = 0.055$, $p_1 = 0.006$ and $Ra = 7.56$ in (5.3),(5.4), and (5.11) and making the corresponding interval operations, we finally reconstructed the bounded intervals of armature's current (x_1), torque load T_l and input signals (u).

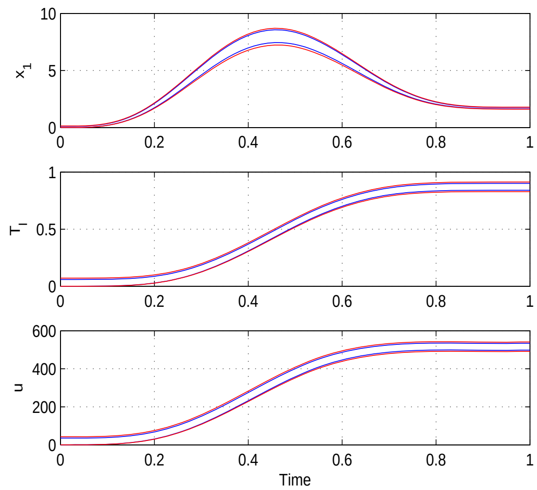


Fig. 5.2. Region of hard and soft trajectories to x_1, T_l and u.

5.1.3 Determining the maximum permissible uncertainty in plant parameters

In the previous problem, we reconstructed the ranges of the state variables and the control signal only with the space of flat output and its derivatives but we do not know the hard and soft uncertainty that could reach the parameters of the process. We suppose that the parameters $\vartheta_{p2} = \{\boldsymbol{K}, \boldsymbol{J}, \boldsymbol{p}_1\}$ considerably do not affect the performance of the system. Therefore, we fixed its nominal values to $\overline{\vartheta}_{p2} = \{K = 3.4775, J = 0.068, p1 = 0.006\}$. We consider that the uncertain parameters $\vartheta_{p1} = \{\boldsymbol{L}_a, \boldsymbol{R}_a, \boldsymbol{D}\}$ could degrade the performance of the system during its operation. Therefore, we want to find the hard and soft plants (permissible maximum uncertainty by the nominal controller).

We want to find the maximum permissible uncertainty in some plant parameters $\vartheta_{p1} = \{\boldsymbol{L}_a, \boldsymbol{R}_a, \boldsymbol{D}\}$ such that the nominal controller $\psi(t, \bar{\vartheta}_s, \bar{\vartheta}_p)$ will ensure ($\exists$) that the output will stay within space of the flat output for all time instant into of the interval t and the constraints are met. A geometric representation of the procedure is indicated in Figure 5.3. The representation is related to the optimization approach described in equation (3.44) from Chapter three.

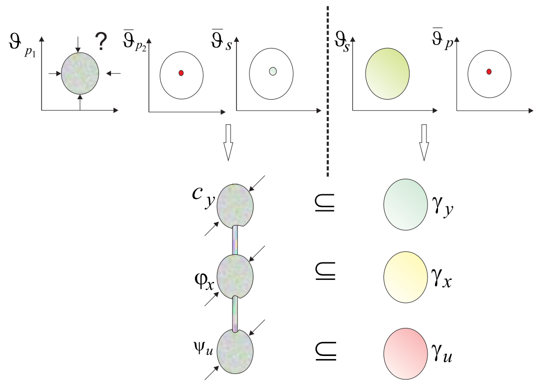


Fig. 5.3. Representation of the interval optimization approach.

With specification intervals $\vartheta_s = \{\boldsymbol{y}_o = [\underline{y}_o, \overline{y}_o], \boldsymbol{y}_1 = [\underline{y}_1, \overline{y}_1])\}$ and the nominal plant $\bar{\vartheta}_p = \{L_a, R_a, D, K, J, p_1\}$, we obtain the region of the trajectories of the flat output $\gamma_y(\vartheta_s, t) \Rightarrow \gamma_y(t)$, the bounding region of state trajectories $\gamma_x(\vartheta_s, \bar{\vartheta}_p, t) \Rightarrow \gamma_x(t)$ and bounding region of control signal $\gamma_u(\vartheta_s, \bar{\vartheta}_p, t) \Rightarrow \gamma_u(t)$. With a fixed specification $\bar{\vartheta}_s = \{y_o = (\bar{y}_o - \underline{y}_o)/2, y_1 = (\bar{y}_1 - \underline{y}_1)/2\}$, we are going to shake(test interval boxes) $\vartheta_{p1} = \{[\underline{L}_a, \overline{L}_a] \times [\underline{R}_a, \overline{R}_a] \times [\underline{D}, \overline{D}]\}$ in the interval functions $c_y(t, \bar{\vartheta}_s)$, $\varphi_x(t, \bar{\vartheta}_s, \bar{\vartheta}_{p2}, \vartheta_{p1})$ and $\psi_u(t, \bar{\vartheta}_s, \bar{\vartheta}_{p2}, \vartheta_{p1})$ such that all constraints $(c_y \subseteq \gamma_y) \wedge (\varphi_x \subseteq \gamma_x) \wedge (\psi_u \subseteq \gamma_u)$ are met. For this case, we are going to see that ϑ_{p1} is within $\varphi_x(t, \bar{\vartheta}_s, \bar{\vartheta}_{p2}, \vartheta_{p1})$ and $\psi_u(t, \bar{\vartheta}_s, \bar{\vartheta}_{p2}, \vartheta_{p1})$ but is not in $c_y(t, \bar{\vartheta}_s)$. So, the values of ϑ_{p1} that we can get, depend of the satisfaction of the bounding regions of γ_x and γ_u. In c_y, φ_x and ψ_u we consider a point of $\vartheta_s = \bar{\vartheta}_s$ because we are evaluating the maximum attainable uncertainty to ϑ_{p1}. In the following section, we are going to obtain the attainable specifications by the nominal controller and we will test specification boxes ϑ_s with a value fixed to $\vartheta_{p1} = \bar{\vartheta}_{p1}$.

The functions γ_y,γ_x,γ_u,c_y,φ_x and ψ_u are described as follows:

$$\begin{aligned}
\gamma_y(\vartheta_s, t) &= y([\underline{y}_o, \overline{y}_o], [\underline{y}_1, \overline{y}_1], t) \Rightarrow \gamma_y(t)\\
\gamma_x(t, \vartheta_s, \bar{\vartheta}_p) &= x_1(D, [\underline{y}_o, \overline{y}_o], [\underline{y}_1, \overline{y}_1], K, J, p_1, t) \Rightarrow \gamma_x(t)\\
\gamma_u(\vartheta_s, \bar{\vartheta}_p, t) &= u(L_a, R_a, D, K, J, p_1, [\underline{y}_o, \overline{y}_o], [\underline{y}_1, \overline{y}_1], t) \Rightarrow \gamma_u(t)\\
c_y(\bar{\vartheta}_s, t) &= x_2(y_o, y_1, t)\\
\varphi_x(\bar{\vartheta}_s, \bar{\vartheta}_{p2}, \vartheta_{p1}, t) &= x_1([\underline{D}, \overline{D}], K, J, p_1, y_o, y_1, t)\\
\psi_u(\bar{\vartheta}_s, \bar{\vartheta}_{p2}, \vartheta_{p1}, t) &= u([\underline{L}_a, \overline{L}_a], [\underline{R}_a, \overline{R}_a], [\underline{D}, \overline{D}], K, J, p_1, y_o, y_1, t)
\end{aligned} \tag{5.12}$$

$[\underline{L}_a, \overline{L}_a] \times [\underline{R}_a, \overline{R}_a] \times [\underline{D}, \overline{D}]$ is the cartesian product of the parameters vector. That is, the points set in the three-dimensional space bounded by the intervals of the parameters. When we started, we must specify an initial box of parameters $[\underline{L}_a, \overline{L}_a]$, $[\underline{R}_a, \overline{R}_a]$ and $[\underline{D}, \overline{D}]$. We verify if the constraints are fulfill with the initial box of parameters. If the constraints are satisfied then we save the parameters box in a solution set. If a parameters box does not meet the constraints and it is outside the solution set, then the parameters box can be saved in a non-solution set. If none of the above steps are met, the width of the parameters box is obtained and we verify if it is less than or equal to certain precision. If the previous condition is verified the box is considered as undefined. If the previous condition is not verified, the parameter box is divided into different boxes and are stored in stack. The following box is read from stack and the previous process is repeated until $stack = 0$. The number of parameters depends on the problem to be solved. The simplest case is when we have a parameter. The parameter is divided by its center and the two resulting intervals are stored in stack. For the case of two parameters, the space is of two dimensions. The box can be divided in four boxes as it is indicated in Figure 5.4. Each box can be easily plotted using its vertices. For example, the vertices for the box number one are: $v_1 = (\underline{p}_1, \underline{p}_2)$, $v_2 = (\underline{p}_1, p_{2c})$, $v_3 = (p_{1c}, p_{2c})$ and $v_4 = (p_{1c}, \underline{p}_2)$. Where p_{1c} and p_{2c} are centers of the intervals of the parameters. A similar procedure is performed for the case of three parameters. A representation is indicated in Figure 5.5. In this case, the eight vertices for the box number one corresponds to $v_1 = (\underline{p}_1, \underline{p}_2, \underline{p}_3)$, $v_2 = (\underline{p}_1, p_{2c}, \underline{p}_3)$, $v_3 = (p_{1c}, p_{2c}, \underline{p}_3)$, $v_4 = (p_{1c}, \underline{p}_2, \underline{p}_3)$, $v_5 = (\underline{p}_1, \underline{p}_2, p_{3c})$, $v_6 = (\underline{p}_1, p_{2c}, p_{3c})$, $v_7 = (p_{1c}, p_{2c}, p_{3c})$, $v_8 = (p_{1c}, \underline{p}_2, p_{3c})$.

So, the set of hard and soft plants $\Sigma_{\forall\exists}$ can be obtained in the following more formal expression:

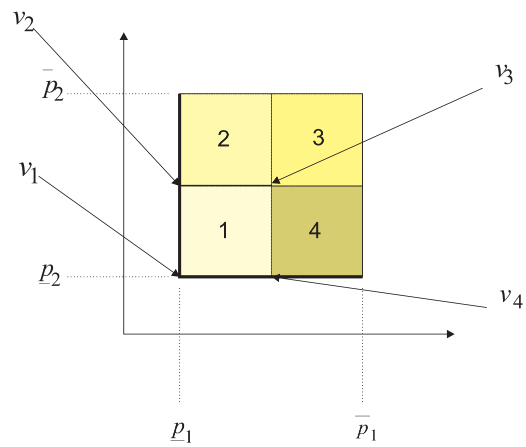


Fig. 5.4. Division of a parameters box.

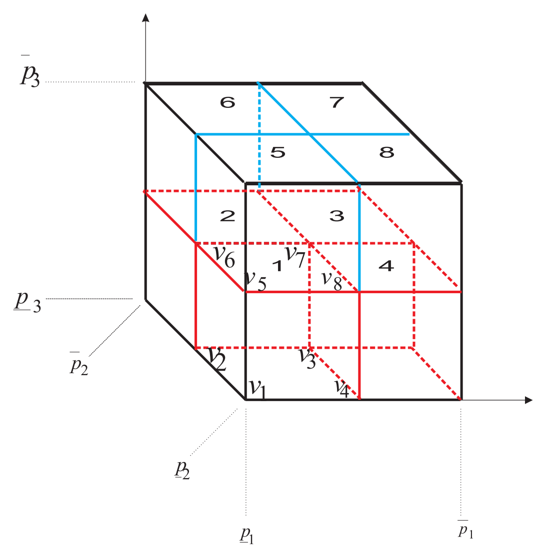


Fig. 5.5. Division for three parameters.

$$
\begin{aligned}
\Sigma_{\forall\exists} = \{ & \vartheta_{p1} | \forall(t \in \boldsymbol{t}^{'}) \forall(\vartheta_{p2} \in \boldsymbol{\vartheta}_{p2}^{'}) \exists(\vartheta_s \in \boldsymbol{\vartheta}_s^{'}) \\
& (c_y(\overline{\vartheta}_s, t) \subseteq \gamma_y(t) \wedge \\
& \varphi_x(\overline{\vartheta}_s, \overline{\vartheta}_{p2}, \vartheta_{p1}, t) \subseteq \gamma_x(t) \wedge \\
& \psi_u(\overline{\vartheta}_s, \overline{\vartheta}_{p2}, \vartheta_{p1}, t) \subseteq \gamma_u(t)\} \\
& \Updownarrow \\
\Sigma_{\forall\exists} = \{ & L_a \times R_a \times D | \forall(t \in \boldsymbol{t}^{'}) \forall(K \in \boldsymbol{K}^{'}) \forall(J \in \boldsymbol{J}^{'}) \forall(p_1 \in \boldsymbol{p}_1^{'}) \\
& \exists(y_o \in \boldsymbol{y}_o^{'}) \exists(y_1 \in \boldsymbol{y}_1^{'})(x_2(y_o, y_1, t) \subseteq \gamma_y(t) \wedge \\
& x_1(D, K, J, p_1, y_o, y_1, t) \subseteq \gamma_x(t) \wedge \\
& u(L_a, R_a, D, K, J, p_1, y_o, y_1, t) \subseteq \gamma_u(t)\}
\end{aligned}
\tag{5.13}
$$

$c_y(\bar{\vartheta}_s, t)$ is equal to the state variable x_2. In this inversion phase from outside to inside we cannot see the effect of the uncertain parameters L_a, R_a, D on $c_y = x_2$. But in internal mode, the uncertain parameters have influence over x_2. So, if we can known the maximum attainable uncertainty on L_a, R_a, D, the nominal controller ensure that x_2 will stay within γ_y under the bounded variation of the uncertain parameters L_a, R_a, D. This is verified by means of robustness tests in next sections.

Finally, the hard and soft plants obtained for both hard $\boldsymbol{y}_o = [0.2, 10]$, $\boldsymbol{y}_1 = [140, 150]$ and soft $\boldsymbol{y}_o = [0.1, 12]$, $\boldsymbol{y}_1 = [138, 152]$ specifications are indicated in Figure 5.6. We programmed in c++ and we used the library to modal intervals (ivalDb), developed by Pau Herrero Viñas **Herrero et al. (2005)**. Solution sets were stored in a data file (.dat) and the results were plotted in Matlab.

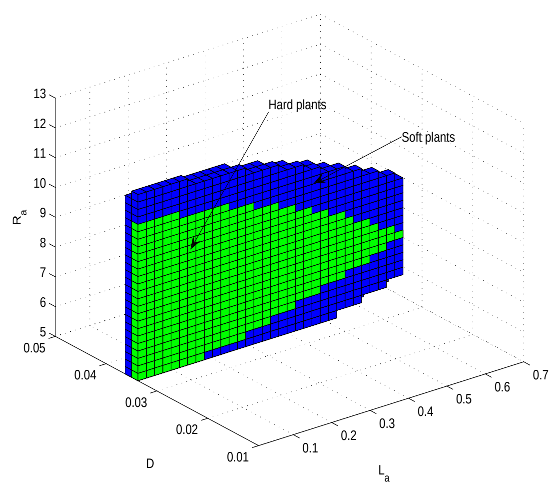


Fig. 5.6. Solution set of hard and soft plants obtained from hard $\boldsymbol{y}_o = [0.2, 10]$, $\boldsymbol{y}_1 = [140, 150]$ and soft $\boldsymbol{y}_o = [0.1, 12]$, $\boldsymbol{y}_1 = [138, 152]$ specifications.

A box of solution of the hard and soft plants are indicated in Table 5.1

Table 5.1. Permissible uncertainty for the parameters of the process

Parameters	hard plants	soft plants
L_a	$[0.01, 0.4]$	$[0.01, 0.7]$
D	$[0.0338, 0.0350]$	$[0.0338, 0.0350]$
R_a	$[6, 8]$	$[6, 9]$

5.1.4 Computation of hard and soft controllers

With the hard and soft plants ϑ_p obtained in previous section and hard and soft specifications ϑ_s we can obtain the hard and soft controllers from equation (5.4) as follows:

$$\begin{aligned} u &= \tfrac{JL_a}{K}\ddot{y} + (\tfrac{L_aD+L_ap_1+R_aJ}{K})\dot{y} + (\tfrac{R_aD}{K} + p_1 + K)y \\ \boldsymbol{u}^* &= \psi(t, \vartheta_p, \vartheta_s) \\ &= \psi(t, [\underline{L}_a, \overline{L}_a], [\underline{R}_a, \overline{R}_a], [\underline{D}, \overline{D}], K, J, p_1, y_o, y_1) \end{aligned} \tag{5.14}$$

Using hard $\boldsymbol{L_a} = [0.01, 0.4]$, $\boldsymbol{R_a} = [6, 8]$, $\boldsymbol{D} = [0.0338, 0.0350]$ and soft $\boldsymbol{L_a} = [0.01, 0.7]$, $\boldsymbol{R_a} = [6, 9]$, $\boldsymbol{D} = [0.0338, 0.0350]$ plants and precise values of $y_o = 5$, $y_1 = 145$, $K = 3.4775$, $J = 0.068$ and $p_1 = 0.006$. We obtain hard and soft controllers depicted in Figure 5.7. It is important to observe that the obtained controllers are including within interval of controllers indicated in Figure 5.2.

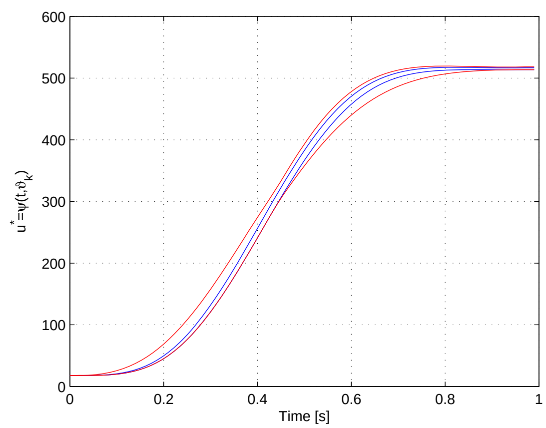


Fig. 5.7. Hard and soft controllers computed with hard $\boldsymbol{L_a} = [0.01, 0.4]$, $\boldsymbol{R_a} = [6, 8]$, $\boldsymbol{D} = [0.0338, 0.0350]$ and soft $\boldsymbol{L_a} = [0.01, 0.7]$, $\boldsymbol{R_a} = [6, 9]$, $\boldsymbol{D} = [0.0338, 0.0350]$ plants and precise values of $y_o = 5$, $y_1 = 145$, $K = 3.4775$, $J = 0.068$ and $p_1 = 0.006$.

From equation (5.14) we can see that in state stable $\dot{y} = 0$ and $\ddot{y} = 0$, so the hard and soft nominal controllers are computed as follows:

$$\boldsymbol{u}^* = (\tfrac{\boldsymbol{R_a D}}{K} + p_1 + K)y \tag{5.15}$$

Using hard $\boldsymbol{L_a} = [0.01, 0.4]$, $\boldsymbol{R_a} = [6, 8]$, $\boldsymbol{D} = [0.0338, 0.0350]$ and soft $\boldsymbol{L_a} = [0.01, 0.7]$, $\boldsymbol{R_a} = [6, 9]$, $\boldsymbol{D} = [0.0338, 0.0350]$ plants and precise values

$y = \bar{y} = 145$, $K = 3.4775$, $J = 0.068$ and $p_1 = 0.006$. We obtain hard and soft nominal controllers indicated in Table 5.2.

Table 5.2. Hard and soft nominal controllers

Parameter	hard controllers	soft controllers
$\boldsymbol{u}^*$	$[513, 515]$	$[513, 518]$

5.1.5 Computation of attainable specifications by a nominal controller

We are going to select a controller from the resulting family of previous section and will try to find the attainable specifications by the selected controller. The space of specifications $[\underline{y}_o, \overline{y}_o] \times [\underline{y}_1, \overline{y}_1]$ will be subdivided and the constraints will be verified. The boxes that fulfill the specifications are stored in the solution set $\Sigma_{\forall\exists}$. The following expression is evaluated.

$$
\begin{aligned}
\Sigma_{\forall\exists} = \{&\vartheta_s | \forall (t \in \boldsymbol{t}')\exists(\vartheta_p \in \boldsymbol{\vartheta}_p')\\
&(c_y(t, \vartheta_s) \subseteq \gamma_y(t) \wedge\\
&\varphi_x(t, \vartheta_s, \overline{\vartheta}_p) \subseteq \gamma_x(t) \wedge\\
&\psi_u(t, \vartheta_s, \overline{\vartheta}_p) \subseteq \gamma_u(t))\}\\
&\Downarrow\\
\Sigma_{\forall\exists} = \{&y_o \times y_1 | \forall (t \in \boldsymbol{t}')\\
&\exists(K \in \boldsymbol{K}')\exists(J \in \boldsymbol{J}')\exists(p_1 \in \boldsymbol{p}_1')\exists(L_a \in \boldsymbol{L}_a')\exists(R_a \in \boldsymbol{R}_a')\exists(D \in \boldsymbol{D}')\\
&(x_2(y_o, y_1, t) \subseteq \gamma_y(t) \wedge\\
&x_1(D, K, J, p_1, y_o, y_1, t) \subseteq \gamma_x(t) \wedge\\
&u(L_a, R_a, D, K, J, p_1, y_o, y_1, t) \subseteq \gamma_u(t)\}
\end{aligned}
\tag{5.16}
$$

The functions $\gamma_y(t)$,$\gamma_x(t)$,$\gamma_u(t)$ are described as follows:

$$
\begin{aligned}
&\gamma_y(t, \vartheta_{ss}) \Rightarrow \gamma_y(t)\\
&\gamma_x(t, \vartheta_{ss}, \bar{\vartheta}_p) \Rightarrow \gamma_x(t)\\
&\gamma_u(t, \vartheta_{ss}, \bar{\vartheta}_p) \Rightarrow \gamma_u(t)
\end{aligned}
\tag{5.17}
$$

Being $\vartheta_{ss} = \{\boldsymbol{y}_{os}, \boldsymbol{y}_{1s}\}$. The parameters and considered controller are indicated in Table 5.3. The results are depicted in Figure 5.8. In this figure we can see that the nominal controller $u^* = 515$ satisfies all the range of specifications.

We can evaluate the specifications that can reach other controllers that are in the core as it is indicated in Figure 5.9.

Table 5.3. Parameters and considered controller

Parameters	Values
R_a	7.56
L_a	0.055
D	0.03475
K	3.4775
J	0.068
u^*	515
p_1	0.006
$\boldsymbol{y}_{os}$	$[0.2, 10]$
$\boldsymbol{y}_{1s}$	$[140, 150]$
$\boldsymbol{y}_o$	$[0, 15]$
$\boldsymbol{y}_1$	$[130, 160]$

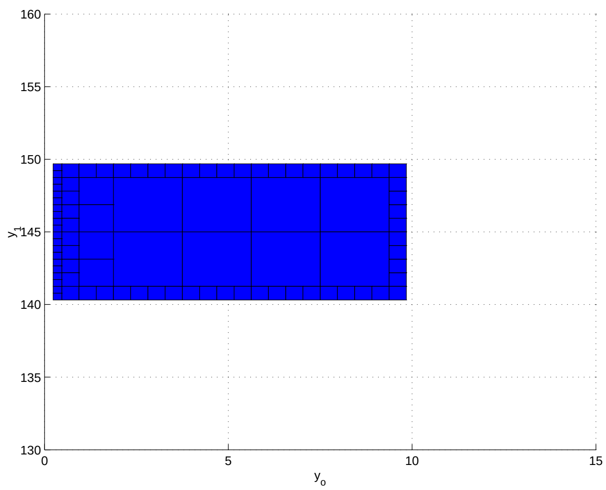


Fig. 5.8. Attainable specifications by the nominal controller $u^* = 515$.

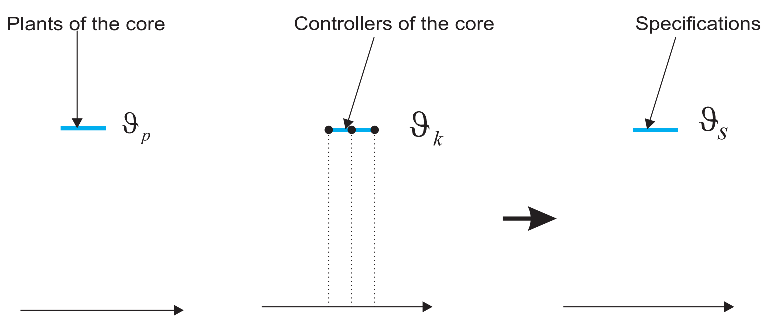


Fig. 5.9. Controllers of the core.

Table 5.4. Parameters

Parameters	Values
R_a	6
L_a	0.01
D	0.0338
K	3.4775
J	0.068
p_1	0.006
u^*	513
$\boldsymbol{y}_{os}$	$[0.2, 10]$
$\boldsymbol{y}_{1s}$	$[140, 150]$
$\boldsymbol{y}_o$	$[0, 15]$
$\boldsymbol{y}_1$	$[130, 160]$

For example, we will take the controller from the left end of the core considering the parameters of the Table 5.4.

The results indicated in Figure 5.10 we can observe that nominal controller $u^* = 513$ can reach only a partial region of specifications.

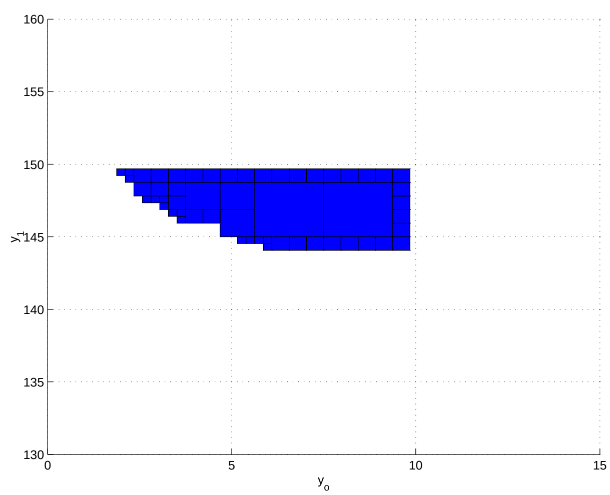


Fig. 5.10. Attainable specifications by the controller $u^* = 513$.

On similar way, now we evaluate the specifications that can fulfill the controller of the right end of the core. The considered parameters are indicated in Table 5.5.

We can verify the results indicated in Figure 5.11 that the corresponding controller also satisfies a determined region of the specifications.

Table 5.5. Parameters

Parameters	Values
R_a	8
L_a	0.4
D	0.0350
K	3.4775
J	0.068
p_1	0.006
u^*	518
$\boldsymbol{y}_{os}$	$[0.2, 10]$
$\boldsymbol{y}_{1s}$	$[140, 150]$
$\boldsymbol{y}_o$	$[0, 15]$
$\boldsymbol{y}_1$	$[130, 160]$

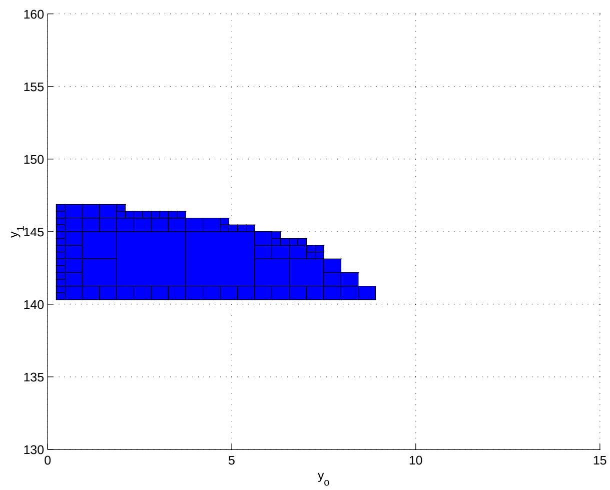


Fig. 5.11. Attainable specifications by the controller $u^* = 518$.

Evaluating a controller that is outside the core with the parameters indicated in Table 5.6, we verified that the specifications are not satisfied.

5.1.6 Robustness test

In this section, we are going to realize a robustness test to the family of controllers $\boldsymbol{u}^* = [513, 518]$ verifying that they fulfill the specifications under parametric uncertainty in the process. We will use different controllers from the family $\boldsymbol{u}^* = [513, 518]$ to control the system in open-loop as it is indicated in Figure 5.12 introducing at certain instant time the variations of the process parameters.

We considered the intervals of the parameters indicated in Table 5.1. In Figure 5.13 it is verified that all the controllers maintain the rotor speed

Table 5.6. Parameters

Parameters	Values
R_a	11
L_a	0.9
D	0.04
K	3.4775
J	0.068
p_1	0.006
u	522
$\boldsymbol{y}_{os}$	$[0.2, 10]$
$\boldsymbol{y}_{1s}$	$[140, 150]$
$\boldsymbol{y}_o$	$[0, 15]$
$\boldsymbol{y}_1$	$[130, 160]$

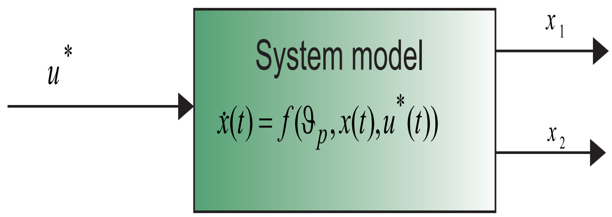


Fig. 5.12. Open-loop control system.

$y = x_2$ into the specifications interval considering the variation of the plant parameters.

5.1.7 Designing a state feedback controller

In this section, we are going to apply the second phase from design procedure. We are going to determine the state feedback control law and its hard and soft parameters such that the rotor speed is within of the region of hard and soft specifications under hard and soft uncertainty of the plant. In the feedback control law, we are going to use a nominal feedforward that was determined in previous section and a robustness test of the feedback controller will be verified. From equation (5.4) we have the expression for the feedforward controller

$$u = \frac{L_a}{K}(J\dddot{y} + D\ddot{y} + p_1\dot{y}) + \frac{R_a}{K}(J\dot{y} + Dy + p_1 y) + Ky \tag{5.18}$$

in terms of the state variables we have that $\dot{x}_1 = \frac{1}{K}(J\ddot{y} + D\dot{y} + p_1 y)$, $x_1 = \frac{1}{K}(J\dot{y} + Dy + p_1 y)$ and $y = x_2$, so

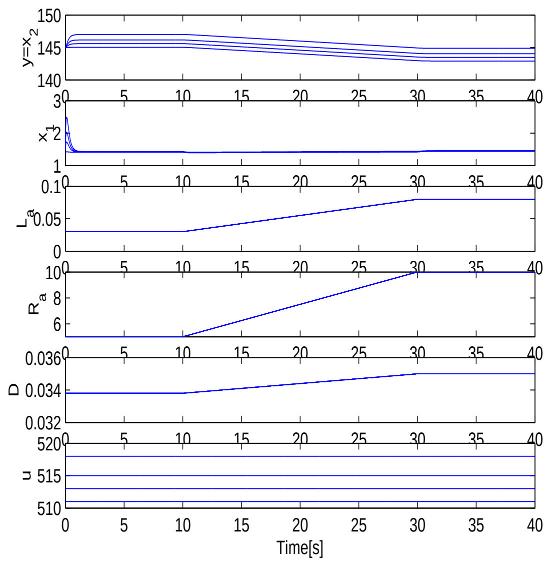


Fig. 5.13. Robustness test.

$$u = L_a \dot{x}_1 + R_a x_1 + K x_2 \tag{5.19}$$

from flat output y and its derivatives $\dot{y}$ and $\ddot{y}$ we can select the coordinate $x_2 = y$, and $x_1 = \dot{y}$. So, $\dot{x}_1 = \ddot{y}$. With the new input of control $v = \ddot{y}$ we have the following expression:

$$u = L_a v + R_a x_1 + K x_2 \tag{5.20}$$

The closed-loop dynamics for the flat output can be stablished with the second order equation $\ddot{y} + k_2 \dot{y} + k_1(y - \bar{y}) = 0$. Which can be made asymtotically stable by a suitable choice of the design parameters k_2 and k_1. We are going to search the set of hard and soft parameters to k_2 and k_1, such that the desired specifications are met. Closed-loop dynamics can be expressed in terms of the state variables as follows:

$$\ddot{y} = -k_2 \dot{x}_2 - k_1(x_2 - \bar{x}_2) \tag{5.21}$$

as $\dot{x}_2 = \frac{K}{J} x_1 - \frac{D}{J} x_2 - \frac{T_l}{J}$ thus

$$\ddot{y} = -k_2\left(\frac{K}{J} x_1 - \frac{D}{J} x_2 - \frac{T_l}{J}\right) - k_1(x_2 - \bar{x}_2) \tag{5.22}$$

replacing $v = \ddot{y}$ from equation (5.22) in (5.20), we obtain the state feedback control law as follows:

$$u = L_a(-k_2(\frac{K}{J}x_1 - \frac{D}{J}x_2 - \frac{T_l}{J}) - k_1(x_2 - \bar{x}_2)) + R_a x_1 + K x_2 \tag{5.23}$$

The state feedback controller can be expressed in function of the nominal feedforward (u^*) determined with equation (5.15) of the following way:

$$\begin{aligned} u &= u^* + L_a v + R_a x_1 + K x_2 - (\frac{R_a D}{K} + p_1 + K)\bar{x}_2 \\ u &= u^* - x_1(-R_a + \frac{L_a k_2 K}{J}) - x_2(-K + L_a k_1 + \frac{L a k_2 D}{J}) - \\ &\quad \bar{x}_2(-L_a k_1 + (\frac{R_a D}{K} + p_1 + K)) - \frac{L_a k_2 T_l}{J} \end{aligned} \tag{5.24}$$

now, let us define a desired region (closed-loop reference model) in time with the following interval function:

$$M(t, \theta_q) = \boldsymbol{\theta}_{q1}(1 - \boldsymbol{\theta}_{q2} exp(-\boldsymbol{\theta}_{q3} t)) \tag{5.25}$$

We desired that the output of the feedback system is within reference model $M(t, \theta_q)$. Where $\boldsymbol{\theta}_{q1}$ is the interval of the hard and soft speficifications, $\boldsymbol{\theta}_{q2}$ and $\boldsymbol{\theta}_{q3}$ are intervals to fix the response speed in time of the function $M(t, \theta_q)$, t is a time interval. If we define the hard $\boldsymbol{\theta}_{q1} = [140, 150]$ and soft $\boldsymbol{\theta}_{q1} = [138, 152]$ specification and $\boldsymbol{\theta}_{q2} = [0.5, 1]$, $\boldsymbol{\theta}_{q3} = [0.2, 1]$ and $t := \{t \in R | 0 \leq t \leq 40\}$ the bound interval of the hard and soft interval function $M(t, \theta_q)$ is depicted in Figure 5.14.

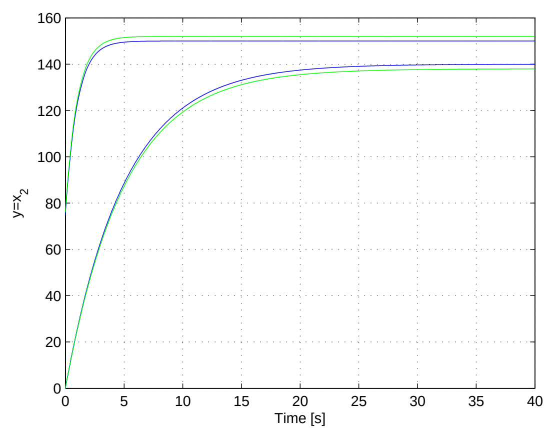


Fig. 5.14. Interval function $M(t, \theta_q)$.

In Figure (5.15) we make a geometric representation of the interval optimization approach to feedback dynamic system. We applied a nominal feedforward $u^* = 515$ (determined in the previous section) in the state feedback law (5.24) and we found a set of boxes $\vartheta_c = \{[\underline{k}_1, \overline{k}_1], [\underline{k}_2, \overline{k}_2]\}$ of the feedback controller such that the trajectory of the state variable $x_2 = c_{x2} = \mu_c l(t, \vartheta_p, \vartheta_c)$ is within the limits of the reference model $\gamma_y = M(t, \theta_q)$.

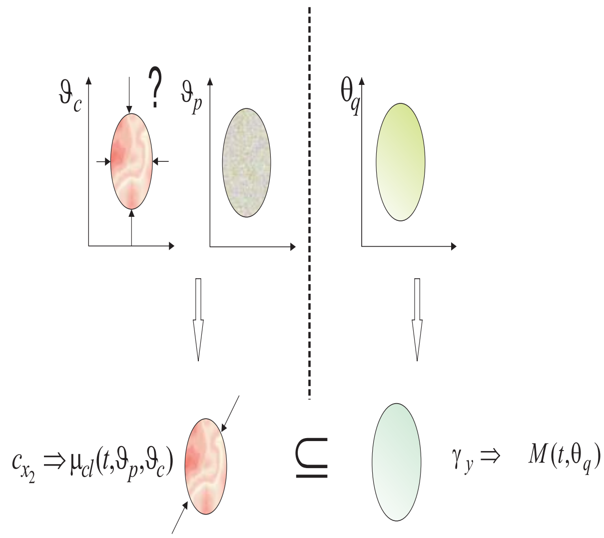


Fig. 5.15. Representation of the interval optimization approach of the feedback dynamic system.

A more formal expression can be expressed as follows:

$$\begin{aligned}
\Sigma_{\forall\exists} = \; & \{\theta_c]|\forall(t \in \boldsymbol{t}')\forall(\vartheta_{p1} \in \boldsymbol{\vartheta}'_{p1})\forall(\vartheta_{p2} \in \boldsymbol{\vartheta}'_{p2})\exists(\theta_q \in \boldsymbol{\theta}'_q) \\
& (\mu_{cl}(t, \vartheta_{p1}, \overline{\vartheta}_{p2}, \theta_c) \subseteq M(t, \vartheta_q))\} \\
& \Downarrow \\
\Sigma_{\forall\exists} = \; & \{k_1 \times k_2|\forall(t \in \boldsymbol{t}')\forall(L_a \in \boldsymbol{L}'_a)\forall(R_a \in \boldsymbol{R}'_a)\forall(D \in \boldsymbol{D}') \\
& \forall(K \in \boldsymbol{K}')\forall(J \in \boldsymbol{J}')\exists(\theta_{q1} \in \boldsymbol{\theta}'_{q1})\exists(\theta_{q2} \in \boldsymbol{\theta}'_{q2})\exists(\theta_{q3} \in \boldsymbol{\theta}'_{q3}) \\
& (x_2(K, J, L_a, R_a, D, k_1, k_2, t) \subseteq M(\theta_{q1}, \theta_{q2}, \theta_{q3}, t))\}
\end{aligned} \tag{5.26}$$

where the parameters $\theta_q = \{\boldsymbol{\theta}_{q1}, \boldsymbol{\theta}_{q2}, \boldsymbol{\theta}_{q3}\}$ are existentially quantified $\exists$, because we want to ensure that x_2 is within $M(t, \theta_q)$. In Table 5.7 the values of the parameters are indicated.

Set of hard and soft parameters k_1, k_2 of the state feedback controller are depicted in Figure 5.16 and 5.17.

We will perform some robustness tests. From the paving indicated in Figure 5.16, three parameters from the feedback controller will be selected

Table 5.7. Parameters

Parameters	Values
$\boldsymbol{R}_a$	$[6, 9]$
$\boldsymbol{L}_a$	$[0.01, 0.7]$
$\boldsymbol{D}$	$[0.338, 350]$
K	3.4775
J	0.068
u^*	515
p_1	0.006
$\boldsymbol{\theta}_{q1}(soft)$	$[140, 150]$
$\boldsymbol{\theta}_{q1}(hard)$	$[138, 152]$
$\boldsymbol{\theta}_{q2}$	$[0.5, 1]$
$\boldsymbol{\theta}_{q3}$	$[0.2, 1]$
$\boldsymbol{k}_1$	$[0.01, 5]$
$\boldsymbol{k}_2$	$[0.01, 5]$

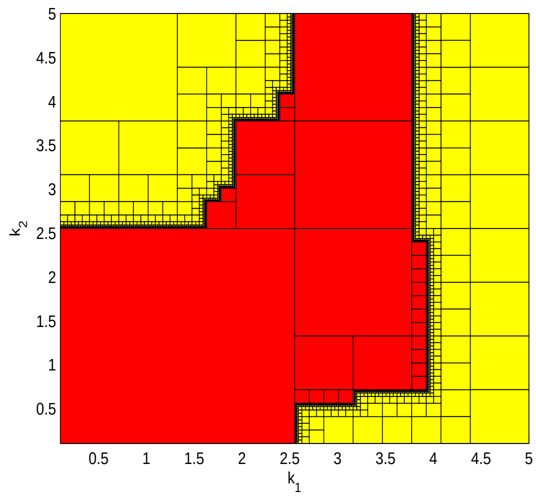


Fig. 5.16. Set of hard parameters k_1, k_2 of the state feedback controller determined with $u^* = 515$, $\boldsymbol{\vartheta}_{q1} = [140, 150]$, $\boldsymbol{\vartheta}_{q1} = [0.5, 1]$, $\boldsymbol{\vartheta}_{q2} = [0.2, 1]$. The red boxes represent the solution set. The yellow boxes are outside of the solution set and the black boxes are undefined.

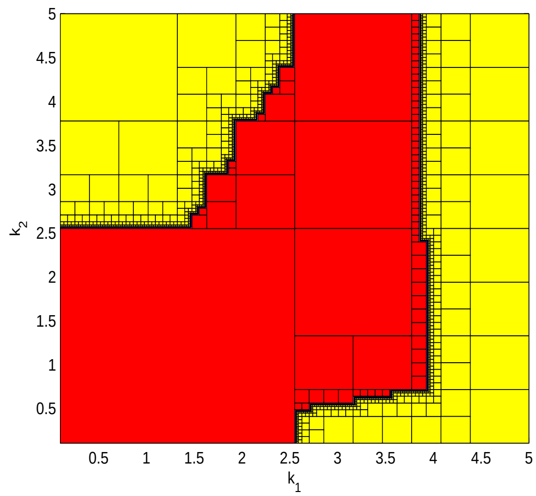


Fig. 5.17. Set of soft parameters k_1, k_2 of the state feedback controller determined with $u^* = 515$, $\boldsymbol{\vartheta}_{q1} = [138, 152]$, $\boldsymbol{\vartheta}_{q1} = [0.5, 1]$, $\boldsymbol{\vartheta}_{q2} = [0.2, 1]$.

$\vartheta_{c1} = \{k_1 = 1.5, k_2 = 1.5\}$, $\vartheta_{c2} = \{k_1 = 3.2, k_2 = 3\}$ and $\vartheta_{c3} = \{k_1 = 4, k_2 = 1.5\}$. ϑ_{c1} and ϑ_{c2} are within the paving and ϑ_{c3} is out of it. With a nominal feedforward $u^* = 515$, the system will be controlled in an interval of time $\{0 \leq t \leq 40\}$. From $t = 10$ to $t = 20$, parameters L_a,D and R_a will be variated within the intervals $\boldsymbol{L}_a = [0.01, 0.7]$, $\boldsymbol{D} = [0.0338, 0.0350]$, $\boldsymbol{R}_a = [6, 9]$. These values belong to maximum attainable uncertainty for the soft specification as indicated in Table 5.1. The load torque also will be variated within the interval $\boldsymbol{Tl} = [0, 0.9]$ from $t = 0$ to $t = 20$. In Figures 5.18 and 5.19, we can see that with parameters $\vartheta_{c1} = \{k_1 = 1.5, k_2 = 1.5\}$ and $\vartheta_{c2} = \{k_1 = 3.2, k_2 = 3\}$, the feedback controller keep the output within $M(t, \theta_q)$ for all $0 \leq t \leq 40$, obtaining a good robust performance under parametric uncertainty. In Figure 5.20, we can see that with parameters $\vartheta_{c3} = \{k_1 = 4, k_2 = 1.5\}$, the feedback controller does not meet the specifications in certain instant of time.

After performing different robustness tests, we determined that the feedback controller presented better robustness performance using parameters located around of the paving centers. A representation is indicated in Figure 5.21. In this manner, the robust controller designed under this procedure, gurantees that the ouput of the system will be placed within the specifications for all future time if the uncertain parameters change within the permissible limits.

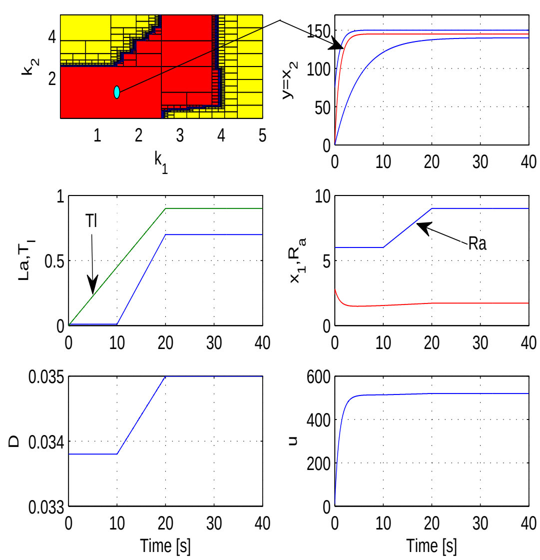


Fig. 5.18. Robustness test of the state feedback controller with $\{k_1 = 1.5\}$, $\{k_2 = 1.5\}$, nominal feedforward $u^* = 515$, soft uncertainty $\boldsymbol{L_a} = [0.01, 0.7]$, $\boldsymbol{D} = [0.0338, 0.0350]$, $\boldsymbol{R_a} = [6, 9]$ and load torque $\boldsymbol{Tl} = [0, 0.9]$.

5.2 Applications to simple pendulum

5.2.1 Dynamic model of the simple pendulum

Consider the simple pendulum in Figure 5.22 **Slotine and Weiping (1991)**. The dynamic is given by the nonlinear equation (5.27)

$$MR^2\ddot{\theta} + b\dot{\theta} + MgR\,sin(\theta) = \tau \tag{5.27}$$

where R is the pendulum's length, M its mass, b the friction coefficient at the hinge, g the gravity constant and τ the control input. Letting $x_1 = \theta$ and $x_2 = \dot{\theta}$, the corresponding state-space equation is indicated in equation (5.28).

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \tfrac{1}{MR^2}(-bx_2 - MgR\,sin(x_1) + \tau) \end{aligned} \tag{5.28}$$

We are going to suppose that the flat output is the displacement angle $y = x_1$. The parametrization of the state variables and control input in function of the flat output and its derivatives is indicated in equation (5.29).

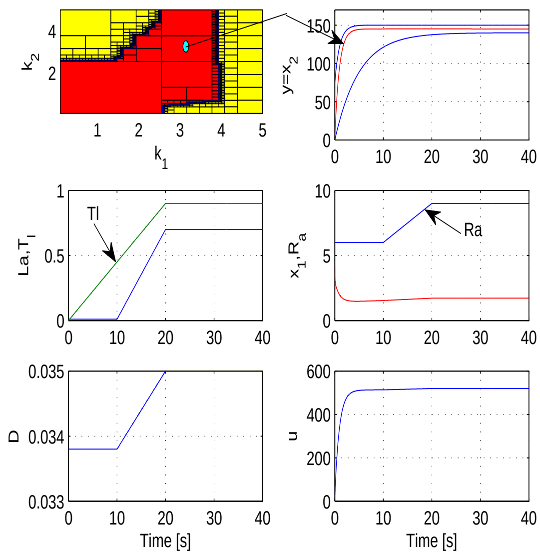


Fig. 5.19. Robustness test of the state feedback controller with $\{k_1 = 3.2\}$, $\{k_2 = 3\}$, feedforward nominal $u^* = 515$ soft uncertainty $\boldsymbol{L}_a = [0.01, 0.7]$, $\boldsymbol{D} = [0.0338, 0.0350]$, $\boldsymbol{R}_a = [6, 9]$ and load torque $\boldsymbol{Tl} = [0, 0.9]$.

$$\begin{aligned} x_1 &= y \\ x_2 &= \dot{y} \\ \tau &= MR^2\ddot{y} + b\dot{y} + MRg\, sin(y) \end{aligned} \tag{5.29}$$

5.2.2 Simulation of the system considering a region of flat output

Of similar way as in the linear case, we can define a space of the trajectory of the displacement angle considering equations 5.30 and 5.31

$$y = y_o + (y_1 - y_o)B(\tau) \tag{5.30}$$

$$\begin{aligned} B(\tau) &= \tau^5(252 - 1050\tau + 1800\tau^2 - 1575\tau^3 + 700\tau^4 - 126\tau^5) \\ \tau &= \frac{t - t_o}{t_1 - t_o} \end{aligned} \tag{5.31}$$

If we considered the hard $\boldsymbol{y}_o = [0.05, 0.2]$, $\boldsymbol{y}_1 = [1.1, 1.2]$ and soft $\boldsymbol{y}_o = [0, 0.3]$, $\boldsymbol{y}_1 = [1, 1.3]$ specifications for $t_o = 0$, $t_1 = 1$ and $t := \{t \in R | 0 \leq t \leq 1\}$ the flat output space will be bounded in time as it is indicated in Figure (5.23).

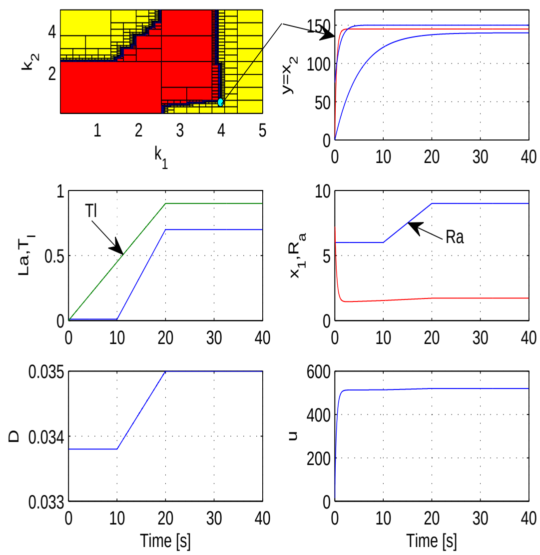


Fig. 5.20. Robustness test of the state feedback controller with $\{k_1 = 1.5\}$, $\{k_2 = 1.5\}$ and feedforward nominal $u^* = 515$ soft uncertainty $\boldsymbol{L_a} = [0.01, 0.7]$, $\boldsymbol{D} = [0.0338, 0.0350]$, $\boldsymbol{R_a} = [6, 9]$ and load torque $\boldsymbol{Tl} = [0, 0.9]$.

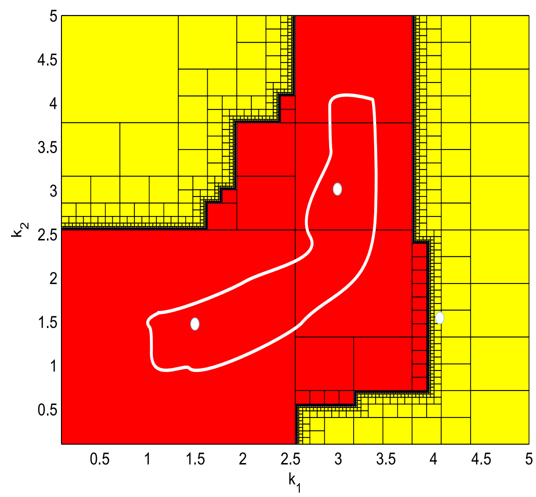


Fig. 5.21. Parameters of the feedback controller located in the robustness region.

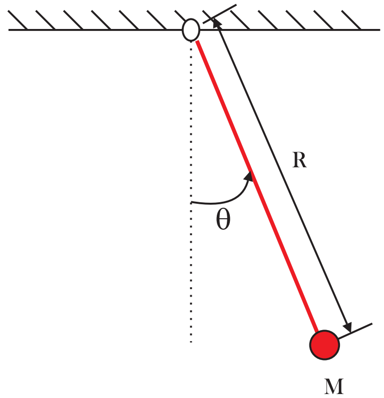


Fig. 5.22. Simple pendulum.

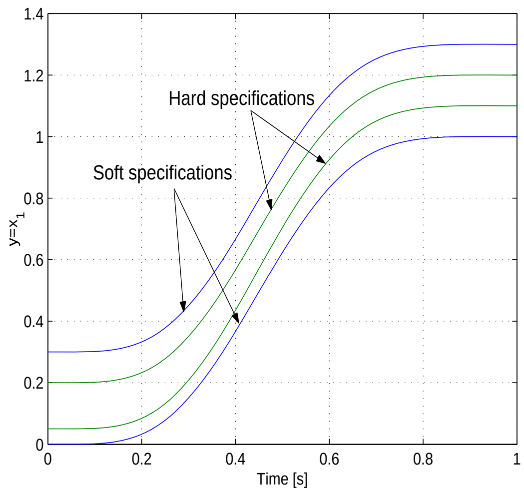


Fig. 5.23. Space of hard and soft specifications.

Using intervals hard $\boldsymbol{y}_o = [0.05, 0.2]$, $\boldsymbol{y}_1 = [1.1, 1.2]$ and soft $\boldsymbol{y}_o = [0, 0.3]$, $\boldsymbol{y}_1 = [1, 1.3]$ and nominal parameters for the pendulum $M = 0.15$, $R = 0.25$, $b = 0.007$, $g = 10$ in equations (5.29) and (5.29) and making the corresponding interval operations, we finally reconstructed bounding regions of the displacement speed (x_2) and input torque (τ).

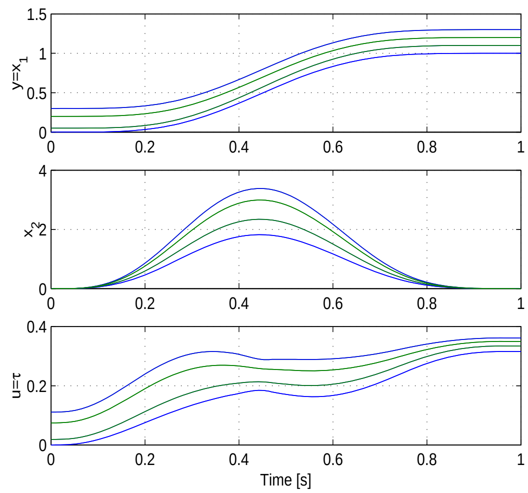


Fig. 5.24. Space of flat output y, state variable x_2 and input torque $u = \tau$.

5.2.3 Determining the maximum permissible uncertainty in plant parameters

From parameters set of the pendulum, we are going to determine the hard and soft maximum uncertainty that could reach the friction coefficient b. We are going to consider that the specifications spaces for the flat output y, state x_2 and input torque τ are the represented ones in Figure 5.24. We emphasize that these spaces were obtained with equations (5.30) and (5.29) for hard $\boldsymbol{y}_o = [0.05, 0.2]$, $\boldsymbol{y}_1 = [1.1, 1.2]$ and soft $\boldsymbol{y}_o = [0, 0.3]$, $\boldsymbol{y}_1 = [1, 1.3]$ specifications and nominal parameters $M = 0.15$, $R = 0.25$, $b = 0.007$, $g = 10$. Thus, the specifications space, states and input torque can be expressed like $E_{sy}(t)$, $E_{sx_2}(t)$ and $E_{s\tau}(t)$.

$$
\begin{aligned}
x_1 &= y \Rightarrow E_{sy}([\underline{y}_o, \overline{y}_o], [\underline{y}_1, \overline{y}_1], t) \Rightarrow E_{sy}(t) \\
x_2 &= \dot{y} \Rightarrow E_{sx_2}([\underline{y}_o, \overline{y}_o], [\underline{y}_1, \overline{y}_1], t) \Rightarrow E_{sx_2}(t) \\
\tau &= MR^2\ddot{y} + b\dot{y} + MRg\ sin(y) \Rightarrow E_{s\tau}([\underline{y}_o, \overline{y}_o], [\underline{y}_1, \overline{y}_1], M, R, b, g, t) \\
&\Rightarrow E_{s\tau}(t)
\end{aligned}
\tag{5.32}
$$

On the other hand, to compute the set of boxes $[\underline{b}, \overline{b}]$ that satisfy the specifications of equations (5.32), we will use the same equations 5.30 and 5.29 with the distinction to evaluate them with a point of y_o, a point of y_1, a nominal plant $M = 0.15$, $R = 0.25$, $b = 0.007$, $g = 10$ and a set of possible boxes $[\underline{b}, \overline{b}]$. Therefore, we can write the following expression.

$$\begin{aligned} x_1 &= y \Rightarrow f_{sy}(y_o, y_1, t) \\ x_2 &= \dot{y} \Rightarrow f_{sx_2}(y_o, y_1, t) \\ \tau &= MR^2\ddot{y} + b\dot{y} + MRg\, sin(y) \Rightarrow f_{s\tau}(y_o, y_1, M, R, [\underline{b}, \overline{b}], g, t) \end{aligned} \tag{5.33}$$

having the solution set $\Sigma_{\forall\exists}$ the following representation

$$\begin{aligned} \Sigma_{\forall\exists} = \{&\vartheta_{p1} | \forall(t \in \boldsymbol{t}') \forall(\vartheta_{p2} \in \boldsymbol{\vartheta}'_{p2}) \exists(\vartheta_s \in \boldsymbol{\vartheta}'_s) \\ &(c_y(t, \overline{\vartheta}_s) \subseteq \gamma_y(t) \wedge \\ &\varphi_x(t, \overline{\vartheta}_s) \subseteq \gamma_x(t) \wedge \\ &\psi_u(t, \overline{\vartheta}_s \overline{\vartheta}_{p2}, \vartheta_{p1}) \subseteq \gamma_u(t))\} \\ &\Downarrow \\ \Sigma_{\forall\exists} = \{&b | \forall(t \in \boldsymbol{t}') \forall(M \in \boldsymbol{M}') \forall(R \in \boldsymbol{R}') \forall(g \in \boldsymbol{g}') \exists(y_o \in \boldsymbol{y}'_o) \exists(y_1 \in \boldsymbol{y}'_1) \\ &(f_{sy}(y_o, y_1, t) \subseteq E_{sy}(t) \wedge \\ &f_{sx_2}(y_o, y_1, t) \subseteq E_{sx_2}(t) \wedge \\ &f_{s\tau}(y_o, y_1, M, R, b, g, t) \subseteq E_{s\tau}(t))\} \end{aligned} \tag{5.34}$$

Being $\vartheta_p = \{\vartheta_{p1}, \vartheta_{p2}\}$, $\vartheta_{p1} = \{\boldsymbol{b}\}$, $\vartheta_{p2} = \{\boldsymbol{M}, \boldsymbol{R}, \boldsymbol{g}\}$, $\overline{\vartheta}_{p2} = \{M, R, g\}$, $\vartheta_s = \{\boldsymbol{y}_o, \boldsymbol{y}_1\}$, $\overline{\vartheta}_s = \{y_o, y_1\}$, $\gamma_y(t) = E_{sy}(t)$, $\gamma_x(t) = E_{sx_2}(t)$, $\gamma_u(t) = E_{s\tau}(t)$, $c_y = f_{sy}$, $\varphi_x = f_{sx_2}$ and $\psi_u = f_{s\tau}$.

Hard and soft plants (uncertainty) for the friction coefficient b is indicated in Table 5.8

Table 5.8. Permissible uncertainty for the friction coefficient b

Parameter	Hard plants	Soft plants
b	$[0.001, 0.0139]$	$[0.001, 0.0259]$

5.2.4 Computation of a family of controllers

With the intervals of hard and soft uncertainty ϑ_p obtained in the previous section and with hard and soft specifications ϑ_s we can obtain the family of controllers with the following equation:

$$\begin{aligned} \tau^* &= \vartheta_k(t, \vartheta_p, \vartheta_s) \\ &= \vartheta_k(t, [\underline{b}, \overline{b}], M, R, g, y_o, y_1) \end{aligned} \tag{5.35}$$

The equation (5.35) is obtained from equation (5.33). Using the intervals of hard $\boldsymbol{b} = [0.001, 0.0139]$ and soft $\boldsymbol{b} = [0.001, 0.0259]$ uncertainty and precise values of $y_o = 0.15$, $y_1 = 1.15$, $M = 0.15$, $R = 0.25$, $g = 10$ we obtain the results indicated in Figure 5.25. It is important to observe that the obtained controllers are including within the interval of controllers obtained initially in Figure (5.24). In this case, we found a unique nominal controller indicated in Table 5.9.

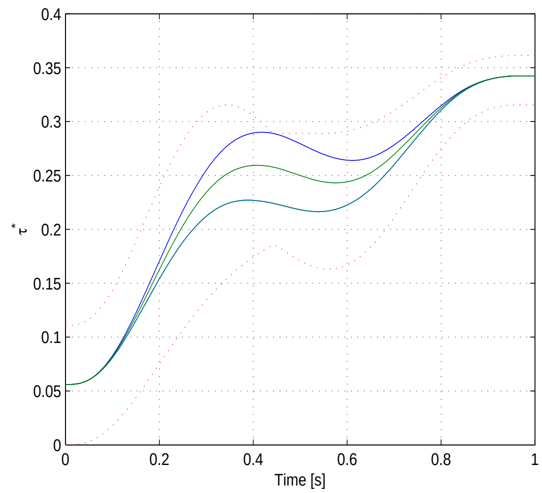


Fig. 5.25. Hard and soft controllers.

Table 5.9. Hard and soft controller τ^*

Parameter	Hard controller	Soft controller
τ^*	0.342	0.342

5.2.5 Computation of attainable specifications by a nominal controller

We are going to select the controller of the previous section and will try to find the attainable specifications by the controller. The specifications space $[\underline{y}_o, \overline{y}_o] \times [\underline{y}_1, \overline{y}_1]$ will be subdivided and the constraints will be verified. The boxes that fulfill the specifications keep in the solution set $\Sigma_{\forall\exists}$ verifying the following expression:

$$\begin{aligned}
\Sigma_{\forall\exists} = \{&\vartheta_s | \forall (t \in \boldsymbol{t}') \exists (\vartheta_p \in \boldsymbol{\vartheta}'_p) \\
&(c_y(t, \vartheta_s) \subseteq \gamma_y(t) \wedge \\
&\varphi_x(t, \vartheta_s) \subseteq \gamma_x(t) \wedge \\
&\psi_u(t, \vartheta_s, \overline{\vartheta}_p) \subseteq \gamma_u(t))\} \\
&\Downarrow \\
\Sigma_{\forall\exists} = \{&y_o \times y_1 | \forall (t \in \boldsymbol{t}') \exists (M \in \boldsymbol{M}') \exists (R \in \boldsymbol{R}') \exists (g \in \boldsymbol{g}') \exists (b \in \boldsymbol{b}') \\
&(f_{sy}(t, y_o, y_1) \subseteq E_{sy}(t) \wedge \\
&f_{sx_2}(t, y_o, y_1) \subseteq E_{sx_2}(t) \wedge \\
&f_{s\tau}(t, y_o, y_1, M, R, b, g) \subseteq E_{s\tau}(t))\}
\end{aligned} \tag{5.36}$$

The parameters and considered controller are indicated in Table 5.10.

Table 5.10. Parameters

Parameters	Values
R	0.25
M	0.15
g	10
b	0.0124
τ	0.342
$\boldsymbol{y}_{os}$	$[0.1, 0.2]$
$\boldsymbol{y}_{1s}$	$[1.1, 1.2]$
$\boldsymbol{y}_{o}$	$[0, 0.4]$
$\boldsymbol{y}_{1}$	$[1, 1.4]$

being $b = (0.0259 - 0.001)/2 = 0.0124$ the midpoint of its interval and $y([\underline{y}_{os}, \overline{y}_{os}], [\underline{y}_{1s}, \overline{y}_{1s}], t)$ a wished solution region bounded by $\boldsymbol{y}_{os}$, $\boldsymbol{y}_{1s}$. The result is indicated in Figure 5.26.

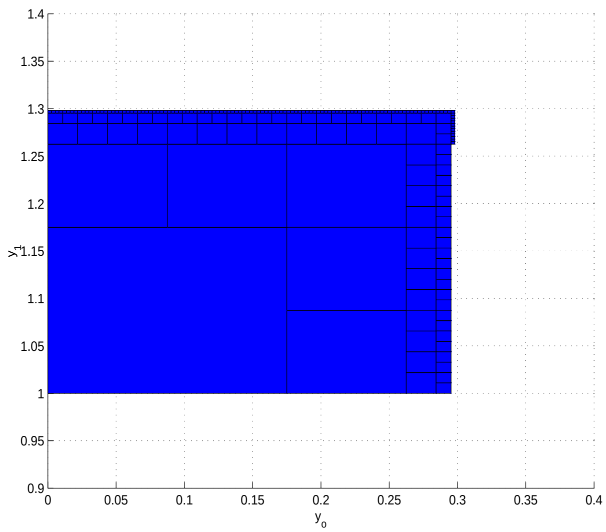


Fig. 5.26. Attainable specifications for nominal controller $\tau = 0.342$.

5.2.6 Robustness test

In this section, we wish to realize a robustness test to controller $\boldsymbol{\tau} = [0.342, 0.342]$ verifying the fulfillment of specifications under parametric uncertainty $\boldsymbol{b} = [0.001, 0.0259]$. For initial conditions near the steady state $x_1 = 1.13$ and $x_2 = 0.01$, in Figure 5.27 it is verified that the controller maintain the position $y = x_1$ into the specification $\boldsymbol{x}_1 = [1, 1.3]$ after a transition of the system states.

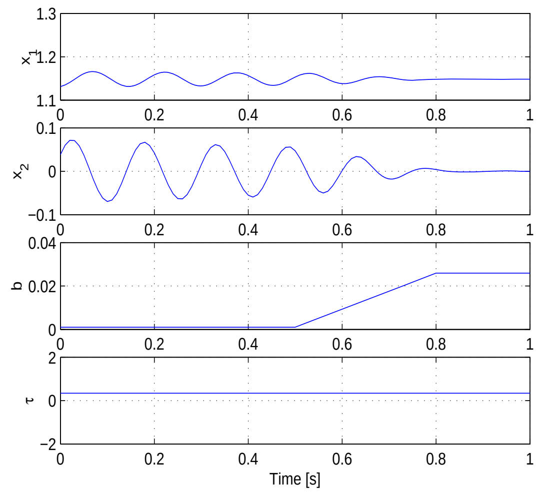


Fig. 5.27. Robustness test with controller $\tau = 0.342$.

5.2.7 Designing a state feedback controller

In this section, we are going to design a state feedback controller in order to stabilize the system. From equation (5.29) the input torque τ was deduced as:

$$\tau = MR^2\ddot{y} + b\dot{y} + MRg\ sin(y) \tag{5.37}$$

suppose it is desired to drive the state $y = x_1$ to its equilibrium value given by, $\overline{y} = \overline{x}_1$. The closed-loop dynamics for the flat output is indicated in equation (5.38)

$$y^{(2)} + k_2\dot{y} + k_1(y - \overline{y}) \tag{5.38}$$

which can be made asymptotically stable by a suitable choice of the design parameters k_2 and k_1. Replacing $y^{(2)}$ from equation (5.38) in equation (5.37) we obtain:

$$\tau = MR^2(-k_2\dot{y} - k_1(y - \overline{y})) + b\dot{y} + MRg\ sin(y) \tag{5.39}$$

as $y = x_1$, $\dot{y} = x_2$ and $\overline{y} = \overline{x}_1$ then, equation 5.39 in terms of state variables is

$$\tau = MR^2(-k_2x_2 - k_1(x_1 - \overline{x}_1)) + bx_2 + MRg\ sin(x_1) \tag{5.40}$$

we can also express the stabilizing controller of equation 5.40, in terms of the nominal controller τ^* (see Table 5.9) of the following form:

$$\tau = \tau^* + MR^2(-k_2x_2 - k_1(x_1 - \overline{x}_1)) + bx_2 + MRg\ (sin(x_1) - sin(\overline{x}_1)) \tag{5.41}$$

the nominal controller τ^* can also be obtained from equation (5.39) considering that $\dot{y} = 0$ and $y = \bar{y}$ then

$$\tau^* = MRg\ sin(\bar{y}) \tag{5.42}$$

now, let us define a desired region (closed-loop reference model) in time with the following interval function

$$M(t, \theta_q) = \theta_{q1}(1 - \theta_{q2} exp(-\theta_{q3}t)) \tag{5.43}$$

we desire that the output of the feedback system is within reference model $M(t, \theta_q)$. Where $\boldsymbol{\theta}_{q1}$ is the interval of the hard and soft speficifications, $\boldsymbol{\theta}_{q2}$ and $\boldsymbol{\theta}_{q3}$ are intervals to fix the response speed in time of the function $M(t, \theta_q)$, t is a time interval. If we define the hard $\boldsymbol{\theta}_{q1} = [1.1, 1.2]$ and soft $\boldsymbol{\theta}_{q1} = [1, 1.3]$ specification and $\boldsymbol{\theta}_{q2} = [0.5, 1]$, $\boldsymbol{\theta}_{q3} = [0.2, 1]$ and $t := \{t \in R | 0 \leq t \leq 40\}$ the bound interval of the hard and soft interval function $M(t, \theta_q)$ is depicted in Figure 5.28.

In Figure (5.29) we make a geometric representation of the interval optimization approach to feedback dynamic system. We applied a nominal feedforward $u^* = 0.342$ in the state feedback law (5.41) and we found a set of boxes $\vartheta_c = \{[\underline{k}_1, \overline{k}_1], [\underline{k}_2, \overline{k}_2]\}$ of the feedback controller such that the trajectory of the state variable $x_1 = c_{x1} = \mu_c l(t, \theta_p, \vartheta_c)$ is within the limits of the reference model $\gamma_y = M(t, \theta_q)$.

A more formal expression can be expressed as follows:

$$\begin{aligned}
\Sigma_{\forall\exists} = \ & \{\theta_c]|\forall(t \in \boldsymbol{t}')\forall(\vartheta_{p1} \in \boldsymbol{\vartheta}'_{p1})\forall(\vartheta_{p2} \in \boldsymbol{\vartheta}'_{p2})\exists(\theta_q \in \boldsymbol{\theta}'_q) \\
& (\mu_{cl}(t, \vartheta_{p1}, \overline{\vartheta}_{p2}, \theta_c) \subseteq M(t, \vartheta_q))\} \\
& \Downarrow \\
\Sigma_{\forall\exists} = \ & \{k_1 \times k_2 | \forall(t \in \boldsymbol{t}')\forall(b \in \boldsymbol{b}')\forall(M \in \boldsymbol{M}')\forall(R \in \boldsymbol{R}')\forall(g \in \boldsymbol{g}') \\
& \exists(\theta_{q1} \in \boldsymbol{\theta}'_{q1})\exists(\theta_{q2} \in \boldsymbol{\theta}'_{q2})\exists(\theta_{q3} \in \boldsymbol{\theta}'_{q3}) \\
& (x_1(M, R, g, b, k_1, k_2, t) \subseteq M(\theta_{q1}, \theta_{q2}, \theta_{q3}, t))\}
\end{aligned} \tag{5.44}$$

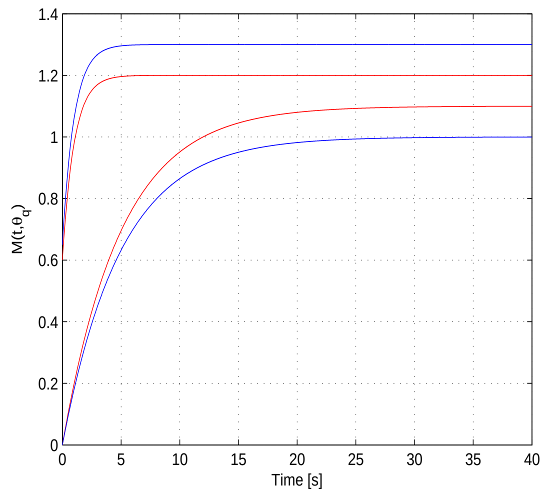


Fig. 5.28. Hard and soft specification.

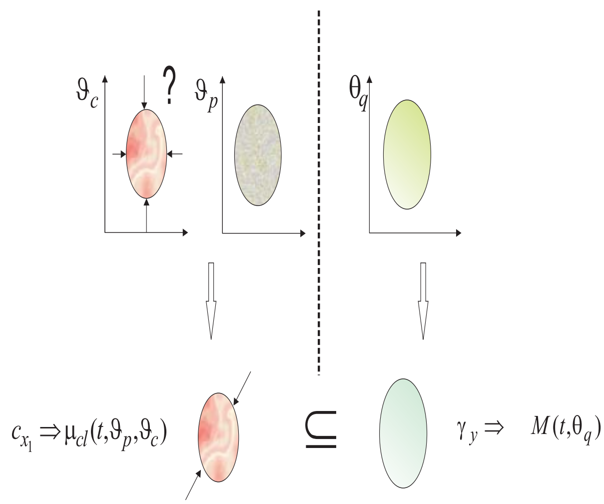


Fig. 5.29. Representation of the interval optimization approach of the feedback dynamic system.

where the parameters $\theta_q = \{\boldsymbol{\theta}_{q1}, \boldsymbol{\theta}_{q2}, \boldsymbol{\theta}_{q3}\}$ are existentially quantified $\exists$, because we want to ensure that x_1 is within $M(t, \theta_q)$.

In Table 5.11 the values of the parameters are indicated.

Table 5.11. Parameters

Parameters	Values
R	0.25
M	0.15
g	10
$\boldsymbol{b}$	$[0.001, 0.0259]$
τ	0.342
$\boldsymbol{\theta}_{q1}(soft)$	$[1, 1.3]$
$\boldsymbol{\theta}_{q1}(hard)$	$[1.1, 1.2]$
$\boldsymbol{\theta}_{q2}$	$[0.5, 1]$
$\boldsymbol{\theta}_{q3}$	$[0.2, 1]$
$\boldsymbol{k}_1$	$[0.01, 5]$
$\boldsymbol{k}_2$	$[0.01, 5]$

Set of hard and soft parameters k_1, k_2 of the state feedback controller are depicted in Figure 5.30.

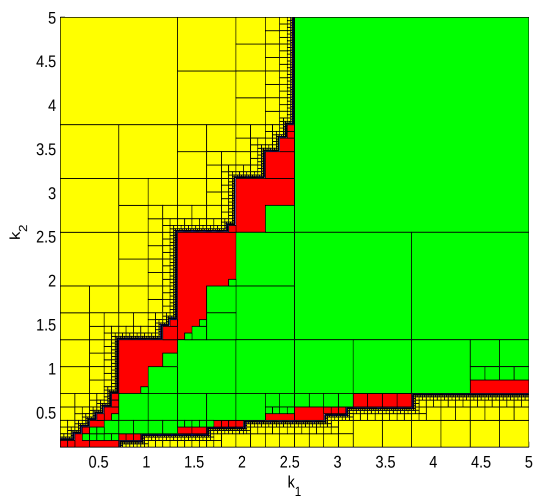


Fig. 5.30. Set of hard and soft parameters k_1, k_2 of the state feedback controller determined with $\tau = 0.342$, $\boldsymbol{\theta}_{q1}(hard) = [1.1, 1.2]$ and $\boldsymbol{\theta}_{q1}(soft) = [1, 1.3]$. The green and red boxes represent the hard and soft solution set, respectively. The yellow boxes are outside of the hard and soft solution set and the black boxes are undefined.

We will perform some robustness tests. From the paving indicated in Figure 5.30, three parameters from the feedback controller will be selected $\vartheta_{c1} = \{k_1 = 3.7, k_2 = 4\}$, $\vartheta_{c2} = \{k_1 = 3.7, k_2 = 1.5\}$ and $\vartheta_{c3} = \{k_1 = 1.5, k_2 = 3\}$. ϑ_{c1} and ϑ_{c2} are within the paving and ϑ_{c3} is out. With a nominal feedforward $u^* = 0.342$, system will be controlled in an interval of time $\{0 \leq t \leq 40\}$. From $t = 0.5$ to $t = 5$, the parameter b will be variated within the interval $b = [0.001, 0.0259]$. These values belong to maximum attainable uncertainty for the soft specification as indicated in Table 5.8. In Figures 5.31 and 5.32, we can see that with parameters $\vartheta_{c1} = \{k_1 = 3.7, k_2 = 4\}$ and $\vartheta_{c2} = \{k_1 = 3.7, k_2 = 1.5\}$, the feedback controller maintaints the output within $M(t, \theta_q)$ for all $0 \leq t \leq 40$, obtaining a good robust performance under parametric uncertainty. In Figure 5.20, we can see that with parameters $\vartheta_{c3} = \{k_1 = 1.5, k_2 = 3\}$, the feedback controller does not meet the specifications in certain instant of time.

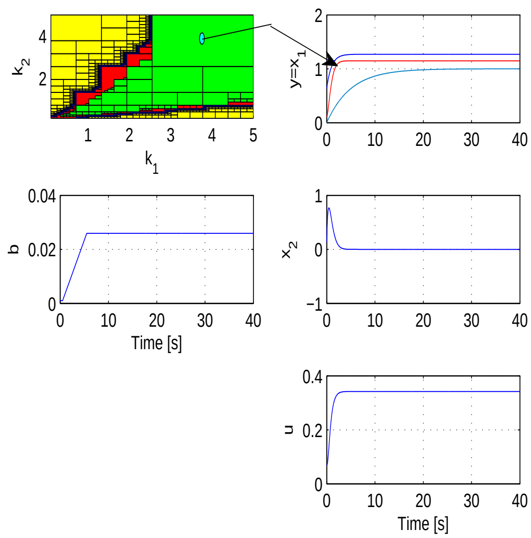


Fig. 5.31. Robustness test of the state feedback controller with $\{k_1 = 3.7, k_2 = 4\}$, nominal feedforward $\tau^* = 0.342$ and soft uncertainty $\boldsymbol{b} = [0.01, 0259]$.

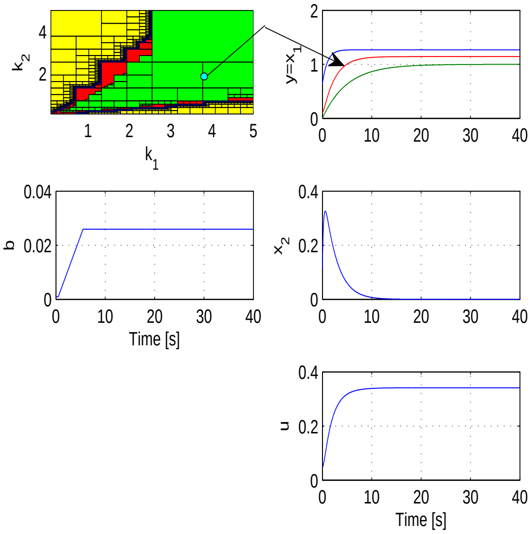


Fig. 5.32. Robustness test of the state feedback controller with $\{k_1 = 3.7, k_2 = 1.5\}$, nominal feedforward $\tau^* = 0.342$ and soft uncertainty $\boldsymbol{b} = [0.01, 0259]$.

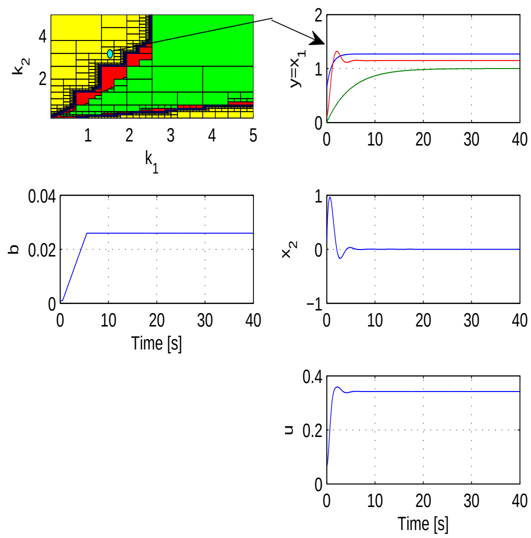


Fig. 5.33. Robustness test of the state feedback controller with $\{k_1 = 1.5, k_2 = 3\}$, nominal feedforward $\tau^* = 0.342$ and soft uncertainty $\boldsymbol{b} = [0.01, 0259]$.

5.3 Applications to fed-batch bioreactors

5.3.1 Dynamic model

A bioreactor is a tank in which several microbial growth reactions and enzyme-catalyzed reactions occur simultaneously in a liquid environment. These nonlinear systems show both unstructured and parametric uncertainty. The form appears mainly due to the simplifications in the model regarding to the behavior of the microorganism and the reactor itself. Some properties of the model can be consulted in thesis **Picó-Marco (2004)**

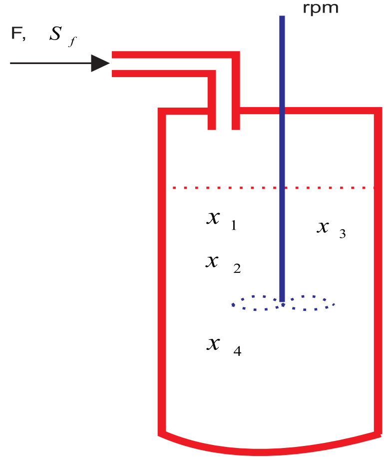


Fig. 5.34. Fed-batch bioreactor.

- The cell is considered as a "black box" and only the main extracellular species are consumed or excreted in the environment, without delve into the intracellular mechanisms.
- The cells may be subject to a phenomenon of "aging" so not all have the same capacity of division, or genetic mutation so that some cells do not produce species of interest. Nevertheless the model is built taking into account an average cell.
- The conditions and concentrations in the tank are supposed to be homogeneous, which is a good approximation for lab-scale fermentators.
- The biomass actually needs several substrates to grow, but all are found in excess, both in the environment and the inflow, except one. This limited substrate alone will take part in the equations, the other substrates will be considered.

It is difficult to identify the value of the model parameters and it is interesting to know the amount of parametric uncertainty that it could have. If the basic considered reaction is

$$X(biomass) + S(substrate) \longmapsto X + P(product) \tag{5.45}$$

then

$$\begin{aligned}
\dot{x}_1 &= \mu(x_2)x_1 - \lambda x_1^2 \\
\dot{x}_2 &= (\lambda(S_f - x_2) - \mu(x_2)/y_{x/s})x_1 \\
\dot{x}_3 &= \lambda x_1 x_3 \\
\dot{x}_4 &= r x_1/y_{x/p} - \lambda x_1 x_4
\end{aligned} \tag{5.46}$$

where

- x_1, x_2, x_4 represent the concentrations of biomass, substrate and product in the tank.
- x_3 is the volume in the bioreactor, $\lambda = D/x_1$ the controller and $D = F/x3$ the dilution.
- F is the input flux.
- S_f the substrate concentration in the input flux.
- $y_{x/s}$ and $y_{x/p}$ are yield coefficients.
- $\mu(x_2)$ is the microorganisms specific growth rate, that is, per unit of biomass.
- r the specific production rate. Generally, it depends on several factors, although in certain cases it is proportional to $\mu(x_2)$ and then the product formation is said to be associated to the growth.

The specific growth rate also depends on several factors such as the concentrations of substrate and product, the PH, temperature, etc. Usually, it is expressed as:

$$\mu = \mu(x_2)\mu(x_4)\mu(pH)\mu(T)... \tag{5.47}$$

Temperature, pH and other environmental variables are often kept constant. The more common expressions for the microorganisms specific growth rate and product concentration are:

- Substrate concentration.

1. Monod (Or Michaelis-Menten)

$$\mu(x_2) = \frac{\mu_m x_2}{k_m + x_2} \tag{5.48}$$

where μ_m is the maximum growth rate, k_m a transport constant.

2. Haldane, in which the inhibition of growth by the substrate is taken into account.

$$\mu(x_2) = \frac{\mu_o x_2}{k_m + x_2 + \frac{x_2^2}{k_i}} \tag{5.49}$$

In practice there is always an inhibition of biomass growth, but in many cases it appears for substrate concentrations that are very high when they are compared to those in the zone of interest. Hence, often a Monod is used. The opposite case, in which it is necessary to use a Haldane, appears in applications such as water decontamination.

- Product concentration. A common example may be

$$\mu(x_4) = \frac{k_p}{k_p + x_4} \tag{5.50}$$

The model obtained under these assumptions is:

$$\begin{aligned} \dot{x}_1 &= \mu(x_2)x_1 - \lambda x_1^2 \\ \dot{x}_2 &= (\lambda(S_f - x_2) - \mu(x_2)/y_{x/s})x_1 \\ \dot{x}_3 &= \lambda x_1 x_3 \\ \mu(x_2) &= \mu_m x_2/(k_m + x_2) \end{aligned} \tag{5.51}$$

the same model can be consulted in **Agrawal et al. (1989)**. In order to simplify its practical implementation, we made some considerations. The control variable λ may be chosen as $\lambda = \lambda_{min} + \lambda_{max}$ and substrate concentration in input flux $S_f \in [S_{fmin}, S_{fmax}]$, then the equation (5.51) of the system model may be represented by

$$\begin{aligned} \dot{x}_1 &= \mu(x_2)x_1 - (\lambda_{max} + \lambda_{min})x_1^2 \\ \dot{x}_2 &= (\lambda_{max}(S_f - x_2) + \lambda_{min}(S_f - x_2) - \mu(x_2)/y_{x/s})x_1 \\ \dot{x}_3 &= (\lambda_{max} + \lambda_{min})x_1 x_3 \\ \mu(x_2) &= \mu_m x_2/(k_m + x_2) \end{aligned} \tag{5.52}$$

5.3.2 Simulation of the system considering a region of flat outputs

If λ, S_f are taken as inputs and x_1, x_3 as flat outputs, then the output variables are $y_1 = x_1$(biomass concentration) and $y_2 = x_3$ (volume). From the third equation in (5.51) we can reconstruct (λ).

$$\lambda = \frac{\dot{x}_3}{x_T} = \frac{\dot{x}_3}{x_1 x_3} \tag{5.53}$$

From the first equation in (5.51) we can reconstruct the microorganisms specific growth rate $\mu(x_2)$

$$\mu(x_2) = \frac{\dot{x}_1}{x_1} + \frac{\dot{x}_3}{x_3} \tag{5.54}$$

using the expression of $\mu(x_2)$ and replacing it in the equation $\mu(x_2) = \frac{\mu_m x_2}{k_m + x_2}$ we can reconstruct the state variable x_2 of this equation

$$\begin{aligned} x_2 &= \frac{k_m(x_3\dot{x}_1+\dot{x}_3x_1)}{\mu_m x_1x_3-(x_3\dot{x}_1+\dot{x}_3x_1)} \\ &= \frac{k_m(\dot{x}_T)}{\mu_m x_T-(\dot{x}_T)} \end{aligned} \tag{5.55}$$

being $x_T = x_1x_3$ the absolute quantity of biomass. Finally, using the second equation of (5.51) the expression of substrate concentration in the input flux S_f is

$$S_f = \frac{k_m\mu(x_2)}{\mu_m-\mu(x_2)} + (\mu(x_2)/y_{x/s})\frac{x_1x_3}{\dot{x_3}} \tag{5.56}$$

Since it maintains a constant microorganisms growth rate $\mu(x_2) = const$ corresponds with an exponential trajectory for the absolute quantity of biomass $x_T = x_1x_3$, then the equation $\mu(x_2) = \frac{\mu_m x_2}{k_m+x_2}$ is transformed into $\mu_r = \frac{\mu_m x_{2r}}{k_m+x_{2r}}$ in steady state operation mode. Thus the first and third equation of (5.51) in steady stable correspond to

$$\begin{aligned} \dot{x}_{1r} &= \mu_r x_{1r} - \lambda_n x_{1r}^2 \\ \dot{x}_{3r} &= \lambda_n x_{1r}x_{3r} \end{aligned} \tag{5.57}$$

if we solve these equations, we obtain the expressions of the flat outputs in steady state. Thus the flat output $y_1 = x_1$ is

$$y_1 = x_1 = \frac{\mu_r/\lambda_n}{(\frac{\mu_r}{\lambda_n x_{10}}-1)e^{-\mu_r t}+1} \tag{5.58}$$

and the flat output $y_2 = x_3$ is

$$y_2 = x_3 = x_{30}(1-\frac{\lambda_n x_{10}}{\mu_r}) + x_{10}x_{30}\frac{\lambda_n}{\mu_r}e^{\mu_r t} \tag{5.59}$$

Being μ_r the nominal microorganisms growth rate, λ_n a nominal controller, x_{10} the uncertain initial condition in biomass and x_{30} the uncertain initial condition in volume. With the flat output equations (5.58) and (5.59) we can obtain the equilibrium states of the system in terms of the equilibrium values of flat outputs. Thus, from equation $\mu_r = \frac{\mu_m x_{2r}}{k_m+x_{2r}}$ we can obtain x_{2r}

$$x_{2r} = \frac{k_m\mu_r}{\mu_m-\mu_r} \tag{5.60}$$

the equation (5.56) is transformed in stable steady to

$$S_{fr} = x_{2r} + (\mu_r/y_{x/s})\frac{1}{\lambda_n} \tag{5.61}$$

therefore the nominal controller λ_n is

$$\lambda_n = \frac{\mu_r / y_{x/s}}{S_{fr} - x_{2r}} \tag{5.62}$$

Let us make some notations. $\overline{\vartheta}_k = \{\lambda_n\}$ is a nominal controller, $\bar{\vartheta}_o = \{x_{10}, x_{30}\}$ are fixed parameters for the flat outputs. $\vartheta_s = \{[\underline{\mu}_r, \overline{\mu}_r], \overline{\vartheta}_k, \bar{\vartheta}_o\}$ contains the specification parameters, $\bar{\vartheta}_s = \{\mu_r = (\bar{\mu}_r - \underline{\mu}_r)/2, \lambda_n, x_{10}, x_{30}\}$ is a point of specification. $\vartheta_p = \{[\underline{k}_m, \bar{k}_m], [\underline{\mu}_m, \overline{\mu}_m], [\underline{y}_{x/s}, \bar{y}_{x/s}]\}$ are uncertain intervals of the plant. $\bar{\vartheta}_p = \{k_m, \mu_m, y_{x/s}\}$ is a nominal plant. Let us define two nominal hard $\boldsymbol{\mu}_r = [0.09, 0.13]$ and soft $\boldsymbol{\mu}_r = [0.07, 0.15]$ specifications as is seen in Figure 5.35.

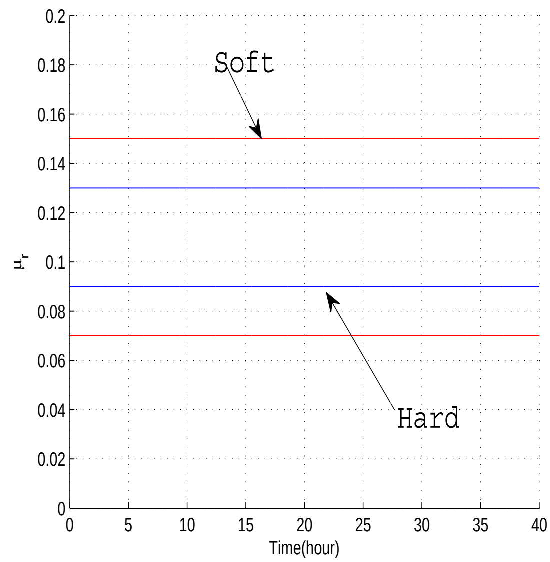


Fig. 5.35. Hard $\boldsymbol{\mu}_r = [0.09, 0.13]$ and soft $\boldsymbol{\mu}_r = [0.07, 0.15]$ specifications.

With these specifications and a nominal plant $k_m = 0.15$, $\mu_m = 0.1$ and $y_{x/s} = 0.6993$, let us compute the bounding intervals x_{2r}, S_{fr} and λ_n from equations (5.60), (5.61) and (5.62) respectively.

$$\begin{aligned}
\mu_r &= \left[\underline{\mu}_r, \overline{\mu}_r\right] \Rightarrow E_{s\mu}(t, \vartheta_s) \Rightarrow E_{s\mu}(t) \\
x_{2r} &= \frac{\left[\underline{k}_m, \bar{k}_m\right]\left[\underline{\mu}_r, \overline{\mu}_r\right]}{\left[\underline{\mu}_m, \overline{\mu}_m\right] - \left[\underline{\mu}_r, \overline{\mu}_r\right]} \Rightarrow E_{sx2}(t, \vartheta_s, \bar{\vartheta}_p) \Rightarrow E_{sx2}(t) \\
S_{fr} &= [\underline{x}_{2r}, \overline{x}_{2r}] + \frac{\left[\underline{\mu}_r, \overline{\mu}_r\right] / \left[\underline{y}_{x/s}, \overline{y}_{x/s}\right]}{\lambda_n} \Rightarrow E_{sf}(t, \vartheta_s, \bar{\vartheta}_p) \Rightarrow E_{sf}(t) \\
\lambda_n &= \frac{\left[\underline{\mu}_r, \overline{\mu}_r\right] / \left[\underline{y}_{x/s}, \overline{y}_{x/s}\right]}{\left[\underline{S}_{fr}, \overline{S}_{fr}\right] - \left[\underline{x}_{2r}, \overline{x}_{2r}\right]} \Rightarrow E_{s\lambda}(t, \vartheta_s, \bar{\vartheta}_p) \Rightarrow E_{s\lambda}(t)
\end{aligned} \tag{5.63}$$

We make the clarification that $E_{s\mu}$, E_{sx2}, E_{sf} and $E_{s\lambda}$ are related to nominal specifications that was explained in Section (4.4) from Chapter four. After realizing the interval operations from equation (5.63) we obtain intervals indicated in Figure 5.36.

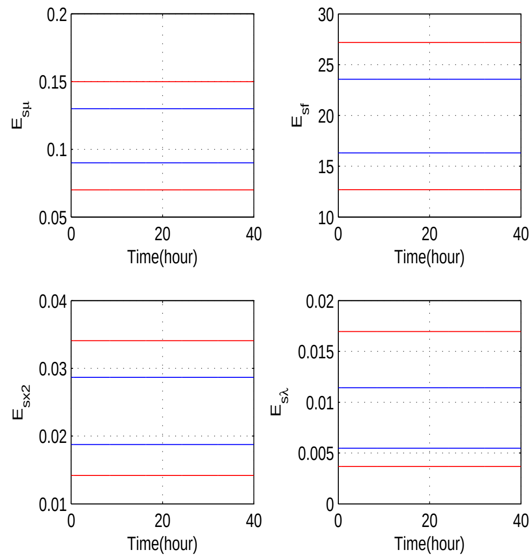


Fig. 5.36. Nominal specifications $E_{s\mu}$, E_{sx_2}, E_{sf} and $E_s\lambda$ computed with hard $\boldsymbol{\mu}_r = [0.09, 0.13]$ and soft $\boldsymbol{\mu}_r = [0.07, 0.15]$ specifications and nominal plant $k_m = 0.15$, $\mu_m = 0.1$ and $y_{x/s} = 0.6993$.

5.3.3 Robustness analysis of a nominal controller

In the following example, we will use the set of nominal specifications obtained in the previous section. The approach for obtaining the maximum permissible uncertainty of the plant by a nominal controller $\overline{\vartheta}_k = \{\lambda_n\}$ is similar to the developed in previous applications. Let us maximize the plant space such that to evaluate the output, the state and controller interval functions are within the set of nominal specifications. The Approach is ilustrated in Figure 5.37.

The interval functions $c_{\mu(x_2)}$, φ_{x_2}, ψ_{sf} and ψ_λ are computed from equations (5.51), (5.55), (5.56) and (5.53) respectively. Thus, we obtain the following expressions

$$\mu(x_2) = \mu_m x_2/(k_m + x_2) \Rightarrow c_{\mu(x_2)}(t, \overline{\vartheta}_s, \vartheta_p) \tag{5.64}$$

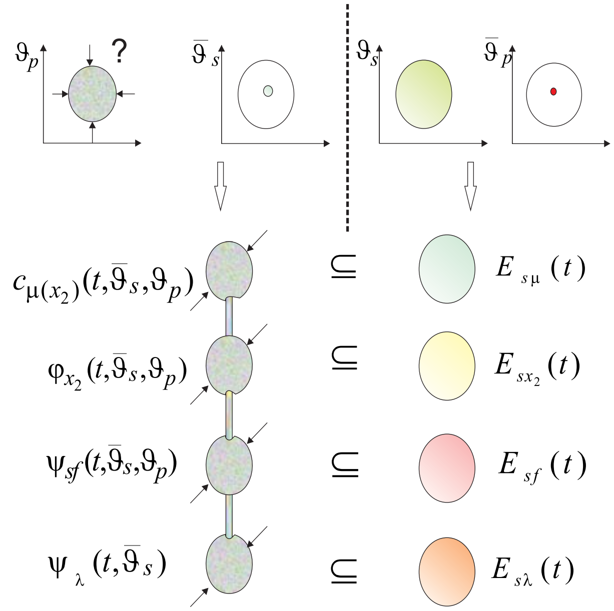


Fig. 5.37. Optimization approach to maximize the plant space ϑ_p.

$$\begin{aligned} x_2 &= \frac{k_m(x_3\dot{x}_1+\dot{x}_3x_1)}{\mu_m x_1x_3-(x_3\dot{x}_1+\dot{x}_3x_1)} \\ &= \frac{k_m(\dot{x}_T)}{\mu_m x_T-(\dot{x}_T)} \Rightarrow \varphi_{x_2}(t,\overline{\vartheta}_s,\vartheta_p) \end{aligned} \tag{5.65}$$

$$S_f = \frac{k_m\mu(x_2)}{\mu_m-\mu(x_2)} + (\mu(x_2)/y_{x/s})\frac{x_1x_3}{\dot{x_3}} \Rightarrow \psi_{sf}(t,\overline{\vartheta}_s,\vartheta_p) \tag{5.66}$$

$$\lambda = \frac{\dot{x}_3}{x_T} = \frac{\dot{x}_3}{x_1x_3} \Rightarrow \psi_\lambda(t,\overline{\vartheta}_s) \tag{5.67}$$

Instead comparing $E_{s\mu}$ with $\mu(x_2) = \frac{\dot{x}_1}{x_1} + \frac{\dot{x}_3}{x_3} \Rightarrow \mu(x_2)(t,\bar{\vartheta}_s)$ from equation (5.54), we compared $E_{s\mu}$ with $\mu(x_2) = \frac{\mu_m x_2}{k_m+x_2} \Rightarrow c_{\mu x_2}(t,\bar{\vartheta}_s,\vartheta_p)$ with the finality to determine maximum permissible uncertainty in the plant and computed the attainable nominal specification $E_{s\mu}$. In previous equations, a set of plants $\mu_m \times k_m \times y_{x/s}$ will be evaluated. Solution sets will be those plants that fulfill the nominal specifications.

The set of constraints is expressed of the following form:

$$C(t, \vartheta_s, \vartheta_p, \vartheta_k, \vartheta_o) = \{c_{\mu(x_2)}(t, \lambda_n, \mu_r, x_{10}, x_{30}, \mu_m, k_m) \subseteq E_{s\mu}(t) \wedge \\ \varphi_{x2}(t, \lambda_n, \mu_r, x_{10}, x_{30}, k_m, \mu_m) \subseteq E_{sx2}(t) \wedge \\ \psi_\lambda(t, \lambda_n, \mu_r, x_{10}, x_{30}) \subseteq E_{s\lambda}(t) \wedge \\ \psi_{sf}(t, \lambda_n, \mu_r, x_{10}, x_{30}, k_m, \mu_m, y_{x/s}) \subseteq E_{sf}(t)\} \tag{5.68}$$

and the solution set $\Sigma_{\forall\exists}$ can be expressed as:

$$\Sigma_{\forall\exists} = \{\mu_m \times k_m \times y_{x/s} | \forall(x_{10} \in \boldsymbol{x}'_{10}) \\ \forall(x_{30} \in \boldsymbol{x}'_{30}) \forall(t \in \boldsymbol{t}') \exists(\lambda_n \in \boldsymbol{\lambda}'_n) \exists(\mu_r \in \boldsymbol{\mu}'_r) \\ (c_{\mu(x_2)}(t, \lambda_n, \mu_r, x_{10}, x_{30}, \mu_m, k_m) \subseteq E_{s\mu}(t) \wedge \\ \varphi_{x_2}(t, \lambda_n, \mu_r, x_{10}, x_{30}, k_m, \mu_m) \subseteq E_{sx2}(t) \wedge \\ \psi_\lambda(t, \lambda_n, \mu_r, x_{10}, x_{30}) \subseteq E_{s\lambda}(t) \wedge \\ \psi_{sf}(t, \lambda_n, \mu_r, x_{10}, x_{30}, k_m, \mu_m, y_{x/s}) \subseteq E_{sf}(t))\} \tag{5.69}$$

Values and assigned quantifiers to the parameters are indicated in Table 5.12.

Table 5.12. Assigned quantifiers to the parameters to meet hard $\exists(\mu_r \in [0.09, 0.13])$ and soft $\exists(\mu_r \in [0.07, 0.15])$ specifications

Quantifiers ($\forall$)	Quantifiers ($\exists$)
$\forall(x_{10} \in [0.7, 0.7])$	$\exists(\mu_r \in [0.09, 0.13])$
$\forall(x_{30} \in [1, 1])$	$\exists(\mu_r \in [0.07, 0.15])$
$\forall(k_m \in [0.1, 0.2])$	$\exists(\lambda_n \in [0.0079, 0.0079])$
$\forall(\mu_m \in [0.5, 1.2])$	
$\forall(y_{x/s} \in [0.4, 1.2])$	
$\forall(t \in [0, 40])$	

Intervals of hard and soft maximum permissible uncertainty by a nominal controller $\lambda_n = 0.0079$ are indicated in Table 5.13 .

Table 5.13. Maximum permissible uncertainty by a nominal controller $\lambda_n = 0.0079$ to meet hard $\exists(\mu_r \in [0.09, 0.13])$ and soft $\exists(\mu_r \in [0.07, 0.15])$ specifications

Variables	$\exists(\mu_r \in [0.09, 0.13])$	$\exists(\mu_r \in [0.07, 0.15])$
μ_m	$[0.783, 0.837]$	$[0.755, 0.865]$
k_m	$[0.1419, 0.1581]$	$[0.1333, 0.1665]$
$y_{x/s}$	$[0.6723, 0.7263]$	$[0.6443, 0.7543]$

5.3.4 Computation of a family of controllers.

Let us compute the family of hard and soft controllers that could meet some hard and soft specifications. The approach consists in maximize the space of controllers ϑ_k such that some specifications are met and the constraints are satisfied. A representation of the approach is depicted in Figure 5.38.

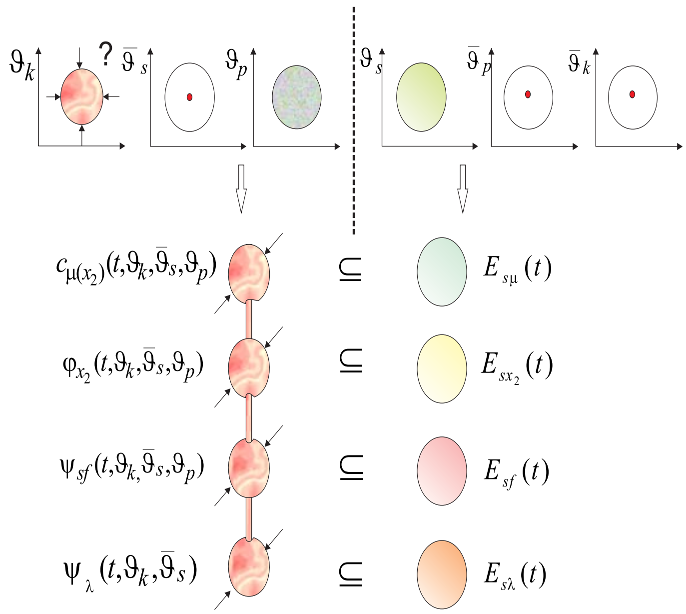


Fig. 5.38. Optimization approach to maximize the space of controllers ϑ_k.

Let us make some notations. $\vartheta_k = \{[\underline{\lambda}_n, \overline{\lambda}_n]\}$ is a family of nominal controllers, $\overline{\vartheta}_k = \{\lambda_n\}$ is a fixed nominal controller, $\bar{\vartheta}_o = \{x_{10}, x_{30}\}$ are fixed parameters for the flat outputs. $\vartheta_s = \{[\underline{\mu}_r, \overline{\mu}_r], \bar{\vartheta}_o\}$ contains the specification parameters, $\bar{\vartheta}_s = \{\mu_r = (\bar{\mu}_r - \underline{\mu}_r)/2, x_{10}, x_{30}\}$ is a point of specification. $\vartheta_p = \{[\underline{k}_m, \bar{k}_m], [\underline{\mu}_m, \overline{\mu}_m], [\underline{y}_{x/s}, \bar{y}_{x/s}]\}$ are uncertain intervals of the plant. $\bar{\vartheta}_p = \{k_m, \mu_m, y_{x/s}\}$ is a nominal plant. As we can see we separed the controller ϑ_k from specifications ϑ_s since it is the space that we want to maximize.

The set of nominal $E_{s\mu}$, E_{sx2}, E_{sf} and $E_{s\lambda}$ specifications are obtained from equation (5.63) but including $\overline{\vartheta}_k$ in some expressions, they are as follows:

$$\begin{aligned}
\mu_r &= \left[\underline{\mu}_r, \overline{\mu}_r\right] \Rightarrow E_{s\mu}(t, \vartheta_s) \Rightarrow E_{s\mu}(t) \\
x_{2r} &= \frac{[\underline{k}_m, \overline{k}_m][\underline{\mu}_r, \overline{\mu}_r]}{[\underline{\mu}_m, \overline{\mu}_m] - [\underline{\mu}_r, \overline{\mu}_r]} \Rightarrow E_{sx2}(t, \vartheta_s, \overline{\vartheta}_p) \Rightarrow E_{sx2}(t) \\
S_{fr} &= [\underline{x}_{2r}, \overline{x}_{2r}] + \frac{[\underline{\mu}_r, \overline{\mu}_r]/\left[\underline{y}_{x/s}, \overline{y}_{x/s}\right]}{\lambda_n} \Rightarrow E_{sf}(t, \vartheta_s, \bar{\vartheta}_p, \overline{\vartheta}_k) \Rightarrow E_{sf}(t) \\
\lambda_n &= \frac{[\underline{\mu}_r, \overline{\mu}_r]/\left[\underline{y}_{x/s}, \overline{y}_{x/s}\right]}{[\underline{S}_{fr}, \overline{S}_{fr}] - [\underline{x}_{2r}, \overline{x}_{2r}]} \Rightarrow E_{s\lambda}(t, \vartheta_s, \bar{\vartheta}_p, \overline{\vartheta}_k) \Rightarrow E_{s\lambda}(t)
\end{aligned} \tag{5.70}$$

on similar way $c_{\mu(x2)}$, φ_{x2}, ψ_{sf} and ψ_λ can be obtained from equations (5.64), (5.65), (5.66) and (5.67) including $\overline{\vartheta}_k$ as follows:

$$\mu(x_2) = \mu_m x_2/(k_m + x_2) \Rightarrow c_{\mu(x_2)}(t, \overline{\vartheta}_s, \vartheta_p, \vartheta_k) \tag{5.71}$$

$$\begin{aligned} x_2 &= \tfrac{k_m(x_3\dot{x}_1+\dot{x}_3x_1)}{\mu_m x_1 x_3-(x_3\dot{x}_1+\dot{x}_3x_1)} \\ &= \tfrac{k_m(\dot{x}_T)}{\mu_m x_T-(\dot{x}_T)} \Rightarrow \varphi_{x_2}(t, \overline{\vartheta}_s, \vartheta_p, \vartheta_k) \end{aligned} \tag{5.72}$$

$$S_f = \tfrac{k_m\mu(x_2)}{\mu_m-\mu(x_2)} + (\mu(x_2)/y_{x/s})\tfrac{x_1x_3}{\dot{x}_3} \Rightarrow \psi_{sf}(t, \overline{\vartheta}_s, \vartheta_p, \vartheta_k) \tag{5.73}$$

$$\lambda = \tfrac{\dot{x}_3}{x_T} = \tfrac{\dot{x}_3}{x_1x_3} \Rightarrow \psi_\lambda(t, \overline{\vartheta}_s, \vartheta_k) \tag{5.74}$$

The set of constraints is expressed of the following form:

$$\begin{aligned} &C(t, \vartheta_s, \vartheta_p, \vartheta_k, \vartheta_o) = \{c_{\mu(x_2)}(t, \lambda_n, \mu_r, x_{10}, x_{30}, \mu_m, k_m) \subseteq E_{s\mu}(t)\wedge \\ &\varphi_{x2}(t, \lambda_n, \mu_r, x_{10}, x_{30}, k_m, \mu_m) \subseteq E_{sx2}(t)\wedge \\ &\psi_\lambda(t, \lambda_n, \mu_r, x_{10}, x_{30}) \subseteq E_{s\lambda}(t)\wedge \\ &\psi_{sf}(t, \lambda_n, \mu_r, x_{10}, x_{30}, k_m, \mu_m, y_{x/s}) \subseteq E_{sf}(t)\} \end{aligned} \tag{5.75}$$

The solution set $\Sigma_{\forall\exists}$ can be expressed as:

$$\begin{aligned} &\Sigma_{\forall\exists} = \{\lambda_n | \forall(x_{10} \in \boldsymbol{x}'_{10})\forall(x_{30} \in \boldsymbol{x}'_{30})\forall(t \in \boldsymbol{t}') \\ &\forall(\mu_m \in \boldsymbol{\mu}'_m)\forall(k_m \in \boldsymbol{k}'_m)\forall(y_{x/s} \in \boldsymbol{y}'_{x/s})\exists(\mu_r \in \boldsymbol{\mu}'_r) \\ &(c_{\mu(x2)}(t, \lambda_n, \mu_r, x_{10}, x_{30}, \mu_m, k_m) \subseteq E_{s\mu}(t)\wedge \\ &\varphi_{x2}(t, \lambda_n, \mu_r, x_{10}, x_{30}, k_m, \mu_m) \subseteq E_{sx2}(t)\wedge \\ &\psi_\lambda(t, \lambda_n, \mu_r, x_{10}, x_{30}) \subseteq E_{s\lambda}(t)\wedge \\ &\psi_{sf}(t, \lambda_n, \mu_r, x_{10}, x_{30}, k_m, \mu_m, y_{x/s}) \subseteq E_{sf}(t))\} \end{aligned} \tag{5.76}$$

The assigned quantifiers to the parameters to obtain the familiy of hard and soft controllers $\vartheta_k(\forall)$ that could ensure that some hard and soft specifications are met $\vartheta_s(\exists)$ and the constraints are satisfied under parametric uncertainty in the plant $\vartheta_p(\forall)$ are indicated in Table 5.14. The family of hard and soft controllers obtained are indicated in Table 5.15.

5.3.5 Computation of attainable specifications by a nominal controller

Now, let us solve the space of attainable specifications by a nominal controller. The optimization approach indicated in Figure 5.39 consists on optimizing the space of specifications ϑ_s under parametric uncertainty in the plant.

Table 5.14. Quantifiers assigned to the parameters to find a family of hard and soft controllers ϑ_k and met some hard $\exists(\mu_r \in [0.09, 0.13])$ and soft $\exists(\mu_r \in [0.07, 0.15]$ specifications

Quatifiers ($\forall$)	Hard plants	Soft plants
$\forall(x_{10} \in [0.7, 0.7])$		
$\forall(x_{30} \in [1, 1])$	$\forall(k_m \in [0.1419, 0.1581])$	$\forall(k_m \in [0.1333, 0.1665])$
$\forall(\lambda_n \in [0.004, 0.01])$	$\forall(\mu_m \in [0.783, 0.837])$	$\forall(\mu_m \in [0.755, 0.865])$
$\forall(t \in [0, 40])$	$\forall(y_{x/s} \in [0.6723, 0.7263])$	$\forall(y_{x/s} \in [0.6443, 0.7543])$

Table 5.15. Family of hard and soft controllers that ensure that some hard $\boldsymbol{\mu}_r = [0.09, 0.13]$ and soft $\boldsymbol{\mu}_r = [0.07, 0.15]$ specifications are met

Parameter	Hard	Soft
λ_n^*	$[0.0077, 0.0080]$	$[0.0076, 0.0081]$

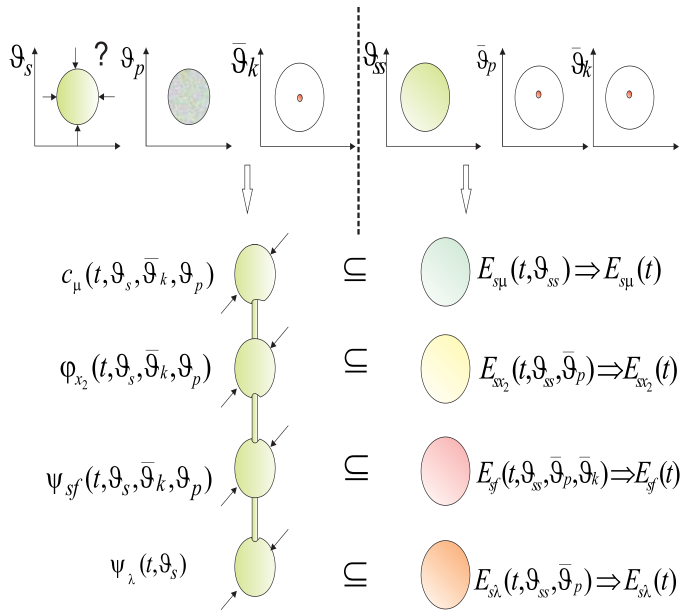


Fig. 5.39. Optimization approach to maximize the space of specifications ϑ_s.

The nominal specifications $E_{s\mu}$, E_{sx2}, E_{sf} and $E_{s\lambda}$ are computed from equation (5.70) as the previous case, the interval functions $c_{\mu(x2)}$, φ_{x2}, ψ_{sf} and ψ_λ are evaluated as follows:

$$\mu(x_2) = \mu_m x_2/(k_m + x_2) \Rightarrow c_{\mu(x_2)}(t, \vartheta_s, \vartheta_p, \bar{\vartheta}_k) \tag{5.77}$$

$$\begin{aligned} x_2 &= \frac{k_m(x_3\dot{x}_1+\dot{x}_3x_1)}{\mu_m x_1x_3-(x_3\dot{x}_1+\dot{x}_3x_1)} \\ &= \frac{k_m(\dot{x}_T)}{\mu_m x_T-(\dot{x}_T)} \Rightarrow \varphi_{x_2}(t,\vartheta_s,\vartheta_p,\bar{\vartheta}_k) \end{aligned} \tag{5.78}$$

$$S_f = \frac{k_m\mu(x_2)}{\mu_m-\mu(x_2)} + (\mu(x_2)/y_{x/s})\frac{x_1x_3}{\dot{x_3}} \Rightarrow \psi_{sf}(t,\vartheta_s,\vartheta_p,\bar{\vartheta}_k) \tag{5.79}$$

$$\lambda = \frac{\dot{x}_3}{x_T} = \frac{\dot{x}_3}{x_1x_3} \Rightarrow \psi_\lambda(t,\vartheta_s,\bar{\vartheta}_k) \tag{5.80}$$

the solution set $\Sigma_{\exists\forall}$ referred to the specifications space corresponds to:

$$\begin{aligned} &\Sigma_{\forall\exists} = \{\mu_r|\forall(x_{10}\in x_{10}^{'})\forall(x_{30}\in x_{30}^{'})\forall(t\in t^{'}) \\ &\cdot\forall(\mu_m\in\mu_m^{'})\forall(k_m\in k_m^{'})\forall(y_{x/s}\in y_{x/s}^{'})\exists(\lambda_n\in\lambda_n^{'}) \\ &(c_{\mu(x2)}(t,\mu_r,\lambda_n,x_{10},x_{30},\mu_m,k_m)\subseteq E_{s\mu}(t)\wedge \\ &\varphi_{x2}(t,\mu_r,\mu_m,k_m,x_{10},x_{30},\lambda_n)\subseteq E_{sx2}(t)\wedge \\ &\psi_\lambda(t,\mu_r,\lambda_n,x_{10},x_{30})\subseteq E_{s\lambda}(t)\wedge \\ &\psi_{sf}(t,\mu_r,\lambda_n,\mu_m,k_m,y_{x/s},x_{10},x_{30})\subseteq E_{sf}(t))\} \end{aligned} \tag{5.81}$$

Let us consider a nominal controller $\lambda_n = 0.0079$ and hard and soft parametric uncertainty from Table 5.13. The quantifiers and values used are indicated in Table 5.16. In Figure 5.40 we can see that the nominal controller $\lambda_n = 0.0079$ met the hard and soft specifications under hard and soft parametric uncertainty.

Table 5.16. Quantifiers to find attainable specifications by a nominal controller $\lambda_n = [0.0079, 0.0079]$

Quatifiers ($\forall$)	Hard plants	Soft plants
$\forall(x_{10}\in[0.7,0.7])$		
$\forall(x_{30}\in[1,1])$	$\forall(k_m\in[0.1419,0.1581])$	$\forall(k_m\in[0.1333,0.1665])$
$\forall(\mu_r\in[0.01,0.3])$	$\forall(\mu_m\in[0.783,0.837])$	$\forall(\mu_m\in[0.755,0.865])$
$\forall(t\in[0,40])$	$\forall(y_{x/s}\in[0.6723,0.7263])$	$\forall(y_{x/s}\in[0.6443,0.7543])$

It can be seen that to maintain a constant microorganisms growth rate $\mu(x_2) = const$ corresponds with an exponential trajectory of the absolute biomass quantity $x_T = x_1x_3$ and vice versa. With the family of obtained controllers it is possible to verify the fulfillment of specifications of the biomass and volume space as it is indicated in Figure 5.41

5.3.6 Robustness test

A robustness test in open-loop can be made by using the nonlinear system model (5.51), with hard controllers $\lambda = \{0.0077, 0.0078, 0.0079, 0.0080\}$, substrate concentrations $S_f = [18, 22]$ and parametric uncertainty in the plant $\mu_m = [0.783, 0.837], k_m = [0.1419, 0.1581], y_{x/s} = [0.6723, 0.7263]$.

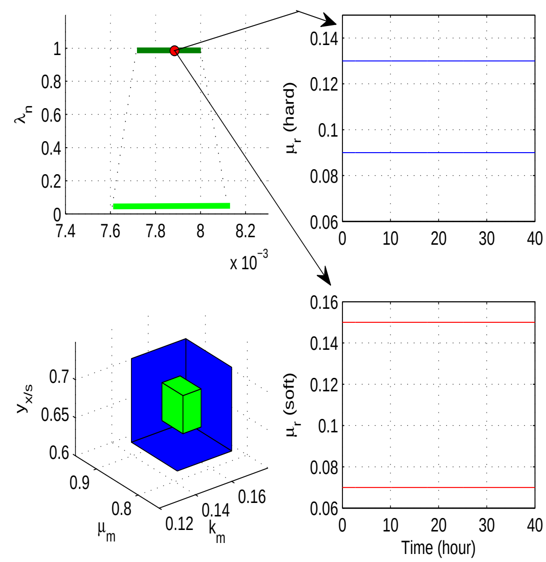


Fig. 5.40. Hard and soft attainable specifications by a nominal controller $\lambda_n = 0.0079$ under hard and soft parametric uncertainty.

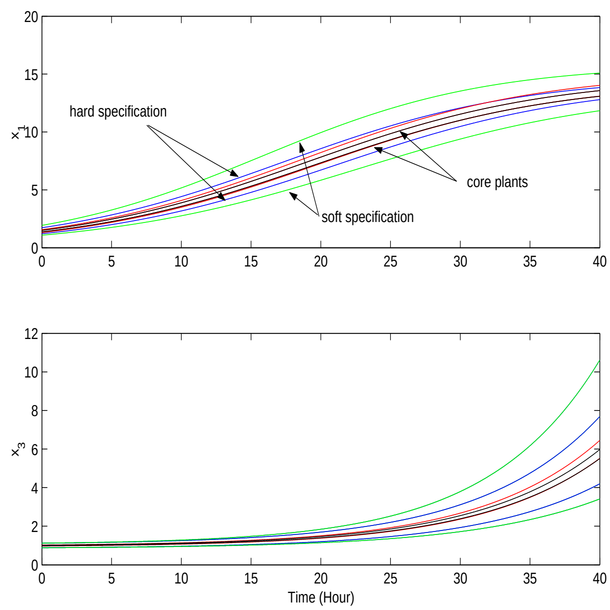


Fig. 5.41. Biomass and volume space.

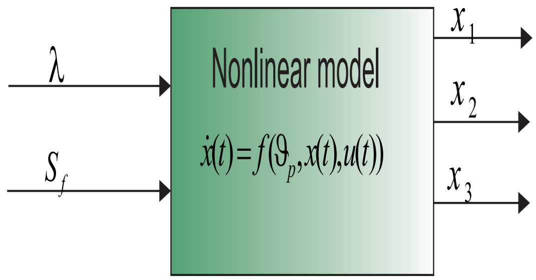


Fig. 5.42. Open-loop control system.

In Figures 5.43 and 5.44 we can see that hard controllers are able to maintain the microorganisms growth rate within the interval $\mu(x_2) = [0.09, 0.13]$ after a certain time, under parametric uncertainty in the plant parameters. The initial conditions considered for the state variables were $x_1(0) = 0.7, x_2(0) = 0.01, x_3(0) = 1$.

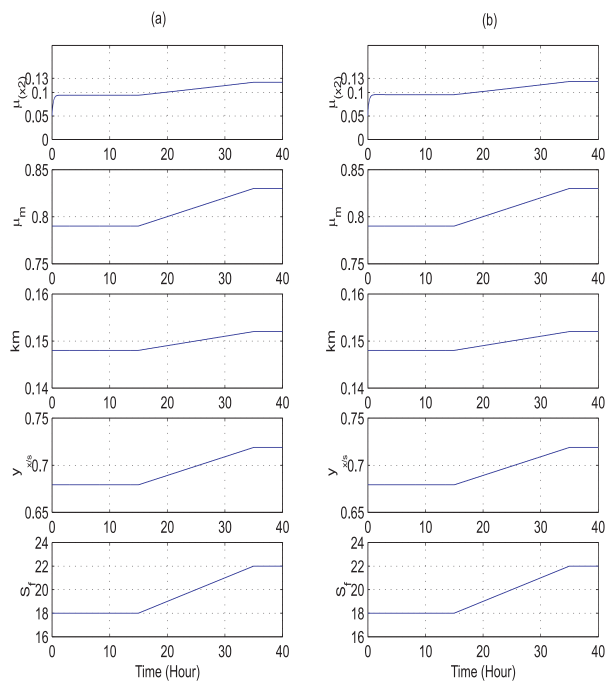


Fig. 5.43. (a) Open-loop response with $\lambda = 0.0077$ (b) Open-loop response with $\lambda = 0.0078$.

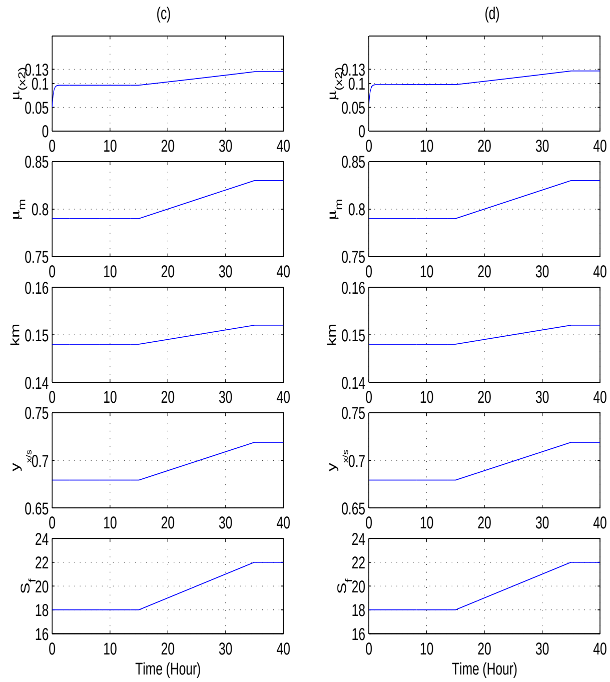


Fig. 5.44. (c) Open-loop response with $\lambda = 0.0079$ (b)Open-loop response with $\lambda = 0.0080$.

5.3.7 Computation of state feedback controller parameters.

In this section we will design a feedback controller based on differential flatness considering the system model 5.52. From second equation in (5.52) we can obtain λ_{min} and λ_{max} in function of the flat outputs.

$$\lambda_{max} = \frac{1}{(S_{fmax} - S_{fmin})} \left[\frac{\dot{x}_2}{x_1} + \frac{\dot{x}_1}{y_{x/s} x_1} - \frac{\dot{x}_3}{x_3 x_1} \left(- \frac{x_1}{y_{x/s}} + S_{fmin} - x_2 \right) \right] \quad (5.82)$$

$$\lambda_{min} = \lambda - \lambda_{max} \quad (5.83)$$

because we desire to drive the states x_1 and x_3 to their equilibrium values $\bar{x}_1 = \bar{y}_1$ and $\bar{x}_3 = \bar{y}_2$ the differential parametrization of the input immediately suggest the stabilizing controller as:

$$\begin{aligned} \lambda_{max} &= \frac{1}{(S_{fmax} - S_{fmin})} \left[\frac{\dot{x}_2}{x_1} + \frac{\dot{x}_1}{y_{x/s} x_1} - \frac{\dot{x}_3}{x_3 x_1} \left(- \frac{x_1}{y_{x/s}} + S_{fmin} - x_2 \right) \right] \\ \dot{x}_2 &= \frac{k_m \mu_m (x_1 x_3 (x_3 \ddot{x}_1 + 2 \dot{x}_1 \dot{x}_3 + x_1 \ddot{x}_3 - (x_3 \dot{x}_1 + \dot{x}_3 x_1)^2)}{(\mu_m x_1 x_3 - (x_3 \dot{x}_1 + \dot{x}_3 x_1))^2} \end{aligned} \quad (5.84)$$

the closed-loop dynamics for the flat outputs can be fixed as follows:

$$\begin{aligned} \ddot{x}_1 &= -k_{2x1}\dot{x}_1 - k_{1x1}(x_1 - \bar{x}_1) \\ \ddot{x}_3 &= -k_{2x3}\dot{x}_3 - k_{1x3}(x_3 - \bar{x}_3) \end{aligned} \tag{5.85}$$

which can be made asymptotically stable by a suitable choice of the design parameters k_{2x1}, k_{1x1}, k_{2x3} and k_{1x3}.

As the objetive is to track a reference model, we fixed $\ddot{x}_3 = u_1$ and $\ddot{x}_1 = u_2$ where u_1 and u_2 are defined by two PI controllers as follows:

$$\begin{aligned} u_1 &= k_{pq1}e_{x3} + k_{iq1}\int e_{x3}d\tau,\ e_{x3} = x_{3r} - x_3 \\ u_2 &= k_{pq2}e_{x1} + k_{iq2}\int e_{x1}d\tau,\ e_{x1} = x_{1r} - x_1 \end{aligned} \tag{5.86}$$

In Figure 5.45 the control diagram is indicated.

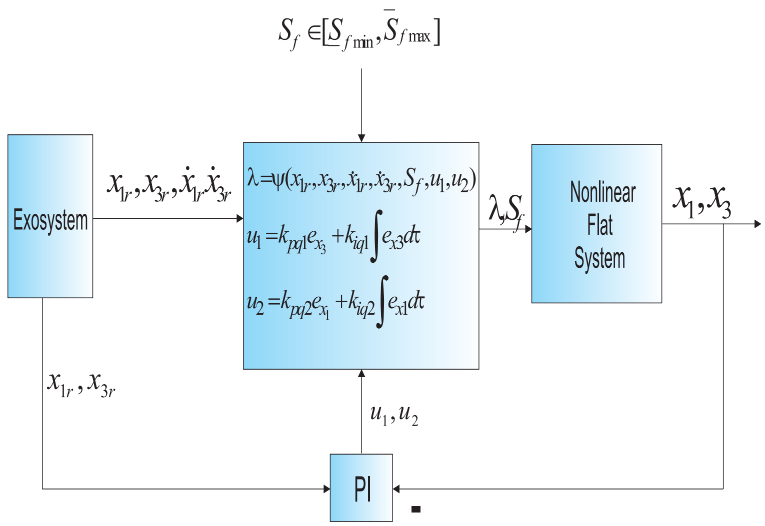


Fig. 5.45. Feedback controller.

A complete reference of nonlinear robust control of biotechnological processes can be consulted in E. Picó-Marco's doctoral thesis **Pico-Marco (2004)**. There, two PI controllers were used considering to λ and S_f as control inputs. λ was associated with volume x_3 and S_f with biomass concentration x_1 as it is indicated in Figure 5.46.

being λ_n a nominal feedforward selected from Table 5.15 with the $\forall\exists$-solution set $\Sigma_{\forall\exists}$ of process parameters ϑ_p of 4.23 and S_{fin} is fixed a priori to $S_{fin} \in [S_{fmin}, S_{fmax}]$. The references, which are the trajectories that must be followed by the flat outputs, are generated by the exosystem 5.57.

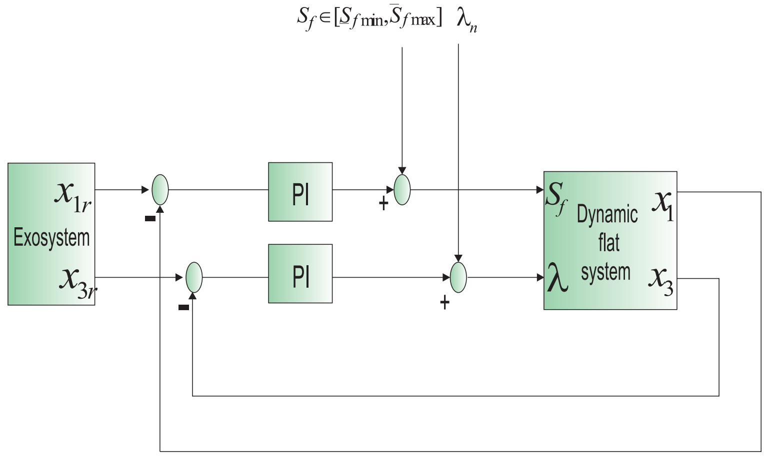


Fig. 5.46. Feedback control scheme.

It is desired to find the solution set of controller parameters in order to guarantee the satisfaction of specifications in function of the microorganisms growth rate $\mu(x_2)$.

To find the $\Sigma_{\forall\exists}$-solution set of controller parameters ϑ_c for which robust performance holds, the Quantified Constraints Satisfaction Problem is proposed as:

$$\Sigma_{\vartheta c} = \{\vartheta_c \in R | \forall(t \in \boldsymbol{t}^{'})\forall(\vartheta_p \in \boldsymbol{\vartheta}_p^{'})\exists(\theta_q \in \boldsymbol{\theta}_q^{'})\mu_{cl}(t, \vartheta_p, \vartheta_c) \subseteq M(t, \theta_q)\} \tag{5.87}$$

being $\mu_{cl}(t, \vartheta_p, \vartheta_c)$ the closed-loop microorganisms growth rate and $M(t, \theta_q)$ a first order reference model of the form

$$M(t, \theta_q) = \boldsymbol{\theta}_{q1}(1 - \boldsymbol{\theta}_{q2} \exp(-\boldsymbol{\theta}_{q3} t)) \tag{5.88}$$

where $\theta_q = (\boldsymbol{\theta}_{q1}, ..., \boldsymbol{\theta}_{qj})$, $\boldsymbol{\theta}_{qi} = \left[\underline{\theta}_{qi}, \overline{\theta}_{qi}\right]$, $i = 1...j$, is a box of specification parameters.

In this Quantified Constraints Satisfaction Problem, the constraint has the form $C(t, \vartheta_c, \vartheta_p, \theta_q) := \{\mu_{cl}(t, \vartheta_p, \vartheta_c) \subseteq M(t, \theta_q)\}$. The three rules to solve the problem are the followings:

Rule 1:

$$\forall(\vartheta_c \in \boldsymbol{\vartheta}_c^{'})\forall(t \in \boldsymbol{t}^{'})\forall(\vartheta_p \in \boldsymbol{\vartheta}_p^{'})\exists(\theta_q \in \boldsymbol{\theta}_q^{'})C_{cl}(t, \vartheta_c, \vartheta_p, \theta_q) \Leftrightarrow \vartheta_c \subseteq \Sigma \tag{5.89}$$

Rule 2:

$$\neg(\forall(t \in \boldsymbol{t}')\forall(\vartheta_p \in \boldsymbol{\vartheta}'_p)\exists(\theta_q \in \boldsymbol{\theta}'_q)\exists(\vartheta_c \in \boldsymbol{\vartheta}'_c)C_{cl}(t,\vartheta_c,\vartheta_p,\theta_q)) \Leftrightarrow \vartheta_c \subseteq \bar{\Sigma} \tag{5.90}$$

Rule 3: otherwise, ϑ_c is undefined.

The variable numeric sets, domains and quantifiers used to find closed-loop controller parameters in order to guarantee the satisfaction of specifications are the following:

- Space of controller parameters to solve $\vartheta_c = \{k_{pq1}, k_{iq1}, k_{pq2}, k_{iq2}\}$;
- Set of controller parameter domains $\boldsymbol{\vartheta}'_c = \{\boldsymbol{k}'_{pq1}, \boldsymbol{k}'_{iq1}, \boldsymbol{k}'_{pq2}, \boldsymbol{k}'_{iq2}\}$;
- Set of universally quantified parameters t,$\vartheta_p = \{k_m, \mu_m, y_{x/s}\}$;
- Set of existentially quantified parameters , $\theta_q = \{\theta_{q1}, \theta_{q2}, \theta_{q3}\}$;
- Set of universally quantified parameter domains $\boldsymbol{t}'$,$\boldsymbol{\vartheta}'_p = \{\boldsymbol{k}'_m, \boldsymbol{\mu}'_m, \boldsymbol{y}'_{x/s}\}$;
- Set of existentially quantified parameters domains , $\boldsymbol{\theta}'_q = \{\boldsymbol{\theta}'_{q1}, \boldsymbol{\theta}'_{q2}, \boldsymbol{\theta}'_{q3}\}$;
- Constraints set

$$C_{cl}(t,\vartheta_c,\vartheta_p,\theta_q) = \{\mu_{cl}(t,\vartheta_p,\vartheta_c) \subseteq M(t,\theta_q)\} \tag{5.91}$$

Finally the solution set $\Sigma_{\forall\exists}$ is

$$\begin{aligned} &\Sigma_{\forall\exists} = \{k_{pq1} \times k_{iq1} \times k_{pq2} \times k_{iq2} | \forall(t \in \boldsymbol{t}')\forall(k_m \in \boldsymbol{k}'_m) \\ &\forall(y_{x/s} \in \boldsymbol{y}'_{x/s})\forall(\mu_m \in \boldsymbol{\mu}'_m) \\ &\exists(\theta_{q1} \in \boldsymbol{\theta}'_{q1})\exists(\theta_{q2} \in \boldsymbol{\theta}'_{q2})\exists(\theta_{q3} \in \boldsymbol{\theta}'_{q3}) \\ &(\mu_{cl}(t,\vartheta_p,\vartheta_c) \subseteq M(t,\theta_q))\} \end{aligned} \tag{5.92}$$

now we are going to consider a hard $\boldsymbol{\mu}(x_2) = [0.10, 0.12]$ and soft $\boldsymbol{\mu}(x_2) = [0.09, 0.13]$ specification for closed-loop microorganisms growth rate and parameters $\boldsymbol{\theta}_{q1} = [0.10, 0.12]$, $\boldsymbol{\theta}_{q2} = [0.5, 1]$ and $\boldsymbol{\theta}_{q3} = [0.1, 1]$ for the reference model of the hard specification and $\boldsymbol{\theta}_{q1} = [0.09, 0.13]$, $\boldsymbol{\theta}_{q2} = [0.5, 1]$ and $\boldsymbol{\theta}_{q3} = [0.1, 1]$ for the soft specification. In Figures 5.47 and 5.48 are depicting solution sets of closed-loop controller parameters that guarantee the fulfillment specifications after a certain time interval.

Selecting the controller parameters $k_{pq1} = 100$, $k_{iq1} = 15$, $k_{pq2} = 100$ and $k_{iq2} = 15$ of the subpaving of figure 5.47 for soft specifications as well as the controller parameters $k_{pq1} = 25$, $k_{iq1} = 1$, $k_{pq2} = 25$, $k_{iq2} = 1$ of the subpaving of figure 5.48 for hard specifications, results indicated in Figure 5.49 we can see that the hard feedback controller maintains the desired output within the hard region and the soft feedback controller maintains the output within the soft region.

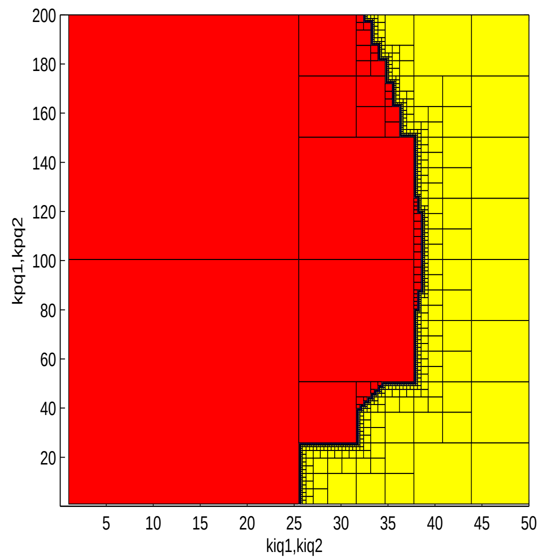


Fig. 5.47. Closed-loop controller parameters that guarantee the fulfillment of soft $\boldsymbol{\mu}(x_2) = [0.09, 0.13]$ specifications. The red boxes represent solution set. The yellow boxes are outside of the solution set and the black boxes are undefined.

5.3.8 Robustness test

With the purpose of being able to verify the design, a controller with parameters values $k_{pq1} = 25, k_{iq1} = 1, k_{pq2} = 25, k_{iq2} = 1$ has been chosen. For this controller, the temporary response for different precise systems $\boldsymbol{\mu}_m = [0.755, 0.865], \boldsymbol{k}_m = [0.1333, 0.1665], \boldsymbol{y}_{x/s} = [0.6443, 0.7543]$ from the soft plants family has been simulated. In Figure 5.50 it is possible to appreciate that the specifications are fulfilled.

5.4 Conclusions

A method has been developed to treat the uncertainty in biological systems and the solution of Quantified Constraints Satisfaction problems. The theory of nonlinear flat systems was applied to obtain a set of algebraic equations and these were analized in the field of the Modal Interval Analysis. Different optimization problems were raised to obtain solution sets such as: Admissible maximum uncertainty by a nominal controller, a set of robust controllers and attainable specifications by some controllers of the family. The solution sets were obtained with different Quantified Sets Inversion Algorithms. In the different applications, the constraints were verified at every point of time within the considered domains.

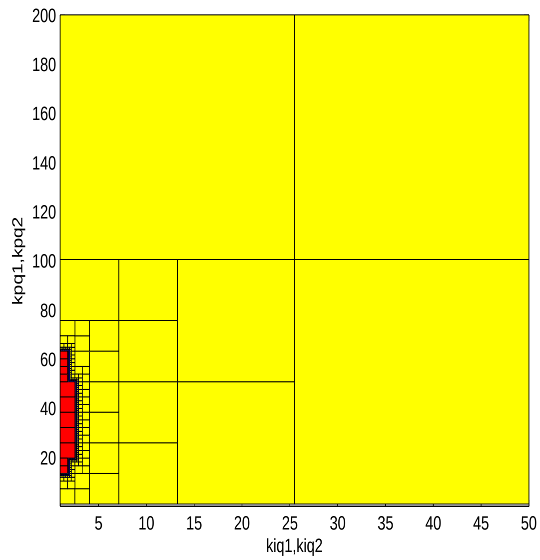


Fig. 5.48. Closed-loop controller parameters that guarantee the fulfillment of hard $\boldsymbol{\mu}(x_2) = [0.10, 0.12]$ specifications. The red boxes represent solution set. The yellow boxes are outside of the solution set and the black boxes are undefined.

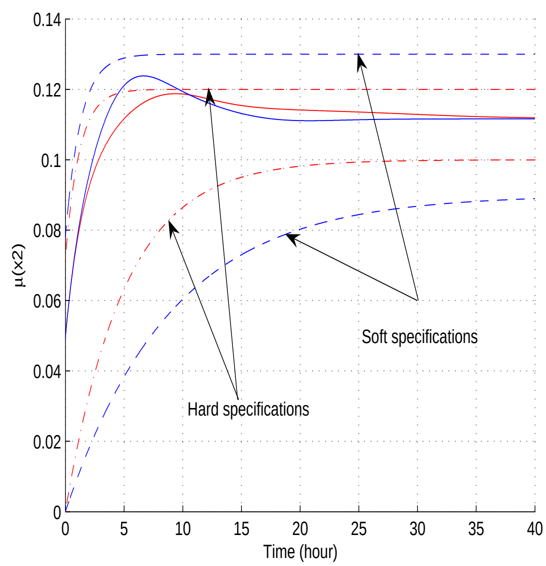


Fig. 5.49. Control of the microorganisms growth rate

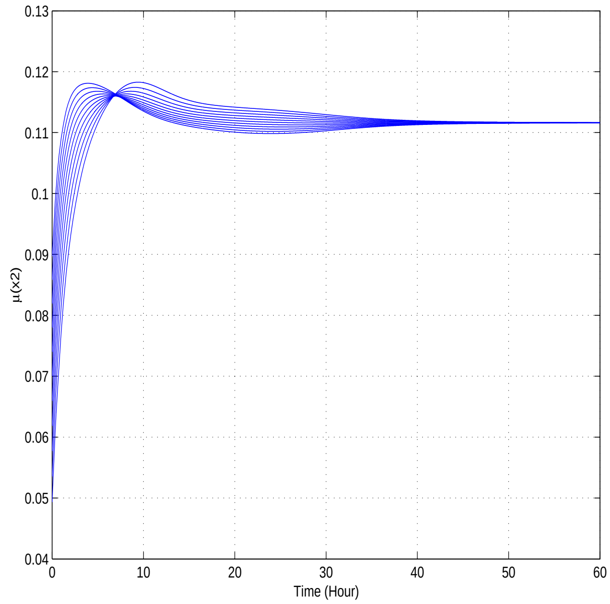


Fig. 5.50. Control of the microorganisms growth rate for a soft plants family

6 Conclusions and future works.

In this Chapter the fulfillment of the objectives formulated in this thesis are commented and the main contributions are summarized. In the second part, the future works are mentioned.

6.1 Conclusions

A new methodology of robust controllers design for nonlinear flat systems was developed.

For the first time, the theory of flat systems was applied to the field of the Modal Interval Analysis. A set of constraints in terms of sets relation between specifications, states, controllers and plants were proposed in different Quantified Constraints Satisfaction Problems. Hard and soft specification spaces of the flat outputs were defined in terms of hard and soft specification intervals of a fuzzy set. From the hard and soft specifications space of the flat outputs, states and controllers specifications spaces were obtained and used in a constraints set.

Three main formulations were proposed: 1) Quantified Constraints Satisfaction Problems to determine the maximum uncertainty admitted by a nominal controller, 2) Quantified Constraints Satisfaction Problems to find a family of robust controllers and 3) Quantified Constraints Satisfaction Problems to obtain solutions spaces referred to the attainable specifications by a family of controllers.

Different Quantified Sets Inversion Algorithms were implanted to obtain the solution sets referred to: the maximum uncertainty admitted by a nominal controller, family of robust controllers and the attainable specifications by a family of controllers. The constraints were evaluated for each point of time within a specified domain.

A Quantified Sets Inversion Algorithm was applied to an interval linear system. The example was developed for different types from parameter quantifications. The parameter space (free variables) was partitioned in conjuntion with the existentially quantified parameters and the constraints were evaluated for each partition. All the ranges of the universally quantified parameters were used in the constraints.

Robustness test to the family of robust controllers was made in open-loop using the nonlinear model of the system. In the simulation each robust controller was tested under parametric uncertainty of the plant. All the evaluated controllers fulfilled the robustness test maintaining the output of the system within the pre-established region.

The approach was applied to a DC motor, to a simple pendulum and to a fed-batch bioreactor. In the case of the DC motor and simple pendulum the flat output spaces were specified only in terms of the specification parameters. However, in the case of the fed-batch bioreactor the flat output spaces were specified in terms of the specification parameters and other aditional parameters such as: a nominal controller and initial conditions in biomass and volume.

A feedback controller based on differential flatness was designed to control the simple pendulum. The design parameters were determined from viewpoint of the input-output behaviour of the feedback system. In all cases, the controller parameters were tuned in order to obtain robust performance under parametric uncertainty. Feedback controllers showed better robustness performances under parametric uncertainty of the plant by using parameters located in the center from paving solutions.

6.2 Future works

Some future works that can be developed are the following:

The approach of robust possibilistic control can be extended for several α-cuts of the fuzzy set of specifications. New tools based on modal intervals to deal directly with fuzzy sets can be developed. Additional information about the distribution of the parameters within the specified uncertainty range can be analized in future works. The specification and input spaces can be considered as universes of discourse. These universes can be partitioned to form fuzzy subsets. A set of rules can be defined from the fuzzy subsets. So, a fuzzy model of the plant can be implemented, and the analysis and design of fuzzy controllers for nonlinear flat systems can be studied in future works. The technique to solve optimization problems with the approach of set inclusion can be extended to solve some control problems for nonlinear general systems. The approach can be extended to feedback control schemes where the inverse of the plant is required such as Internal Model Control (IMC).

A Modal intervals

A.1 Definitions and properties

Modal Interval Analysis (MIA) **Gardeñes et al. (1985)**; **Gardeñes et al. (1995)**; **Gardeñes et al. (2001)**, extends real numbers to intervals, identifying the intervals by the predicates that the real numbers fulfill, unlike classical interval analysis identifies the intervals by the set of real numbers that they contain. In the following results, some of the properties of the modal intervals are stated. The proofs of all the results presented in this appendix as well as other recent results of modal intervals can be found in **SIGLA/X**.

Given the set of closed intervals of R, $I(R) = \{[a, b]^{'} | a, b \in R, a \leq b\}$, and the set of logical existential and universal quantifiers $\exists, \forall$, a modal interval is defined by a pair:

$$X := (X^{'}, QX) \tag{A.1}$$

in which $X^{'} \in I(R)$ and $QX \in \{\exists, \forall\}$. $X^{'}$ is called the extension and QX is the modality. The set of modal intervals will be denoted by $I^*(R)$. On similar form that real numbers are associated in pairs having the same absolute value but opposite signs, the modal intervals are associated in pairs too, each member corresponds to the same closed interval of the real axis but each one having the opposite selection modalities, existential or universal.

The universal and existential quantifiers are represented by $\forall$ and $\exists$. Moreover, since the quantifiers are operators which transform real predicates into interval predicates, it will be written $\exists(x, X^{'})P(x)$ and $\forall(x, X^{'})P(x)$, indicating both arguments, the real index x and the argument $X^{'}$.

The canonical notation for modal intervals is:

$$[a, b] := \left\{ \begin{array}{l} ([a_1, a_2]^{'}, \exists) \; if \;\; a_1 \leq a_2 \\ ([a_2, a_1]^{'}, \forall) \; if \;\; a_1 \geq a_2 \end{array} \right\}. \tag{A.2}$$

A modal interval $([a_1, a_2]^{'}, \exists)$ is called existential interval or proper interval whereas $([a_2, a_1]^{'}, \forall)$ is called universal interval or improper interval.

A.2 Modal interval relations and operations

Given the modal interval $X = (X^{'}, QX) \in I^{*}(R)$, then

$$\begin{aligned} &Set(X^{'}, QX) := X^{'}, \\ &Mod(X^{'}, QX) := QX. \end{aligned} \tag{A.3}$$

The canonical coordinates of modal intervals are defined by

$$\begin{aligned} Inf(X) &:= \begin{cases} \underline{if}\ Mod(X) = \exists\ \underline{then}\ min(Set(X)) \\ \underline{if}\ Mod(X) = \forall\ \underline{then}\ max(Set(X)) \end{cases} \\ Sup(X) &:= \begin{cases} \underline{if}\ Mod(X) = \exists\ \underline{then}\ max(Set(X)) \\ \underline{if}\ Mod(X) = \forall\ \underline{then}\ min(Set(X)) \end{cases} \end{aligned} \tag{A.4}$$

The canonical notation of modal intervals is introduced by the definition

$$[a, b] := \begin{cases} \underline{if}\ a \leq b\ \underline{then}\ ([a, b]^{'}, \exists) \\ \underline{if}\ a \geq b\ \underline{then}\ ([b, a]^{'}, \forall) \end{cases} \tag{A.5}$$

Canonical notations for the modal and canonical coordinates are

$$\begin{aligned} &Inf([a, b]) = a, \\ &Sup([a, b]) = b, \\ &Set([a, b]) = [min(a, b), max(a, b)], \\ &Mod([a, b]) = \begin{cases} \underline{if}\ a \leq b\ \underline{then}\ \exists, \\ \underline{if}\ a \geq b\ \underline{then}\ \forall. \end{cases} \end{aligned} \tag{A.6}$$

For example, the modal coordinates of the modal interval $([2, 3]', \forall)$ are $Set([2, 3]^{'}, \forall) = [2, 3]^{'}$ and $Mod([2, 3]^{'}, \forall) = \forall$.

The canonical coordinates are $Inf([2, 3]^{'}, \forall) = 3$ and $Sup([2, 3]^{'}, \forall) = 2$. The canonical notation is $[3, 2] = ([2, 3]^{'}, \forall)$.

With this canonical notation, "natural" modal interval sets are

$$\begin{aligned} &I^{*}(R) = \{[a, b] | a \in R, b \in R\}, \\ &I_{\exists}(R) = \{[a, b] \in I^{*}(R) | a \leq b\}, \\ &I_{\forall}(R) = \{[a, b] \in I^{*}(R) | a \geq b\}, \\ &I_{p}(R) = \{[a, b] \in I^{*}(R) | a = b\}. \end{aligned} \tag{A.7}$$

An interval $[a, b] \in I_{\exists}(R)$ is a quantified as a "proper interval"; an interval $[a, b] \in I_{\forall}(R)$ as "improper"; an interval $[a, a] \in I_{p}(R)$ as a point-interval.

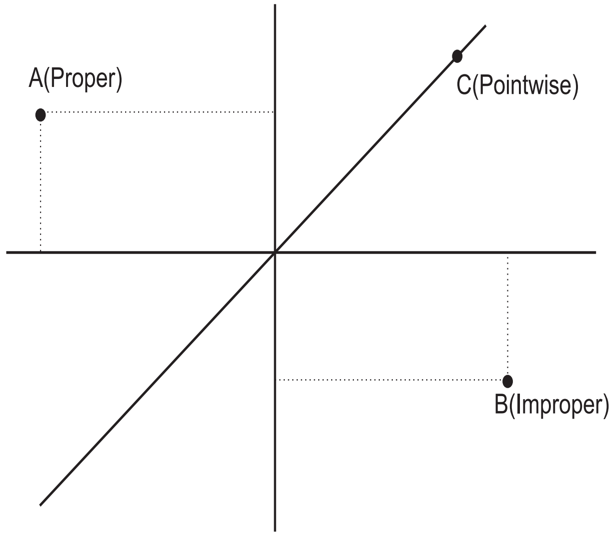


Fig. A.1. (Inf-sup)-diagram

The modal quantifier Q associates to every real predicate $P(.) \in Pred(R)$ has a unique hereditary interval predicate: for a variable x on R and $(X^{'}, QX) \in I^{*}(R)$

$$Q(x(X^{'}, QX))P(x) := QX(x, X^{'})P(x) \tag{A.8}$$

It will be obtained, by example,

$$Q(x, ([-3, 1]^{'}, \exists))x \geq 0 \ := \exists(x, [-3, 1]^{'})x \geq 0,$$
$$Q(x, ([1, 2]^{'}, \forall))x \geq 0 \ := \forall(x, [1, 2]^{'})x \geq 0.$$

Using the canonical notation, the operation of the Q-quantifier is displayed as follows:

$$Q(x, [a, b]) = \begin{cases} \underline{if}\ a \leq b\ \underline{then}\ \exists(x, [a, b]^{'}), \\ \underline{if}\ a \geq b\ \underline{then}\ \forall(x, [b, a]^{'}). \end{cases} \tag{A.9}$$

as well as a classic interval is identifiable with a predicate: $X^{'} \leftrightarrow x \in X^{'}$, a modal interval also identifies a set of predicates: $X = (X^{'}, QX) \leftrightarrow Pred(X)$, which allows to extend the relations of equality and inclusion of the classic intervals to the modal intervals. The set of real predicates accepted by a modal interval corresponds to

$$Pred(x(X^{'}, QX)) := \{P(.) \in Pred(R) | Q(x, (X^{'}, QX))P(x)\} \tag{A.10}$$

The parallel relation to the inclusion of two set-theoretical of intervals can be introduced into system of modal intervals.

It will be defined the set-theoretical of modal inclusion, for $A, B \in I^*(R)$, as

$$A \subseteq B \Leftrightarrow Pred(A) \subseteq Pred(B) \tag{A.11}$$

Thus the inclusion among modal intervals, $A \subseteq B$, makes valid the implication $Q(x, A)P(x) \Rightarrow Q(x, B)P(x)$ for any property $P(x)$ on the real numbers. In terms of the canonical notation

$$[a_1, a_2] \subseteq [b_1, b_2] \Leftrightarrow (a_1 \geq b_1, a_2 \leq b_2) \tag{A.12}$$

is identical to the $\subseteq$-relation for $I(R)$. Naming the existential intervals "proper intervals" to the universal ones "improper intervals" that comes from the identification of $I_{\exists}(R)$ and $I_{\forall}(R)$ suggested by their coinciding inclusions programming theorems.

In a dual way it is possible to define the set of real "co predicates" or predicates rejected by a modal interval, given by its modal coordinates $(X^{'}, QX)$

$$Copred(X^{'}, QX) := \{P(.) \in Pred(R) | \neg Q(x, (X^{'}, QX))P(x)\} \tag{A.13}$$

There exists a complementarity between Pred and Copred by means of the duality operator

$$Dual([a, b]) := [b, a] \tag{A.14}$$

since $A \subseteq B \Leftrightarrow Dual(A) \supseteq Dual(B) \Leftrightarrow Copred(A) \supseteq Copred(B)$.

The "less or equal" relation, generated by the completation of the modal inclusion, is defined as follows, in the (Inf, Sup)-diagram, a representation for "inclusion" and "less or equal" relationships it is indicated in Figure A.2

The system $(I^*(R), \subseteq)$ is a lattice, the infimum and the supremum are named meet and join operators represented by $\wedge$ and $\vee$, respectively: for a family of modal intervals $A(i)$ with $i \in I$ it will be defined

$$\begin{aligned} &\wedge(i, I)A(i) = A \in I^*(R) \textit{ is such that } \forall(i, I)X \subseteq A(i) \Leftrightarrow X \subseteq A, \\ &\vee(i, I)A(i) = B \in I^*(R) \textit{ is such that } \forall(i, I)X \supseteq A(i) \Leftrightarrow X \supseteq B \end{aligned} \tag{A.15}$$

annotated $A \wedge B$ and $A \vee B$ for the corresponding two-operands case.

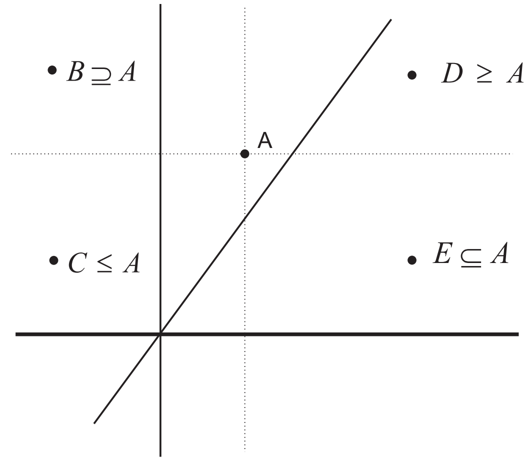


Fig. A.2. Inclusion and less than relations

In terms of the canonical notations $A(i) = [a_1(i), a_2(i)]$

$$\begin{aligned} &\wedge(i,I)A(i) = [max(i,I)a_1(i), min(i,I)a_2(i)],\\ &\vee(i,I)A(i) = [min(i,I)a_1(i), max(i,I)a_2(i)]. \end{aligned} \tag{A.16}$$

The system $I^*(R), \leq$ is a lattice, the infimum and the supremum are called Bottom and Top operators: For a family of modal intervals $A(i)$ with $i \in I$ it will be defined

$$\begin{aligned} &Bottom(i,I)A(I) = min(i,I)A(i)\\ &min(i,I)A(i) = C \in I^*(R) \; is\ such\ that \;\; \forall(i,I)X \leq A(i) \Leftrightarrow X \leq C,\\ &Top(i,I)A(I) = max(i,I)A(i)\\ &max(i,I)A(i) = D \in I^*(R) \; is\ such\ that \;\; \forall(i,I)X \geq A(i) \Leftrightarrow X \geq D \end{aligned} \tag{A.17}$$

In terms of the canonical notations $A(i) = [a_1(i), a_2(i)]$

$$\begin{aligned} &min(i,I)A(i) = [min(i,I)a_1(i), min(i,I)a_2(i)]\\ &max(i,I)A(i) = [max(i,I)a_1(i), max(i,I)a_2(i)] \end{aligned} \tag{A.18}$$

In the Modal Interval Analysis, if $DI \subseteq R$ is a digital scale for the real numbers

$$I^*(DI) := \{[a,b] \in I^*(R) | a \in DI, b \in DI\} \tag{A.19}$$

the modal outer and inner roundings of $A \in I^*(R)$ are defined by

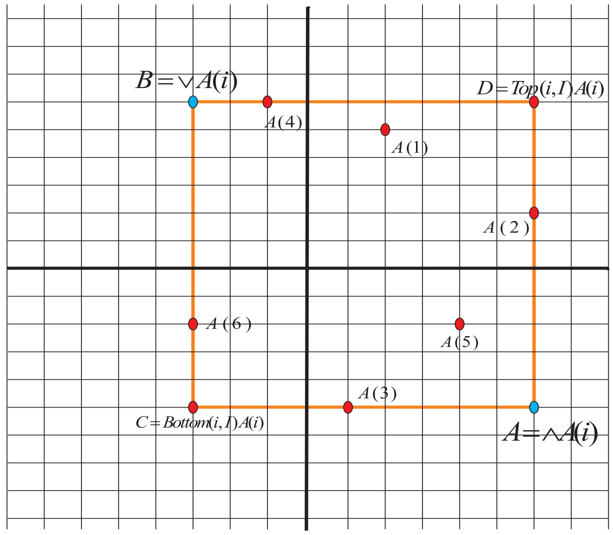


Fig. A.3. Join, meet, bottom and top

$$\begin{array}{l} Inn([a,b]) = [Right(a), Left(b)] \in I^*(DI), \\ Out([a,b]) = [Left(a), Right(b)] \in I^*(DI). \end{array} \tag{A.20}$$

The condition $Inn([a,b]) \subseteq [a,b] \subseteq Out([a,b])$ is fulfilled and equality $Inn(A) = Dual(Out(Dual(A)))$ makes unnecessary the implementation of the inner rounding.

The generalization to intervals with n components is direct. The set of n-dimensional intervals

$$I^*(R^n) := \{([a_1, b_1], ..., [a_n, b_n])|[a_1, b_1] \in I^*(R), ..., [a_n, b_n] \in I^*(R)\} \tag{A.21}$$

and the inclusion relation for two n-dimensional intervals $A = (A_1, ..., A_n)$, $B = (B_1, ..., B_n) \in I^*(R^n)$ is

$$A \subseteq B \Leftrightarrow (A_1 \subseteq B_1, ..., A_n \subseteq B_n). \tag{A.22}$$

A.3 Semantic extensions of continuous functions

The dual formulation of the modal intervals allows one to define two semantic interval functions, noted by f^* and f^{**} respectively. These play a very important role in the theory because they are in close relation with the modal interval extensions and provide meaning to the interval computations.

Definition **A.1.** * and **-semantic functions. If f is an R^n to R continuous function and $X \in I^*(R^n)$ then

$$f^*(X) := \vee(x_p, X_p^{'}) \wedge (x_i, X_i^{'})[f(x_p, x_i), f(x_p, x_i)]$$
$$= [min(x_p, X_p^{'})max(x_i, X_i^{'})f(x_p, x_i), max(x_p, X_p^{'})min(x_i, X_i^{'})f(x_p, x_i)],$$

$$f^{**}(X) := \wedge(x_i, X_i^{'}) \vee (x_p, X_p^{'})[f(x_p, x_i), f(x_p, x_i)]$$
$$= [max(x_i, X_i^{'})min(x_p, X_p^{'})f(x_p, x_i), min(x_i, X_i^{'})max(x_p, X_p^{'})f(x_p, x_i)]. \tag{A.23}$$

where (x_p, x_i) is the component splitting corresponding to $X = (X_p, X_i)$, with X_p a sub vector containing the proper components of X and X_i a sub vector containing the improper components of X.

If $X_i = 0$

$$f^*(X) = f^{**}(X)$$
$$= \vee(x, X^{'})[f(x), f(x)] = [min(x, X^{'})f(x), max(x, X^{'})f(x)] \tag{A.24}$$

and if $X_p = 0$

$$f^*(X) = f^{**}(X)$$
$$= \wedge(x, X^{'})[f(x), f(x)] = [max(x, X^{'})f(x), min(x, X^{'})f(x)] \tag{A.25}$$

which can be identified with the united extension of the classical interval extensions.

Using this definition, the expressions of the arithmetic operations by means of the interval bounds can be obtained. As example, this definition is applied in the following addition. Let us consider the function $f = x_1 + x_2$ and given $X_1 = [\underline{x}_1, \overline{x}_1]$ and $X_2 = [\underline{x}_2, \overline{x}_2]$, the semantic extensions f^* and f^{**} will be:

1. If X_1 and X_2 are proper:

$$f^*(X) = [min(x_1)min(x_2)(x_1 + x_2), max(x_1)max(x_2)(x_1 + x_2)]$$
$$= [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2] \tag{A.26}$$

$$f^{**}(X) = [min(x_1)min(x_2)(x_1 + x_2), max(x_1)max(x_2)(x_1 + x_2)]$$
$$= [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2] \tag{A.27}$$

2. If X_1 is proper and X_2 is improper:

$$f^*(X) = [min(x_1)max(x_2)(x_1 + x_2), max(x_1)min(x_2)(x_1 + x_2)] \\ = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2] \tag{A.28}$$

$$f^{**}(X) = [max(x_2)min(x_1)(x_1 + x_2), min(x_2)max(x_1)(x_1 + x_2)] \\ = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2] \tag{A.29}$$

3. If X_1 is improper and X_2 is proper:

$$f^*(X) = [min(x_2)max(x_1)(x_1 + x_2), max(x_2)min(x_1)(x_1 + x_2)] \\ = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2] \tag{A.30}$$

$$f^{**}(X) = [max(x_1)min(x_2)(x_1 + x_2), min(x_1)max(x_2)(x_1 + x_2)] \\ = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2] \tag{A.31}$$

4. If X_1 and X_2 are improper:

$$f^*(X) = [max(x_1)max(x_2)(x_1 + x_2), min(x_1)min(x_2)(x_1 + x_2)] \\ = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2] \tag{A.32}$$

$$f^{**}(X) = [max(x_1)max(x_2)(x_1 + x_2), min(x_1)min(x_2)(x_1 + x_2)] \\ = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2] \tag{A.33}$$

In this case $f^*(X) = f^{**}(X) = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2]$ for any modality of X_1 and X_2 and hence the sum of two intervals is noted $X_1 + X_2$ and in terms of the bounds of X_i becomes

$$X_1 + X_2 = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2] \tag{A.34}$$

For example, for the continuous real function $f(x_1, x_2) = x_1^2 + x_2^2$ the computation of the $*$-semantic and the $**$-semantic functions for $X = ([-1,1],[1,-1])$ yields the following results:

$$\begin{aligned} f^*([-1,1],[1,-1]) &= \vee(x_1,[-1,1]')\wedge(x_2,[-1,1]')[x_1^2+x_2^2, x_1^2+x_2^2] \\ &= \vee(x_1,[-1,1]')[x_1^2+1, x_1^2] = [1,1], \end{aligned}$$

$$\begin{aligned} f^{**}([-1,1],[1,-1]) &= \wedge(x_2,[-1,1]')\vee(x_1,[-1,1]')[x_1^2+x_2^2, x_1^2+x_2^2] \\ &= \wedge(x_2,[-1,1]')[x_2^2, 1+x_2^2] = [1,1]. \end{aligned}$$

For the real continuous function $g(x_1, x_2) = (x_1 + x_2)^2$ the corresponding $*$-semantic and $**$-semantic functions for $X = ([-1,1],[1,-1])$ do not have coincident values:

$$\begin{aligned} g^*([-1,1],[1,-1]) &= \vee(x_1,[-1,1]')\wedge(x_2,[-1,1]')[(x_1+x_2)^2,(x_1+x_2)^2] \\ &= [\underline{if}\ x_1 < 0\ \underline{then}(x_1-1)^2\ \underline{else}(x_1+1)^2, 0] = [1,0], \end{aligned}$$

$$\begin{aligned} g^{**}([-1,1],[1,-1]) &= \wedge(x_2,[-1,1]')\vee(x_1,[-1,1]')[(x_1+x_2)^2,(x_1+x_2)^2] \\ &= \wedge(x_2,[-1,1]')[0, \underline{if} x_2 < 0 \underline{then}(x_2-1)^2 \underline{else}(x_2+1)^2] \\ &= [0,1]. \end{aligned}$$

The definition can be applied to the other arithmetic operations and then the rules for their modal interval extensions are obtained:

- Difference:

$$X_1 - X_2 = [\underline{x}_1 - \overline{x}_2, \overline{x}_1 - \underline{x}_2] \tag{A.35}$$

- Division:

$$\frac{X_1}{X_2} = X_1 * \frac{1}{X_2}\ if\ 0 \notin X_2 \tag{A.36}$$

$$\frac{1}{X} = [\frac{1}{\overline{x}}, \frac{1}{\underline{x}}]\ if\ 0 \notin X \tag{A.37}$$

- Exponential and logarithm:

$$\begin{aligned} &ln\ X = ln[\underline{x}, \overline{x}] = [ln\ \underline{x}, ln\ \overline{x}]\ if\ X > 0 \\ &e^X = exp[\underline{x}, \overline{x}] = exp[exp\ \underline{x}, exp\ \overline{x}] \end{aligned} \tag{A.38}$$

- Absolute value:

$$|X| = |[\underline{x}, \overline{x}]| = \begin{cases} [\underline{x}, \overline{x}] & \text{if } X \geq 0 \\ [\overline{x}, \underline{x}] & \text{if } X \leq 0 \\ [0, max(|\underline{x}|, |\overline{x}|)] & \text{if } \underline{x} < 0 \text{ and } \overline{x} \geq 0 \\ [max(|\underline{x}|, |\overline{x}|), 0] & \text{if } \underline{x} \geq 0 \text{ and } \overline{x} < 0 \end{cases}$$

- Power of an interval:

$$X^n = [\underline{x}, \overline{x}]^n = \begin{cases} [\underline{x}^n, \overline{x}^n] & \text{if } n \text{ is odd} \\ [\underline{x}^n, \overline{x}^n] & \text{if } \underline{x} \geq 0 \text{ and } \overline{x} \geq 0 \text{ if } n \text{ is even} \\ [\overline{x}, \underline{x}] & \text{if } \underline{x} < 0 \text{ and } \overline{x} < 0 \text{ if } n \text{ is even} \\ [0, (max(|\underline{x}|, |\overline{x}|))^n] & \text{if } \underline{x} < 0 \text{ and } \overline{x} \geq 0 \text{ if } n \text{ is even} \\ [(max(|\underline{x}|, |\overline{x}|))^n, 0] & \text{if } \underline{x} \geq 0 \text{ and } \overline{x} < 0 \text{ if } n \text{ is even} \end{cases}$$

Two key theorems revealing completely the meaning of the interval results f^* and f^{**} and characterizing them as the key reference for the semantic interval extensions previously defined in logical terms.

THEOREM A.1. ***∗-Semantic Theorem.*** *If $A \in I^*(R^n)$, $f : R^n \to R$ is continuous on A' and there exists an interval which is called $F(A) \in I^*(R)$ then,*

$$f^*(A) \subseteq F(A) \Leftrightarrow \forall(a_p, A'_p)Q(z, F(A))\exists(a_i, A_i')(z = f(a_p, a_i)) \qquad (A.39)$$

◇

This interpretation can be read: "For all elements belonging to the proper intervals there exists at least one element in the improper intervals that fulfills the function".

Example A.1.$[10, 20] + [20, 15] = [30, 35]$ means

$$\forall(a, [10, 20]')\exists(f, [30, 35]')\exists(b, [15, 20]')(a + b = f) \qquad (A.40)$$

Dual semantics for proper and improper modal intervals are established by the dual semantic theorem.

THEOREM A.2. *(∗∗**-Semantic Theorem**). If $A \in I^*(R^n)$, $f : R^n \to R$ is continuous on A' and there exists an interval which is called $F(A) \in I^*(R)$ then,*

$$f^{**}(A) \supseteq F(A) \Leftrightarrow \forall(a_i, A'_i)Q(z, Dual(F(A)))\exists(a_p, A_p^{'})(z = f(a_p, a_i)) \tag{A.41}$$

◇

Example A.2.The semantic interpretation of

$$[1, 2] + [5, 7] = [6, 9] \tag{A.42}$$

in the context of Classic Interval Arithmetic is

$$\forall(a, [1, 2]^{'})\forall(b, [5, 7]^{'})\exists(f, [6, 9])(a + b = f) \tag{A.43}$$

In addition to this one, in the context of Modal Interval Analysis the $**$-semantic is

$$\forall(f, [6, 9]^{'})\exists(a, [1, 2]^{'})\exists(b, [5, 7]^{'})(a + b = f) \tag{A.44}$$

unfortunately, the computation of the $*$- and $**$-extensions is, in general, a difficult challenge hence the usual procedure is to find over bounded computations of f^* and under bounded computations of f^{**} which maintain the semantic interpretations.

When the continuous function f is a rational function, there are two modal rational extensions. They are obtained by using the computation program defined by the syntax tree of the function expression, in which the real arguments are transformed into interval arguments and real operators are transformed into their $*$ or $**$-semantic extensions. The function defined through the computational program indicated by the syntax of f is called modal rational function, $fR(A)$. In general, $fR(A)$ is not interpretable. The interpretation problem for modal rational functions, which are the core of numerical computing, consists in relating them by means of inclusion relations to the corresponding $*$- and $**$-semantic extensions, which have a standard meaning (defined by the Semantic Theorems) referring to their originary real continuous functions. Computations with $fR(A)$ must be done with external truncation of each operator to obtain inclusions $f^*(A) \subseteq fR(A)$, and with inner truncation to obtain inclusions $fR(A) \subseteq f^{**}(A)$.

There are several theorems that relate to the modal rational function $fR(A)$ and to the modal semantic extensions f^* and f^{**}. The following ones give two $*$ and $**$-interpretable coercion theorems.

Definition **A.2.** A component x_i of x is uni-incident in a rational function $f(x)$ if it occupies only one leaf of the syntactical tree of f; otherwise, x_i is multi-incident in $f(x)$.

Example A.3.In the rational function f of R^2 in R given by

$$f(x_1, x_2) = x_2 + \frac{x_1^2}{x_2} \tag{A.45}$$

x_2 is multi-incident and x_1 is uni-incident.

THEOREM A.3. ***If in*** $fR(A)$ ***all arguments are uni-incident***, *then*

$$f^*(A) \subseteq fR(A) \subseteq f^{**}(A) \tag{A.46}$$

In particular, if all the components of A *are uni-incident and with the same modality,*

$$f^*(A) = fR(A) = f^{**}(A) \tag{A.47}$$

◇

Modal rational interval functions are not interpretable but they also have the property of being isotonic, i.e., for $A_1 \subseteq B_1, ..., A_n \subseteq B_n$ the relation

$$fR(A_1, ..., A_n) \subseteq fR(B_1, ..., B_n) \tag{A.48}$$

keeps if there are not intervals containing zero in the division.

THEOREM A.4. ***If in*** $fR(A)$ ***there are multi-incident improper components and if*** AT^* ***is obtained from*** A ***by transforming, for every multi-incident improper component, all incidences but one into their duals, then***

$$f^*(A) \subseteq fR(AT^*) \tag{A.49}$$

if all components of A *are proper, then* $AT^* = A$ *and*

$$f^*(A) \subseteq fR(A) \tag{A.50}$$

when these computations are performed using digital numbers, appropriate roundings have to be made:

$$f^*(A) \subseteq Out(fR(AT^*)) \tag{A.51}$$

◇

THEOREM A.5. ***If in*** $fR(A)$ ***there are multi-incident proper components and if*** AT^{**} ***is obtained from*** A ***by transforming, for every multi-incident proper component, all incidences but one into their duals, then***

$$f^{**}(A) \supseteq fR(AT^{**}) \tag{A.52}$$

if all components of A *are improper, then* $AT^{**} = A$ *and*

$$f^{**}(A) \supseteq fR(A) \tag{A.53}$$

In this case, the roundings that have to be performed in order to maintain the semantics are:

$$f^{**}(A) \supseteq Inn(fR(A)) \tag{A.54}$$

An interpretable rational interval computation program $fR(A)$ *may nevertheless result in a loss of information far more important that the one produced by numerical roundings. Then it is very important to find out criteria to characterize the rational interval functions for which* $fR(A)$*, with an ideal computation (infinite precision), is such that*

$$f^{*}(A) = fR(A) = f^{**}(A) \tag{A.55}$$

In this case, it is said that $fR(.)$ *is optimal for* A*.*

There are several results which characterize the optimality of a modal rational function according to its monotonicity properties. ◇

Definition **A.3.** A continuous function $f(x, y)$, is an uniformly monotonic function with respect to x in a domain $(X^{'}, Y^{'}) \subseteq (R, R^m)$ if it is a monotonic function with respect to x in $X^{'}$ and keeps the same direction of monotonicity for all the values $y \in Y^{'}$.

Definition **A.4.** A continuous function $f(x, y)$, is a totally monotonic function with respect to a multi-incident variable x in a domain $(X^{'}, Y^{'}) \subseteq (R, R^m)$ if it is a uniformly monotonic function with respect to x in $X^{'}$ and, for every incidence of x considered as an independent variable, it is also a uniformly monotonic function.

THEOREM A.6. ***(Operators of n variables uniformly monotonous)*** *All continuous function* $f(x, y)$*, with* $(x, y) \in R^n$ *which is uniformly monotonous,* *isotonic regarding the sub vector x and anti tonic regarding the sub vector y in the domain* $(X^{'}, Y^{'})$*, is JM-commutative for* (X, Y) *and*

$$f^*(X, Y) = [f(Inf(X), Sup(Y)), f(Sup(X), Inf(Y))] \tag{A.56}$$

where $Inf(X) := (Inf(X_1), .., Inf(X_m))$,
$Sup(X) := (Sup(X_1), .., Sup(X_m))$ *and in a manner analogous to y.*

◇

THEOREM A.7. *(**optimal coercion for uni-modal arguments**) Let A be an interval vector and* fR *defined in the domain* $A^{'}$*, such that each incident of every multi-incident component is included in AD as an independent component, but transformed into its dual if the corresponding incidence-point has a monotonicity sense contrary to the global one of the corresponding A-component. Then*

$$f^*(A) = fR(AD) = f^{**}(A) \tag{A.57}$$

◇

THEOREM A.8. *(**$*$-partially optimal coercion**) Let A be an interval vector and* fR *defined in the domain* $A^{'}$ *and totally monotonic for a subset B of multi-incident components. Let* ADT^* *be the enlarged vector of A, such that each incidence of every multi-incident component of the subset with total monotonicity is included in* ADT^* *as an independent component, but transformed into its dual if the corresponding incidence-point has a monotonicity sense contrary to the global one of the corresponding B-component; for the rest, the multi-incident improper components are transformed into their dual in every incidence except one. Then*

$$f^*(A) \supseteq fR(ADT^{**}) \tag{A.58}$$

◇

These theorems, as well as other recent results of Modal Interval Analysis related to rational functions can be found in **SIGLA/X**.

A.4 Examples of range computations

The calculation of the optimal ranges for different functions and the application of various theorem are shown in the following examples.

Example A.4.Determination of the range of $f = x - x$ in the parameter space $X = [1, 2]$. The exact range is $f(x) = [0, 0]$ because all possible values of x fulfill $x - x = 0$. The modal rational function

$$fR(X) = X - X = [\underline{x}, \overline{x}] - [\underline{x}, \overline{x}] = [\underline{x} - \overline{x}, \overline{x} - \underline{x}] = [-1, 1] \tag{A.59}$$

is an over bounded approximation to $f^*(X)$. The following theorem allows to interpret the $*$-semantically results

Thus

$$\forall(x, [1, 2]')\exists(f, [-1, 1]')(x - x = f) \tag{A.60}$$

The function is monotonic with respect to x

$$\frac{df}{dx} = 1 - 1 = 0 \tag{A.61}$$

and with respect to every incidence of x considered as an independent variable

$$\begin{array}{l} \frac{df}{dx_1} = 1 > 0 \\ \frac{df}{dx_2} = -1 < 0 \end{array} \tag{A.62}$$

therefore it is totally monotonic and the theorem of optimal coercion for uni-modal arguments can be applied to obtain the exact range of the function:

such that

$$f^*(X) = f^{**}(X) = fR(XD) = X - Dual(X) = Dual(X) - X = [0, 0] \tag{A.63}$$

This result is semantically interpretable with the $*$-Semantic Theorem

Leaving

$$\forall(x, [1, 2]')\exists(f, [0, 0]')(x - x = f) \tag{A.64}$$

and it is also interpretable with the $**$-Semantic Theorem.

so

$$\forall(f, [0, 0]')\exists(x, [1, 2]')(x - x = f) \tag{A.65}$$

The following example shows that sometimes it is necessary to use splitting algorithms when the function is not totally monotonic.

Example A.5.Determination of the range of the function $f = x_1x_2 - x_1^2 - 2x_2$ in the parameter space $X_1 = [1, 2]$ and $X_2 = [3, 4]$. The range of the function is $R_f = [-5, -3.75]$. An over bounded approximation is obtained using the natural extension

$$\begin{aligned} &fR(X) = X_1X_2 - X_1X_1 - 2X_2 = [1, 2][3, 4] - [1, 2][1, 2] - 2[3, 4] \\ &= [3, 8] - [1, 4] - [6, 8] = [-1, 7] - [6, 8] = [-9, 1] \end{aligned} \tag{A.66}$$

so, $fR(X) \supseteq f^*(X)$

The function is monotonic with respect to x_2

$$\tfrac{df}{dx_2} = x_1 - 2 = [1, 2] - [2, 2] = [-1, 0] \le 0 \tag{A.67}$$

and with respect to every incidence of x_2 considered as an independent variable, the function f can be redefined as $f = x_1x_{21} - x_1^2 - 2x_{22}$ then

$$\begin{aligned} &\tfrac{df}{dx_{21}} = x_1 = [1, 2] > 0 \\ &\tfrac{df}{dx_{22}} = -2 < 0 \end{aligned} \tag{A.68}$$

since it is totally monotonic. Nevertheless, it is not totally monotonic with respect to x_1

$$\tfrac{df}{dx_1} = x_2 - 2x_1 = [3, 4] - 2[1, 2] = [3, 4] - [2, 4] = [-1, 2] \ni 0 \tag{A.69}$$

Therefore, theorem $*$-partially optimal coercion can be applied to x_2 to obtain a better approximation of the range of the function:

The results

$$\begin{aligned} &f^*(X) \subseteq fR(XDT^*) = X_1Dual(X_2) - X_1^2 - 2X_2 = [-8, -1] \\ &= [1, 2][4, 3] - [1, 2][1, 2] - 2[3, 4] \\ &= [4, 6] - [1, 4] - [6, 8] = [-8, -1] \end{aligned} \tag{A.70}$$

In this case, to obtain even better approximations of the exact range, the parameter space has to be split. The advantage provided by the theorem of $*$-partially optimal coercion, it indicates that only the variable x_1 must be divided. Moreover, the range at every sub-space can be computed more exactly because the theorem of $*$-partially optimal coercion has already been applied to x_2. In fact, the function whose range has to be determined now is

$$F = X_1[4, 3] - X_1^2 - [6, 8] \tag{A.71}$$

which is an interval function of one variable.

In conclusion, the number of sub-spaces considered to compute an approximation of the range of the function is smaller when modal intervals are used. This is illustrated by Vehí **Vehí (1998)**, where modal intervals combined with a branch and bound algorithm have been applied to the analysis and design of robust controllers.

The next example shows that sometimes it is necessary to compute the range of higher order derivatives in order to compute the range of a function. It has been seen that to study the monotonicity of a function is necessary to compute approximations to the range of its derivative. In the previous examples, the variables in the derivative are uni-incident so the range obtained using the natural extension is exact. If there are multi-incident variables in the derivative, this range is over bounded.

Example A.6.Determination of the range function $f = x_1^2 x_2 - 2x_1^2 + 2x_2^2$ in the parameter space $X_1 = [1,2]$ and $X_2 = [3,4]$. The range of the function is $R_f = [19, 40]$. An over bounded approximation is obtained by means of the natural extension

$$\begin{aligned} &fR(X) = X_1^2 X_2 - 2X_1^2 + 2X_2^2 \\ &fR(X) = [1,2][1,2][3,4] - 2[1,2][1,2] + 2[3,4][3,4] \\ &fR(X) = [3,16] - [2,8] + [18,32] = [13,46] \\ &f^*(X) \subseteq fR(X) \end{aligned} \tag{A.72}$$

The function is totally monotonic with respect to x_2, as it is monotonous with respect to x_2

$$\frac{df}{dx_2} = X_1^2 + 4X_2 = [1,2][1,2] + 4[3,4] = [1,4] + [12,16] = [13,20] > 0 \tag{A.73}$$

and with respect to every incidence of x_2 considered as an independent variable, being $f = x_1^2 x_{21} - 2x_1^2 + 2x_{22}^2$, thus

$$\begin{aligned} &\frac{df}{dx_{21}} = X_1^2 = [1,2][1,2] = [1,4] > 0 \\ &\frac{df}{dx_{22}} = 4X_2 = 4[3,4] = [12,16] > 0 \end{aligned} \tag{A.74}$$

It seems that the function f is not monotonic when the range of its derivative with respect to x_1 is computed by means of the natural extension:

$$\begin{aligned} &\frac{df}{dx_1} R(X) = 2X_1X_2 - 4X_1 = 2[1,2][1,2] - 4[1,2] \\ &= [6,16] - [4,8] = [-2,12] \ni 0 \end{aligned} \tag{A.75}$$

However, this is an over bounded approximation of this range, as x_1 is multi-incident, the exact range of the first derivative can be computed by studying the monotonicity of the second derivative:

$$\frac{d^2f}{dx_1^2} = 2X_2 - 4 = [2,4] > 0 \tag{A.76}$$

with respect to each multi-incident variable x_1 in $\frac{df}{dx_1}$

$$\begin{array}{l} \frac{d^2f}{dx_{11}^2} = 2X_2 = [6,8] > 0 \\ \frac{d^2f}{dx_{12}^2} = -4 < 0 \end{array} \tag{A.77}$$

as $\frac{d^2f}{dx_1^2} > 0$, $\frac{d^2f}{dx_{11}^2} > 0$ and $\frac{d^2f}{dx_{12}^2} < 0$ then it has to apply the dual operator to the term $2x_1^2$ to change the monotocity sense. Hence, the exact range of the first derivative is:

$$\frac{df}{dx_1}R(X) = 2X_1X_2 - 4Dual(X_1) = [6,16] - [8,4] = [2,8] > 0 \tag{A.78}$$

and the function f is totally monotonic:

$$\begin{array}{l} \frac{df}{dx_{11}} = 2X_1X_2 = [6,16] > 0 \\ \frac{df}{dx_{12}} = -4Dual(X_1) = [-4,-8] < 0 \end{array} \tag{A.79}$$

Then, the exact range of f is:

$$\begin{array}{l} fR(XD) = X_1^2X_2 - 2(Dual(X_1))^2 + 2X_2^2 \\ = [3,16] - 2([2,1][2,1]) + [18,32] \\ = [3,16] - [8,2] + [18,32] = [19,40] \\ f^*(X) = f^{**}(X) = fR(XD) \end{array} \tag{A.80}$$

Example A.7.Inclusion relations in time of rational functions. To consider the specifications $\vartheta_{s1} \subseteq \vartheta_{s2}$

$$\begin{array}{l} \vartheta_{s1} = \{\forall(\mu_r,[0.09,0.13]),\forall(\lambda_n,[0.0079,0.0079]),\forall(x_{10},[0.6,0.8])\}, \\ \vartheta_{s2} = \{\forall(\mu_r,[0.07,0.15]),\forall(\lambda_n,[0.0079,0.0079]),\forall(x_{10}=[0.5,0.9])\}. \end{array}$$

if we evaluated the rational extensions $fR_{x1}(\vartheta_{s1},t)$ and $fR_{x1}(\vartheta_{s2},t)$ with the equation (5.58), the inclusion relations $(fR_{x1}(\vartheta_{s1},t) \subseteq fR_{x1}(\vartheta_{s2},t))$ are fulfilled for $\forall(t,[0,40])$ as it is depicted in Figure A.4.

Example A.8.To consider the following equation

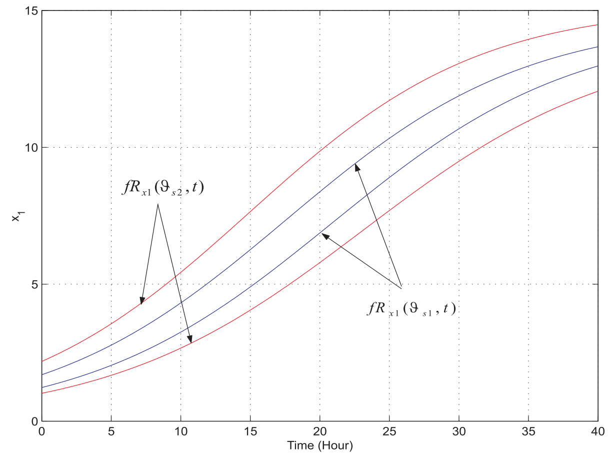


Fig. A.4. Inclusion relations of flat outputs

$$y = y_o + (y_1 - y_o)B(\tau)$$

being $B(\tau)$

$$\begin{aligned} B(\tau) &= \tau^5(252 - 1050\tau + 1800\tau^2 - 1575\tau^3 + 700\tau^4 - 126\tau^5) \\ \tau &= \frac{t-t_o}{t_1-t_o} \end{aligned}$$

to calculate the $*$ extension of the function considering $y_o = [0.2, 10]$ and $y_1 = [140, 150]$. The function y can be renamed as follows:

$$y = y_{o1} + (y_{11} - y_{o2})B(\tau)$$

the derivative of the function y respecting to y_{o1} is

$$y = 0$$

respecting to y_{11} is

$$y = B(\tau)$$

and finally to y_{o2} is

$$y = -B(\tau)$$

given the function y is creasing monotic respecting to y_{o1} and y_{11}, and decreasing monotonic respecting to y_{o2} then we will apply the dual operation to the term y_{o2} to change its monotonicy sense of decrasing to creasing.

$$y = y_{o1} + (y_{11} - Dual(y_{o2}))B(\tau)$$

In Figure A.5

$$f^* and the natural extension results are shown.$$

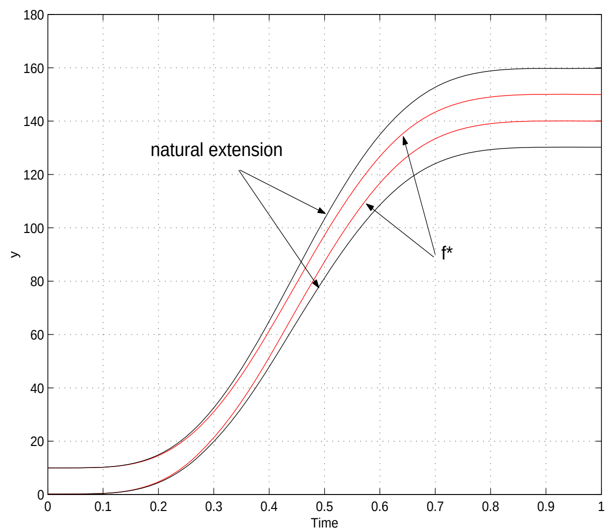


Fig. A.5. f^* and natural extension

Example A.9.Consider a function $F(q)$ depending on an uncertain parameter vector q belonging to an uncertainty domain Q. To find the semantic extension F^* of the following function

$$\begin{aligned} F(Q) &= -266.6q_2 - 3.2q_2^3 + 8.8q_1q_2^3 + 0.3q_1^4q_2^2 + 1.8q_1^3q_2 \\ &\cdot + 69.5q_1^2q_2^2 + 6.6q_1^3q_2^2 - 7.8q_1^2q_2^3 + 5.9q_1^3q_2^3 + 0.1q_1^4q_2^3 \\ &\cdot + 0.12q_1^4q_2^4 + 96.5q_1^2q_2 - 130.5q_1q_2^2 + 103.1q_2^2 + 119 \\ &\cdot + 110.6q_1 + 60.4q_1q_2 - 32.4q_1^2 - 0.1q_1^3q_2^4 + 0.03q_2^4q_1^2 \end{aligned} \tag{A.81}$$

The natural extension of the function $fR(Q)$ with uncertain intervals $q_1 = [0, 0.05]$, $q_2 = [0, 0.05]$ is computed using a library of modal intervals "IvalDb" developed by Herrero **Herrero et al. (2005)**. Here is only presented the main programm.

```
int main(int argc, char* argv[])
{
ivalDb q1;
ivalDb q2;
ivalDb fRQ
q1=ivalDb(0,0.5);
q2=ivalDb(0,0.5);

float x,y;

fRQ=-266.6*q2-3.2*(q2^3)+8.8*q1*(q2^3)+
0.3*(q1^4)*(q2^2)+1.8*(q1^3)*(q2)+69.5*(q1^2)*(q2^2)+
6.6*(q1^3)*(q2^2)-7.8*(q1^2)*(q2^3)+5.9*(q1^3)*(q2^3)+
0.1*(q1^4)*(q2^3)+0.12*(q1^4)*(q2^4)+96.5*(q1^2)*(q2)-
130.5*q1*(q2^2)+103.1*(q2^2)+119+110.6*q1+60.4*q1*q2-
32.4*(q1^2)-0.1*(q1^3)*(q2^4)+0.3*(q2^4)*(q1^2);

x=fRQ.GetInf();
y=fRQ.GetSup();

cout<<x;
cout<<y;
return 0;
}
```

the resulting interval is $fR([0,0.5][0,0.5]) = [-39.36, 232.59]$. The range is over bounded and contains zero. So applying the coercion theorem, it is possible to obtain an optimal expression for $F(q)$, if it is monotonic with respect to each variable q_i. This is checked by computing the partial derivatives of $F(q)$ with respect to q_1 and q_2.

$$\begin{array}{l} \frac{dF}{dq_1} = 8.8q_2^3 + 0.3(3q_1^3)q_2^2 + 1.8(3q_1^2)q_2 \\ \cdot + 69.5(2q_1)q_2^2 + 6.6(3q_1^2)q_2^2 + 5.9(3q_2^2) + 0.1(4q_1^3)q_2^3 \\ \cdot + 0.12(4q_1^3)q_2^4 + 96.5(2q_1)q_2 - 130.5q_2^2 \\ \cdot + 110.6 + 60.4q_2 - 32.4(2q_1) - 0.1(3q_1^2)q_2^4 + 0.03q_2^4(2q_1) \end{array} \tag{A.82}$$

The derivate of the function $F(q)$ respect to q_2 is

$$\begin{array}{l} \frac{dF}{dq_2} = -266.6 - 32.2(3q_2^2) + 8.8q_1(3q_2^2) + 0.3q_1^4(2q_2) + 1.8q_1^3 \\ \cdot + 0.12q_1^4(4q_2^3) + 96.5q_1^2 - 130.5q_1(2q_2) + 103.1(2q_2) \\ \cdot + 60.4q_1 - 0.1q_1^3(4q_2^3) + 0.03(4q_2^3)q_1^2 \end{array} \tag{A.83}$$

The ranges obtained are: $\frac{dFR}{dq_1} = [44.59, 210.05]$,$\frac{dFR}{dq_2} = [-335.71, -86.83]$. This means that the function $F(Q)$ is growing monotonous with respect to

q_1 and decreasing monotone respect to q_2. Since the function $F(Q)$ is totally monotonic, then applying the optimal coercion theorem for uni-modal arguments, the following steps are realised.

1. Applying the coercion theorem, the terms with positive coefficients that contain variables q_1 are separated from $F(Q)$, thus

$$\begin{aligned}
&F1(Q) = 8.8q_1q_2^3 + 0.3q_1^4q_2^2 + 1.8q_1^3q_2\\
&\cdot + 69.5q_1^2q_2^2 + 6.6q_1^3q_2^2 + 5.9q_1^3q_2^3 + 0.1q_1^4q_2^3\\
&\cdot + 0.12q_1^4q_2^4 + 96.5q_1^2q_2\\
&\cdot + 110.6q_1 + 60.4q_1q_2 + 0.03q_2^4q_1^2
\end{aligned} \tag{A.84}$$

   As each coefficient is positive and q_1 is growing monotone, then it has to change the monotony sense of q_2 of decreasing monotony to growing monotony with $Dual(q_2)$ for each term of the previous function $F1(Q)$

$$\begin{aligned}
&F1(Q) = 8.8q_1Dual(q_2)^3 + 0.3q_1^4Dual(q_2)^2 + 1.8q_1^3Dual(q_2)\\
&\cdot + 69.5q_1^2Dual(q_2)^2 + 6.6q_1^3Dual(q_2)^2\\
&\cdot + 5.9q_1^3Dual(q_2)^3 + 0.1q_1^4Dual(q_2)^3\\
&\cdot + 0.12q_1^4Dual(q_2)^4 + 96.5q_1^2Dual(q_2)\\
&\cdot + 110.6q_1 + 60.4q_1Dual(q_2) + 0.03Dual(q_2)^4q_1^2
\end{aligned} \tag{A.85}$$

   thus all the sense monotony in $F1(Q)$ is growing.
2. Decreasing monotony q_2, it has to be obtained from $F(Q)$ all the terms with negative coefficients containing q_2

$$\begin{aligned}
&F_2(Q) = -266.6q_2 - 3.2q_2^3\\
&\cdot - 7.8q_1^2q_2^3\\
&\cdot - 130.5q_1q_2^2\\
&\cdot - 0.1q_1^3q_2^4
\end{aligned} \tag{A.86}$$

   since each coefficient is negative and q_2 is decreasing monotonic, it has to be changed the monotone sense of q_1 of growing sense to decreasing sense, with $dual(q_1)$ in each of the terms that contain q_1 in the function $F_2(Q)$

$$\begin{aligned}
&F_2(Q) = -266.6q_2 - 3.2q_2^3\\
&\cdot - 7.8Dual(q_1)^2q_2^3\\
&\cdot - 130.5Dual(q_1)q_2^2\\
&\cdot - 0.1Dual(q_1)^3q_2^4
\end{aligned} \tag{A.87}$$

   thus all the monotony sense in $F2(Q)$ is decreasing.

3. Evaluating the monotony sense of other terms, these are

$$F_3(Q) = -32.4q_1^2 + 103.1q_2^2 \tag{A.88}$$

For the first term is observed that the coefficient is negative and q_1 is growing monotony so it has to be changed its monotony sense of growing monotone to a decreasing monotony with $Dual(q_1)$ to maintain a monotony sense decreasing in the term. In the case of the second term, the coefficient is positive and q_2 is decreasing monotony so it has to be changed the monotone sense of q_2 from decreasing monotony to growing monotony with $Dual(q_2)$

$$F_3(Q) = -32.4Dual(q_1)^2 + 103.1Dual(q_2)^2 \tag{A.89}$$

4. As last step adding all the results $FR(Q) = F_1(Q) + F_2(Q) + F_3(Q)$ and calculating the optimal extension of the function $FR(QD)$

$$\begin{aligned}
&FR(QD) = 8.8q_1Dual(q_2)^3 + 0.3q_1^4Dual(q_2)^2 + 1.8q_1^3Dual(q_2) \\
&\cdot + 69.5q_1^2Dual(q_2)^2 + 6.6q_1^3Dual(q_2)^2 + 5.9q_1^3Dual(q_2)^3 + \\
&0.1q_1^4Dual(q_2)^3 \\
&\cdot + 0.12q_1^4Dual(q_2)^4 + 96.5q_1^2Dual(q_2) \\
&\cdot + 110.6q_1 + 60.4q_1Dual(q_2) + 0.03Dual(q_2)^4q_1^2 \\
&\cdot - 266.6q_2 - 3.2q_2^3 \\
&\cdot - 7.8Dual(q_1)^2q_2^3 \\
&\cdot - 130.5Dual(q_1)q_2^2 \\
&\cdot - 0.1Dual(q_1)^3q_2^4 \\
&\cdot - 32.4Dual(q_1)^2 + 103.1Dual(q_2)^2
\end{aligned} \tag{A.90}$$

The optimal extension is $FR(QD) = [11.0975, 166.1975]$. This example also has been raised by Vehí **Vehí (1998)**, to verify the positivity of a function with uncertain parameters.

Example A.10.Consider a function $F(q)$ depending on an uncertain parameter vector q belonging to an uncertainty domain Q. To find the semantic extension F^* of the following function

$$\begin{aligned}
&F(Q) = 5 + 12q_3 - 6q_2 - 3q_1q_2 - 8q_2q_1^2q_3 + 3q_1^2q_2^2 \\
&\cdot - 6q_1q_2q_3 + 3q_1q_2^2 - 4q_2(q_4^2) + 4q_2^2q_4^2
\end{aligned} \tag{A.91}$$

The natural extension of the function $fR(Q)$ with uncertain intervals $q_1 = [0, 0.05]$, $q_2 = [0, 0.05]$, $q_3 = [0, 0.05]$, $q_4 = [0, 0.05]$ it is $fR(Q) = [-0.5, 11.81]$.

The range is over bounded and contains zero. Doing a study of monotony is obtained:

$$\frac{dF}{dq_1} = -3q_2 + 6q_1(q_2^2) - 6q_2q_3 = [-3, 1.5] \tag{A.92}$$

$$\begin{aligned}&\frac{dF}{dq_2} = -6 - 3q_1 + 8(q_4^2)q_3 + 6(q_1^2)q_2 - 6q_1q_3 \\ &\cdot + 6q_1q_2 - 4(q_4^2) + 8q_2(q_4^2) = [-11, -2.75]\end{aligned} \tag{A.93}$$

$$\frac{dF}{dq_3} = 12 - 8q_2(q_4^2) - 6q_1q_2 = [9.5, 12] \tag{A.94}$$

$$\frac{dF}{dq_4} = 16q_2q_4q_3 - 8q_2q_4 + 8(q_2^2)q_4 = [-4, 1] \tag{A.95}$$

This study shows that the function $F(Q)$ is decreasing monotone respect to q_2 and growing monotone respect to q_3. Applying the partial coercion theorem for q_2 and q_3, the sub-optimal form of $FR(Q)$ is:

$$\begin{aligned}&FR(QDT^*) = 5 + 12q_3 - 6q_2 - 3q_1q_2 - 8q2(q_4^2)Dual(q3) \\ &\cdot + 3(q_1^2)Dual(q_2)^2 - 6q_1q_2Dual(q_3) + 3q_1Dual(q_2)^2 \\ &\cdot - 4q_2q_4^2 + 4Dual(q_2)^2q_4^2\end{aligned} \tag{A.96}$$

For $q_i = [0, 0.5]$, $Dual(q_2) = [0.5, 0]$ and $Dual(q_3) = [0.5, 0]$ the computed range is $[0.75, 11]$

Example A.11.To determine the $*$-semantic extension of the function

$$g(x, y, z, t) = \frac{x-y}{z-t} \tag{A.97}$$

for $X = [-1, 1]$, $Y = [2, 1]$, $Z = [-1, 1]$ and $T = [3, 2]$. As

$$\begin{aligned}&\frac{dg}{dx} = \frac{1}{z-t} = \frac{1}{[-1,1]-[3,2]} = [-\tfrac{1}{2}, -\tfrac{1}{3}] < 0,\ \textit{is antitonic respect to } x \\ &\frac{dg}{dy} = \frac{-1}{z-t} = \frac{-1}{[-1,1]-[3,2]} = [\tfrac{1}{3}, \tfrac{1}{2}] > 0,\ \textit{is isotonic respect to } y \\ &\frac{dg}{dz} = \frac{y-x}{(z-t)^2} = [\tfrac{1}{9}, \tfrac{1}{2}] > 0,\ \textit{is isotonic respect to } z \\ &\frac{dg}{dt} = \frac{x-y}{(z-t)^2} = [-\tfrac{1}{2}, -\tfrac{1}{9}] < 0,\ \textit{is antitonic respect to } t\end{aligned} \tag{A.98}$$

so

$$g^*(X,Y,Z,T) = [g(x_2,y_1,z_1,t_2), g(x_1,y_2,z_2,t_1)] = [\frac{1-2}{-1-2}, \frac{-1-1}{1-3}]$$
$$\cdot = [\frac{-1}{-3}, \frac{-2}{-2}] = [\frac{1}{3}, 1] \quad \text{(A.99)}$$

in this example theorem A.6 was applied. In $g(x_2,y_1,z_1,t_2)$, $\overline{x}$ and $\overline{t}$ were used since they are anti tonics and $\underline{y}$ and $\underline{x}$ since they are isotonics and in $g(x_1,y_2,z_2,t_1)$ were used $\underline{x}$, $\underline{t}$, $\overline{y}$ and $\overline{z}$.

Example A.12.To determine the $**$- and $*$- semantic extensions of the function

$$f(x,y) = xy + \frac{1}{x+y} \quad \text{(A.100)}$$

for $X = [5,10]$ and $Y = [2,1]$. If the variables x, y in the function $f(x,y)$ are taken as independent variables, the function $f(x,y)$ can be transformed into:

$$f(x,y) = x_1 y_1 + \frac{1}{x_2+y_2} \quad \text{(A.101)}$$

as

$$\frac{df}{dx_1} = y_1 > 0, \ \textit{is isotonic respect to } x_1$$
$$\frac{df}{dy_1} = x_1 > 0, \ \textit{is isotonic respect to } y_1$$
$$\frac{df}{dx_2} = \frac{-1}{(x_2+y_2)^2} < 0, \ \textit{is antitonic respect to } x_2$$
$$\frac{df}{dy_2} = \frac{-1}{(x_2+y_2)^2} < 0, \ \textit{is antitonic respect to } y_2 \quad \text{(A.102)}$$

so

$$XD = (X, Y, Dual(X), Dual(Y)) \Rightarrow fR(XD)$$
$$= X * Y + \frac{1}{Dual(X)+Dual(Y)}$$
$$fR(XD) = [5,10] * [2,1] + \frac{1}{[10,5]+[1,2]} = [10,10] + \frac{1}{[11,7]}$$
$$= [71/7, 111/11] \quad \text{(A.103)}$$

On the other hand as X is proper and Y is improper, the $**$ and $*$-semantic extensions are

$$f^*(X) := \vee(x_p, X_p^{'}) \wedge (x_i, X_i^{'})[f(x_p,x_i), f(x_p,x_i)]$$
$$= [min(x_p, X_p^{'})max(x_i, X_i^{'})f(x_p,x_i), max(x_p, X_p^{'})min(x_i, X_i^{'})f(x_p,x_i)],$$

$$f^{**}(X) := \wedge(x_i, X_i^{'}) \vee (x_p, X_p^{'})[f(x_p,x_i), f(x_p,x_i)]$$
$$= [max(x_i, X_i^{'})min(x_p, X_p^{'})f(x_p,x_i), min(x_i, X_i^{'})max(x_p, X_p^{'})f(x_p,x_i)]. \quad \text{(A.104)}$$

thus

$$\begin{aligned}
&f^*(X,Y) := [min(x,[5,10]^{'})max(y,[1,2]^{'})(xy+\tfrac{1}{x+y}),\\
&\cdot max(x,[5,10]^{'})min(y,[1,2]^{'})(xy+\tfrac{1}{x+y})]\\
&\cdot = [min(x,[5,10]^{'})(2x+\tfrac{1}{x+2}), max(x,[5,10]^{'})(x+\tfrac{1}{x+1})]\\
&\cdot = [71/7, 111/11]
\end{aligned} \tag{A.105}$$

$$\begin{aligned}
&f^{**}(X,Y) := [max(y,[1,2]^{'})min(x,[5,10]^{'})(xy+\tfrac{1}{x+y}),\\
&\cdot min(y,[1,2]^{'})max(x,[5,10]^{'})(xy+\tfrac{1}{x+y})]\\
&\cdot = [max(y,[1,2]^{'})(5y+\tfrac{1}{5+y}), min(y,[1,2]^{'})(10y+\tfrac{1}{10+y})]\\
&\cdot = [71/7, 111/11]
\end{aligned} \tag{A.106}$$

Example A.13.Application of the algorithm f^*

Apply the algorithm f^* step by step to find the extension of the following function:

$$f(u,v,z) := u^2 + v^2 + uv - 20u - 20v + 100 - 10sin(z) \tag{A.107}$$

where the variables u, v, z have the following intervals $u = [0,6], v = [8,2]$ and $z = [9,-4]$.

1. Initialization: Create $Cell([0,6],[8,2],[9,-4])$
   We initialize the cell with the proper u and improper v components as it is illustrated in Figure A.6.

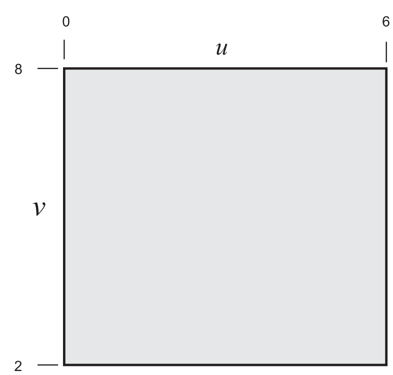


Fig. A.6. Proper $u = [0,6]$ and improper $v = [8,2]$ components

2. To divide the proper and improper components in Strips and Cells.
   We are going to suppose that we divided the cell u, v in a Strip set $StripSet = \{Strip_1, Strip_2\}$ and that each Strip contains two cells that is to say $Strip_1 = \{Cell_{11}, Cell_{12}\}$ and $Strip_2 = \{Cell_{21}, Cell_{22}\}$. A representation is depicted in Figure A.7

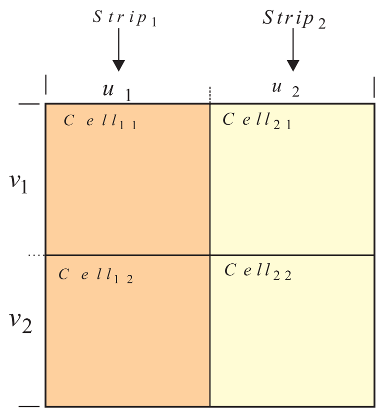


Fig. A.7. Strips and Cells

3. **Compute inner and outer approximations of Cells**
   In Figure A.8 we indicated that we will realize the inner and outer approximations of each cell

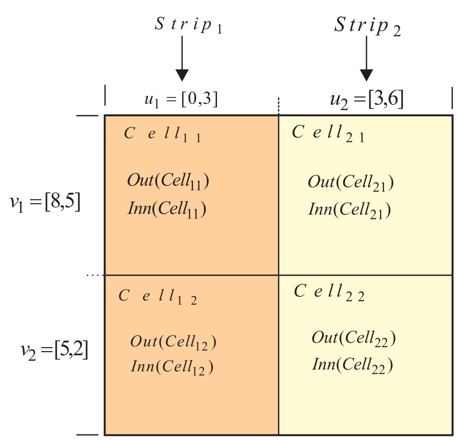


Fig. A.8. Strips and Cells

The function can be written as:

$$f(u,v,z) := u_1^2 + v_1^2 + u_2 v_2 - 20u_3 - 20v_3 + 100 - 10sin(z) \tag{A.108}$$

where the sub indexes represent the different multi-incidences. First of all, the monotony of f with respect to each variable and with respect to each one of its incidences, considered as different variables, has to be computed.

$$\begin{aligned}
&df(u,v,z)/du := 2u + v - 20 \in 2 * [0,3] + [5,8] - [20,20] = [-15,-6] \leq 0\\
&df(u,v,z)/du_1 := 2 * u \in 2 * [0,3] = [0,6] \geq 0\\
&df(u,v,z)/du_2 := v \in [5,8] = [5,8] \geq 0\\
&df(u,v,z)/du_3 := -20 \in [-20,-20] = [-20,-20] \leq 0\\
&df(u,v,z)/dv := 2v + u - 20 \in 2 * [5,8] + [0,3] - [20,20] = [-10,-1] \leq 0\\
&df(u,v,z)/dv_1 := 2v \in 2 * [5,8] = [10,16] \geq 0\\
&df(u,v,z)/dv_2 := u \in [0,3] = [0,3] \geq 0\\
&df(u,v,z)/dv_3 := -20 \in [-20,-20] = [-20,-20] \leq 0\\
&df(u,v,z)/dz := -10cos(z) \in 10 * cos([-4,9]) = [-10,10] \supseteq 0
\end{aligned} \tag{A.109}$$

since $df(u,v,z)/du \leq 0$, $df(u,v,z)/du_1 \geq 0$, $df(u,v,z)/du_2 \geq 0$ and $df(u,v,z)/du_3 \leq 0$ we must change to the sense of the monotony of the variable u_1 and u_2 to maintain a same sense of the monotony of u. In the same way, we observed that $df(u,v,z)/dv \leq 0$, $df(u,v,z)/dv_1 \geq 0$, $df(u,v,z)/dv_2 \geq 0$ and $df(u,v,z)/dv_3 \leq 0$ we must change the sense of the monotony of the variable v_1 and v_2 to maintain a same sense of monotony of v. As the variable z is not multi-incident is considered in the internal and external approximations of the cells. The function is expressed as it is indicated

$$\begin{aligned}
&f(u,v,z) := Dual(u)_1^2 + Dual(v)_1^2 + Dual(u)_2 Dual(v)_2 -\\
&20u_3 - 20v_3 + 100 - 10sin(z)
\end{aligned} \tag{A.110}$$

With the following main program, the arithmetical operations and the calculation of the external range were realized. Obtaining the following result $Out(Cell_{11}) = [-1,-6]$.

```
int main(int argc, char* argv[])
{
secureFpu();
ivalDb u;
ivalDb v;
ivalDb f;
```

```
ivalDb z;
ivalDb dfu;
ivalDb dfu1;
ivalDb dfu2;
ivalDb dfu3;
ivalDb dfv;
ivalDb dfv1;
ivalDb dfv2;
ivalDb dfv3;
ivalDb dfz;

u=ivalDb(0,3);
v=ivalDb(8,5);
z=ivalDb(9,-4);

dfu=2*u+Du(v)-20;
dfu1=2*u;
dfu2=Du(v);
dfu3=-20;

dfv=2*Du(v)+u-20;
dfv1=2*Du(v);
dfv2=u;
dfv3=-20;

dfz=-10*cos(Du(z));

f=((Du(u))^2)+((Du(v))^2)+(Du(u))*(Du(v))-
20*u-20*v+100-10*sin(z);
cout<<f;
return 0;
}
```

As the function is totally monotonic with respect to $u = [0, 3]$ and $v = [8, 5]$ then in order to realize the internal calculation of the cell, we will take the center of $u = \check{u} = [1.5, 1.5]$ and will apply the dual operation to the variables that corresponds to v.

$$\begin{aligned} &Inn(cell_{11}) = InnR(\check{u}, Dual(v)) \\ &f(u,v,z) := \check{u}_1^2 + Dual(v)_1^2 + \check{u}_2 Dual(v)_2 - 20\check{u}_3 - \\ &20v_3 + 100 - 10sin(z) \end{aligned} \tag{A.111}$$

The result is $Inn(Cell_{11}) = [14.75, -21.75]$.

Now we are going to calculate $Out(Cell_{12})$ and $Inn(Cell_{12})$. Following the same procedure, we verified the sense of monotony of each one of the variables in the intervals $u = [0,3]$ and $v = [5,2]$.

$$\begin{array}{l} df(u,v,z)/du := 2u + v - 20 \in 2*[0,3] + [2,5] - [20,20] = [-18,-9] \leq 0 \\ df(u,v,z)/du_1 := 2*u \in 2*[0,3] = [0,6] \geq 0 \\ df(u,v,z)/du_2 := v \in [2,5] = [2,5] \geq 0 \\ df(u,v,z)/du_3 := -20 \in [-20,-20] = [-20,-20] \leq 0 \\ df(u,v,z)/dv := 2v + u - 20 \in 2*[2,5] + [0,3] - [20,20] = [-16,-7] \leq 0 \\ df(u,v,z)/dv_1 := 2v \in 2*[2,5] = [4,10] \geq 0 \\ df(u,v,z)/dv_2 := u \in [0,3] = [0,3] \geq 0 \\ df(u,v,z)/dv_3 := -20 \in [-20,-20] = [-20,-20] \leq 0 \\ df(u,v,z)/dz := -10cos(z) \in 10*cos([-4,9]) = [-10,10] \supseteq 0 \end{array} \tag{A.112}$$

We observe that the function is totally monotonic, because of this reason we make that $u_1 = Dual(u_1)$, $u_2 = Dual(u_2)$, $v_1 = Dual(v_1)$ and $v_2 = dual(v_2)$

$$\begin{array}{l} f(u,v,z) := Dual(u)_1^2 + Dual(v)_1^2 + Dual(u)_2 Dual(v)_2 - \\ 20u_3 - 20v_3 + 100 - 10sin(z) \end{array} \tag{A.113}$$

obtaining the following result $Out(Cell_{12}) = [29,15]$.
As the function is totally monotonic with respect to $u = [0,3]$ and $v = [5,2]$ then, in order to realize the internal calculation of the cell, we will take the center of $u = \check{u} = [1.5,1.5]$ and will apply the dual operation to the variables that corresponds to v.

$$\begin{array}{l} Inn(cell_{12}) = InnR(\check{u}, Dual(v)) \\ f(u,v,z) := \check{u}_1^2 + Dual(v)_1^2 + \check{u}_2 Dual(v)_2 - 20\check{u}_3 - \\ 20v_3 + 100 - 10sin(z) \end{array} \tag{A.114}$$

The result it is $Inn(Cell_{12}) = [49.25,-5.25]$.
For the calculation of $Out(Cell_{21})$ and $Inn(Cell_{21})$ in intervals $u = [3,6]$ and $v = [8,5]$ we realize the same procedure.

$$\begin{array}{l} df(u,v,z)/du := 2u + v - 20 \in 2*[3,6] + [5,8] - [20,20] = [-9,0] \leq 0 \\ df(u,v,z)/du_1 := 2*u \in 2*[3,6] = [6,12] \geq 0 \\ df(u,v,z)/du_2 := v \in [5,8] = [5,8] \geq 0 \\ df(u,v,z)/du_3 := -20 \in [-20,-20] = [-20,-20] \leq 0 \\ df(u,v,z)/dv := 2v + u - 20 \in 2*[5,8] + [3,6] - [20,20] = [-7,2] \supseteq 0 \\ df(u,v,z)/dv_1 := 2v \in 2*[5,8] = [10,16] \geq 0 \\ df(u,v,z)/dv_2 := u \in [3,6] = [3,6] \geq 0 \\ df(u,v,z)/dv_3 := -20 \in [-20,-20] = [-20,-20] \leq 0 \\ df(u,v,z)/dz := -10cos(z) \in 10*cos([-4,9]) = [-10,10] \supseteq 0 \end{array}$$

(A.115)

We observe that the function is monotonic with respect to u but not with respect to v. Reason why we will use the following expression to realize the external calculation of the cell

$$\begin{aligned}&Out(Cell_{21}) = OutR(fR(Dual(u), \check{v}))\\&fR(u,v,z) := Dual(u)_1^2 + \check{v}_1^2 + Dual(u)_2\check{v}_2 - 20u_3 - 20\check{v}_3+\\&100 - 10sin(z)\end{aligned} \tag{A.116}$$

being $\check{v} = [6.5, 6.5]$. The result it is $Out(Cell_{21}) = [-22.75, -29.25]$. Now for the internal calculation of the cell we used the following expression

$$\begin{aligned}&Inn(Cell_{21}) = InnR(fR(\check{u}, v)\\&fR(u,v,z) := \check{u}_1^2 + v_1^2 + \check{u}_2v_2 - 20\check{u}_3 - 20v_3+\\&100 - 10sin(z)\end{aligned} \tag{A.117}$$

being $\check{u} = [4.5, 4.5]$. To verify that in the previous expression we cannot use $Dual(v)$ since the function is not monotonic respect to v. Therefore $Inn(Cell_{21}) = [40.25, -92.25]$.

In order to conclude with this step, we will calculate to outer and inner approximations of $Cell_{22}$ with $u = [3, 6]$ and $v = [5, 2]$. For this, we again verified the sense of monotony of each one of the variables

$$\begin{aligned}&df(u,v,z)/du := 2u + v - 20 \in 2*[3,6] + [2,5] - [20,20] = [-12, 13] \leq 0\\&df(u,v,z)/du_1 := 2*u \in 2*[3,6] = [6,12] \geq 0\\&df(u,v,z)/du_2 := v \in [2,5] = [2,5] \geq 0\\&df(u,v,z)/du_3 := -20 \in [-20,-20] = [-20,-20] \leq 0\\&df(u,v,z)/dv := 2v + u - 20 \in 2*[2,5] + [3,6] - [20,20] = [-13,-4] \leq 0\\&df(u,v,z)/dv_1 := 2v \in 2*[2,5] = [4,10] \geq 0\\&df(u,v,z)/dv_2 := u \in [3,6] = [3,6] \geq 0\\&df(u,v,z)/dv_3 := -20 \in [-20,-20] = [-20,-20] \leq 0\\&df(u,v,z)/dz := -10cos(z) \in 10*cos([-4,9]) = [-10,10] \supseteq 0\end{aligned} \tag{A.118}$$

since the function is totally monotonic with respect to u and v then we express the external calculation of the cell on the following form

$$\begin{aligned}&Out(Cell_{22}) = OutR(Dual(u), Dual(v))\\&f(u,v,z) := Dual(u)_1^2 + Dual(v)_1^2 + Dual(u)_2Dual(v)_2-\\&20u_3 - 20v_3 + 100 - 10sin(z)\end{aligned} \tag{A.119}$$

obtaining the following result $Out(Cell_{22}) = [2, -21]$. For the internal calculation we considered a point of u and $dual(v)$ since the function is monotonic with respect to v. The expression is the following one

$$\begin{aligned} &Inn(cell_{21}) = InnR(\check{u}, Dual(v)) \\ &f(u,v,z) := \check{u}_1^2 + Dual(v)_1^2 + \check{u}_2 Dual(v)_2 - 20\check{u}_3 - \\ &20v_3 + 100 - 10sin(z) \end{aligned} \tag{A.120}$$

The result is $Inn(Cell_{22}) = [13.25, -32.25]$.

4. **Compute inner and outer approximations of Strips**
   In this step we will calculate the internal and external approximations of Strips. We will use the following expressions

$$\begin{aligned} &Out(Strip_1) = Meet(Out(cell_{11}), Out(cell_{12})) \\ &Inn(Strip_1) = Meet(Inn(cell_{11}), Inn(cell_{12})) \\ &Out(Strip_2) = Meet(Out(cell_{21}), Out(cell_{22})) \\ &Inn(Strip_2) = Meet(Inn(cell_{21}), Inn(cell_{22})) \end{aligned} \tag{A.121}$$

   The results are

$$\begin{aligned} &Out(Strip_1) = Meet([-1, -6], [29, 15]) = [29, -6] \\ &Inn(Strip_1) = Meet([14.75, -21.75], [49.25, -5.25]) = [49.25, -21.75] \\ &Out(Strip_2) = Meet([-22.75, -29.25], [2, -21]) = [2, -29.25] \\ &Inn(Strip_2) = Meet([40.25, -92.25], [13.25, -32.25]) = [40.25, -92.25] \end{aligned} \tag{A.122}$$

   In Figure A.9 we indicated the internal and external calculations of strips. With those results we can realize the Inner calculation and Outer using a Join. That is to say

$$\begin{aligned} &Outer = Join(Out(strip_1), Out(strip_2)) \\ &Inner = Join(Inn(strip_1), Inn(strip_2)) \end{aligned} \tag{A.123}$$

   The results are

$$\begin{aligned} &Outer = Join([29, -6], [2, -29.25]) = [2, -6] \\ &Inner = Join([49.25, -21.75], [40.25, -21.75]) = [40.25, -21.75] \end{aligned} \tag{A.124}$$

   Finally in Figure A.10 we indicate that we have obtained the external and internal calculation of the function for a cell with proper and improper components.

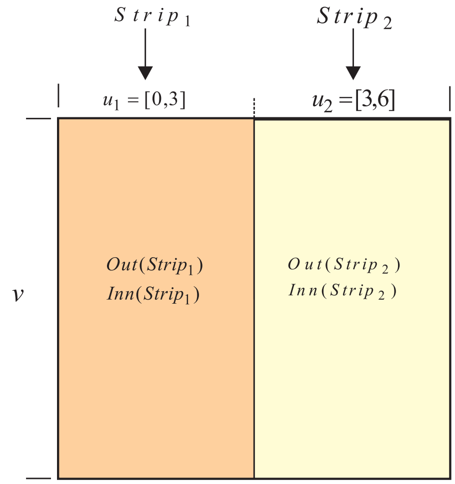


Fig. A.9. Out and Inn of Strips

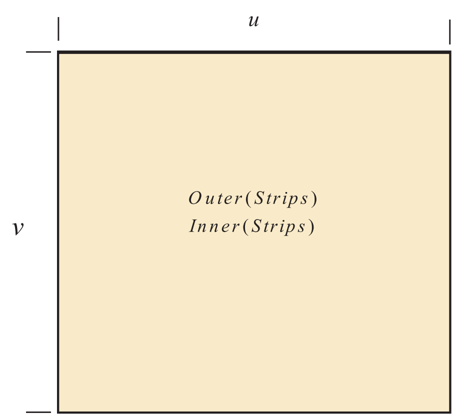


Fig. A.10. Outer and Inner of Strips

B Exact linearization controller design

B.1 SISO systems input-output linearization

Theoretical foundations of input-output linearization can be consulted in the Isidori's book **Isidori (1985)**. For simplicity, it is considered affine SISO control system:

$$\begin{aligned} \frac{dx}{dt} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{B.1}$$

where $u \in \Re$ is the control input, $x \in \Re^n$ is the state vector, $y \in \Re$ is the controlled output, $f(x)$ and $g(x)$ are n-dimensional smooth functions on $\Re^n$, $h(x)$ is a smooth function on $\Re^n$.

The nonlinear system is said to have a relative degree of r at the point x_0 if for all x in a neighbourhood of x_0

$$\begin{aligned} &L_g L_f^i h(x) = 0, \ \forall \ 0 \leq i < r-1 \\ &L_g L_f^{r-1} h(x) \neq 0 \end{aligned} \tag{B.2}$$

where $L_f h$ is the Lie derivative, i.e. $L_f h = \frac{dh}{dx} f(x)$. If a nonlinear system has a finite relative degree, it can always construct a nonlinear state coordinate transformation $\eta = \phi(x)$ such that

$$\begin{aligned} &\phi_i(x) = L_f^{i-1} h(x), \ 1 \leq i \leq r \\ &L_g \phi_i(x) = 0, \ r+1 \leq i \leq n \end{aligned} \tag{B.3}$$

This transforms the nonlinear systems into the normal form:

$$\begin{aligned} \frac{d\eta_i}{dt} &= \eta_{i+1}, \ 1 \leq i \leq r-1 \\ \frac{d\eta_r}{dt} &= \alpha(\eta) + \beta(\eta)u = L_f^r h(x) + L_g L_f^{r-1} h(x)u, \end{aligned} \tag{B.4}$$

$$\begin{aligned} \frac{d\eta_i}{dt} &= \gamma(\eta), \ r+1 \leq i \leq n \\ y &= \eta_1 \end{aligned} \tag{B.5}$$

It is obvious that the nonlinear state feedback control law

$$u = \frac{1}{L_g L_f^{r-1} h}(v - \sum_{i=0}^{r-1} k_{i+1} L_f^i h - L_f^r h) \tag{B.6}$$

will make the system linear from v to y, i.e.

$$y^{(r)} + \sum_{i=0}^{r-1} k_{i+1} y^{(i)} = v \tag{B.7}$$

Then, a linear controller can be designed for the linearized system. If the objetive is to track a set point y_{sp}, one simple way is to let

$$v = k_0 \int (y_{sp} - y) d\tau \tag{B.8}$$

this control linearizing law makes the last $(n-r)$ state variables of η unobservable from the output. Internal stability requires those unobservable modes to be stable. To be precise, it needs the concept of zero dynamics. which is a generalization of the concept of zeros to nonlinear systems. Let it partion the state vector as

$$\varsigma = [\eta_1 ... \eta_r]', \; z = [\eta_{r+1} ... \eta_n]' \tag{B.9}$$

Then eq. B.5 can be rewritten as

$$\frac{dz}{dt} = \gamma(\varsigma, z) \tag{B.10}$$

Zero dynamics of a nonlinear system is defined as

$$\frac{dz}{dt} = \gamma(0, z) \tag{B.11}$$

This is equivalent to the dynamics with the output $y(t)$ constrained identically to zero. Exact input-output linearization is, in fact, a nonlinear analog of placing poles at plant zeros, hence cancels the zero dynamics and leads to z unobservable. It is obvious now that the zero dynamics must be stable to guarantee internal stability.

The applicability of exact input-output linearization depends on the existence of relative degree on the stability of zero dynamics. However, both relative degree and stability of zero dynamics are local properties of a nonlinear systems. This local nature greatly complicates the applicability problem. It is no longer so simple as whether or not applicate to a system. Zero dynamics of a nonlinear system may be stable in some operating regions but unstable in others. Similarly, a nonlinear system may have singular points where the relative degree cannot be defined. So applicability only applies to specific operating regions of a nonlinear system.

B.1.1 Application to control of fed-batch bioreactors

Using λ as the control variable and x_1 as output variable. In this case, we have

$$f(x) = \begin{bmatrix} \mu(x_2)x_1 \\ -\frac{\mu(x_2)x_1}{y_{xs}} \\ 0 \end{bmatrix}, \; g(x) = \begin{bmatrix} -x_1^2 \\ (S_f - x_2)x_1 \\ x_1 x_3 \end{bmatrix} \tag{B.12}$$

Since λ appears on the right side of each of the state equations, the relative degree is 1 without mattering if x_1 or $\mu(x_2)$ is the controlled variable.

$$\begin{array}{l} y_1 = h(x) = x_1 \\ L_f h(x) = \mu(x_2)x_1 \\ L_f^0 h = h(x) = x_1 \\ L_g h(x) = -x_1^2 \end{array} \tag{B.13}$$

The control law

$$\lambda = -\tfrac{1}{x_1^2}(k_0 \int (x_{1p} - x_1)d\tau - k_1 x_1 - \mu(x_2)x_1) \tag{B.14}$$

will make the closed-loop system input-output linear with a transfer function

$$T(s) = \tfrac{k_0}{s^2 + k_1 s + k_0} \tag{B.15}$$

Singular point $L_g h(x) = -x_1^2 = 0$ means wash-out, so it is not possible to have singular point under a normal operation condition.

If $y_1 = x_2$ is taken as output variable, it has

$$\begin{array}{l} y_1 = h(x) = x_2 \\ L_f h(x) = -\frac{\mu(x_2)x_1}{y_{xs}} \\ L_f^0 h = h(x) = x_2 \\ L_g h(x) = (S_f - x_2)x_1 \end{array} \tag{B.16}$$

The control law

$$\lambda = \tfrac{1}{x_1(S_f - x_2)}(k_0 \int (x_{2p} - x_2)d\tau - k_1 x_2 + \tfrac{\mu(x_2)x_1}{yxs}) \tag{B.17}$$

If $y_1 = \mu(x_2)$ is taken as output variable, it has

$$\begin{aligned} y_1 &= h(x) = \mu(x_2) \\ L_f h(x) &= -\frac{\dot{\mu}(x_2)\mu(x_2)x_1}{y_{xs}} \\ L_f^0 h &= h(x) = \mu(x_2) \\ L_g h(x) &= (S_f - x_2)x_1\dot{\mu}(x_2) \end{aligned} \tag{B.18}$$

The control law

$$\lambda = \frac{1}{\dot{\mu}(x_2)x_1(S_f - x_2)}(k_0 \int (\mu(x_{2p}) - \mu(x_2))d\tau - k_1\mu(x_2) - \frac{\dot{\mu}(x_2)\mu(x_2)x_1}{yxs}) \tag{B.19}$$

If the output variable is $y = [x_1\ \mu(x_2)]^T$ and the input control is $u = [\lambda\ S_f]^T$, it has

$$\begin{aligned} L_f h(x) &= \begin{bmatrix} \mu(x_2)x_1 \\ -\frac{\dot{\mu}(x_2)\mu(x_2)x_1}{yxs} \end{bmatrix} \\ L_f^0 h = h(x) &= \begin{bmatrix} x_1 \\ \mu(x_2) \end{bmatrix} \\ L_g h(x) &= \begin{bmatrix} -x_1^2 & 0 \\ -\dot{\mu}(x_2)x_2x_1 & \dot{\mu}(x_2)x_1 \end{bmatrix} \end{aligned} \tag{B.20}$$

The control law

$$u(x) = (L_g h)^{-1}(\begin{bmatrix} k_{011}\int (x_{1p} - x_1)d\tau \\ k_{022}\int (\mu(x_{2p}) - \mu(x_2))d\tau \end{bmatrix} - \begin{bmatrix} k_{111}x_1 \\ k_{122}\mu(x_2) \end{bmatrix} - \tag{B.21}$$

$$\begin{bmatrix} \mu(x_2)x_1 \\ -\frac{\dot{\mu}(x_2)\mu(x_2)x_1}{y_{xs}} \end{bmatrix}) \tag{B.22}$$

will make the closed-loop system input-output linear with a transfer function matrix

$$T(s) = (s^2 I + sK_1 + K_0) \tag{B.23}$$

if diagonal $K_0 = diag\{k_{011}, k_{022}\}$ and $K_1 = diag\{k_{111}, k_{122}\}$ is selected, it will also have input-output decoupling, i.e

$$T(s) = diag\{\frac{k_{011}}{s^2 + k_{111}s + k_{011}}, \frac{k_{022}}{s^2 + k_{122}s + k_{022}}\} \tag{B.24}$$

References

1. ANDÚJAR J. M., BRAVO J.M., AND PEREGRÍN A. (2004). *Stability analysis and synthesis of multivariable fuzzy systems using interval arithmetic*, Fuzzy Sets and Systems, Vol. 148, pp. 337-353.
2. AGRAWAL R., KOSHY G., AND RAMSEIER M. (1989). *An algorithm for operating a fed-batch fermenter at optimum specific-growth rate*, Biotechnology and Bioengineering, Vol. 33, pp. 115-125.
3. ANTRITTER F., KLETTING M., AND HOFER P. (2007). *Robust analysis of flatness based control using interval methods*, International Journal of Control, Vol. 80, No. 5, pp. 816-823.
4. ARMENGOL J. (1999). *Application of modal interval analysis to the simulation of the behaviour of synamic systems with uncertain parameters*, PhD thesis, Universidad de Girona. Spain.
5. BENHAMOU, F. AND GOULARD, F. (2000). *Universally Quantified Interval Constraints*, Lecture Notes in Computer Science, Vol. 5(4).
6. BONDIA, J., (2002). *Sistemas con incertidumbre paramétrica borrosa: Análisis y diseño de controladores* Tésis Doctoral, Universidad Politécnica de Valencia, Departamento de Ingeniería de Sistemas y Automática.
7. BONDIA, J., KIEFFER, E., WALTER, J., MONREAL, J., AND PICO J., (2004). *Guaranteed tunning of PID controllers for parametric uncertain systems.* 43th. IEEE Conf. on Decision and Control, Bahamas.
8. BONDIA, J., SALA, A., AND PICO J., (2005). *Possibilistic robust control for fuzzy plants: Controlling performance degradation.* 16^{th} IFAC Word Congress, Praha.
9. BONDIA, J., SALA, A., PICO J., AND SAINZ A. (2006). *Controller design under fuzzy pole-placement specifications. An interval arithmetic approach.* IEEE Transactions on fuzzy systems. Vol. 14, No. 6, pp. 822-836.
10. CAZAURANG, F.,(1997). *Commande robuste des systèmes plats. Application à la commande d'une machine synchrone.* Thèse de doctorat, Université de Boxdeaux I (France).
11. CARTAN, E.,(1953). *Sur l'equivalence absolute de certains systemes d'equations differentielles et sur certaines families de courbes*, Euvres Comletes, Vol II, Gauthier-Villars, pp. 1133-1168.
12. CHARLET, B., LEVINE, J., AND MARINO, R., (1989). *On dynamic feedback linearization*, Systems Control Lett., Vol. 13., pp 143-151.
13. CHARLET, B., LEVINE, J., AND MARINO, R., (1991). *Sufficient conditions for dynamic state feedback linearization*, SIAM J. Control and Optimization, Vol. 29., pp 38-57.
14. CHELOUACH, A., DELALEAU, E., MARTIN, P., AND ROUCHON, P., (1996). *Differential flatness and control of induction motors.* In P. Borne, M. Staroswiecki,

J.P. Cassar and S. El Khattabi (eds) Proc. Sumposium on Control, Optimization and Supervision; Computacional Engineering in Systems Applications IMACS Multiconference, Lille, France, pp. 80-85.

15. Chelouah, A., (1997). *Extension of differential flat fields and liouvillian systems* In Proceedings of the 36th Conference On Decision and control, San Diego, California USA., December 1997, pp. 4268-4273.
16. Dubois D., Prade H., and Ughetto L. (2003). *A New Perspective on Reasoning with Fuzzy Rules*, International Journal Of Intelligent Systems, Vol 18, pp. 541-567.
17. Faiz, N., Agrawal, S. K., and Murray, R.M. (2001). *Trajectory Planning of Differentially Flat Systems with Dynamics and Inequalities*, Journal of guidance, control, and dynamics Vol.24, pp. 219-227.
18. Fliess, M., Lévine, J., and Rouchon, P. (1993). *Generalized state variable representation for a simplified crane description*, Int. J. Control, Vol 58(2), pp. 277-283.
19. Fliess, M., Lévine, J., Martin, Ph., and Rouchon, P. (1995a). *Design of trajectory stabilization feedback for drifless flat systems*, in Proc. 3rd European Control Conference, 1995, pp. 1882-1887. Rome.
20. Fliess, M., Lévine, J., Martin, Ph., and Rouchon, P. (1995b). *Flatness and defect of nonlinear systems: Introductory theory and examples*, International Journal of Control, Vol. 61, pp. 1327-1361.
21. Fliess, M., Lévine, J., Martin, Ph., Olliver, F., and Rouchen, P. (1995c). *Flatness dynamic feedback linearizability: two approaches*, Proc. European Control Conf.
22. Fliess, M., Lévine, J., Martin, Ph., and Rouchon, P. (1999). *A Lie-Backlund approach to equivalence and flatness of nonlinear systems*, IEEE Transactions on Automatic Control, Vol. 44, pp. 922-937.
23. Gardeñes E., Sainz M.Á., Jorba L., Calm R., Estela R., Mielgo H., and Trepat A. (2001). *Modal intervals.* Realible Computing, Vol. 7, pp. 77-111.
24. Gardeñes E., Trepat A., and Janer J.M. (1980). *SIGLA-PL1/(0). Development and applications in: Nickel K.*, (ed.), Interval Mathematics, Academic Press, New York, pp. 301-315.
25. Gardeñes E., Mielgo H., and Trepat A. (1985). *Modal intervals: reasons and ground semantics*, Lecture notes in computer science, Vol. 212, pp.27-35.
26. Gardeñes E., Mielgo H., and Sainz M.A. (1995). *Presentation of the research group sigla/x*, Technical report, Universitat de Girona. Catalonia, Spain.
27. Garloff J., and Graf B. (1999). *Solving strict polynomial inequalities by Bernstein expansions*, IEE London, Chaper 14, pp. 339-351.
28. Gent I.P., Nightingale P., Rowley A., and Stergiou K. (2008). *Solving quantified constraint satisfaction problems*, Artificial Intelligence Vol. 172, pp.738-771.
29. Goldsztejn A., and Chabert G. (2006). *On the approximation of linear* $\forall\exists$-*solution sets*, SCAN-12th International Symposium of Scientific Computing.
30. Goldsztejn A., Michel C., and Rueher M. (2009). *Efficient handling of universally quantified enequalities*, Constraints, Vol. 14, pp. 117-135.
31. Hagenmeyer, V., and Delaleau, E.(2003a). *Exact feedforward linearisation based on differential flatness: The SISO case. In A. Zinober and D. Owens (Eds), Nonlinear and Adaptive Control (NCN4 2001), Lecture notes in Control and Information Sciences*, Vol. 281. Berlin, Heidelberg: Springer, pp. 161-170.

32. Hagenmeyer, V., and Delaleau, E. (2003b). *Exact feedforward linearization based on differential flatness*, International Journal of Control, 76 pp. 537-556.
33. Hagenmeyer, V., and Delaleau, E. (2003c). *Robustness analysis of exact feedforward linearization based on differential flatness*, Automatica, 39 pp. 1941-1946.
34. Hansen, E. (2004). *Global optimization using interval analysis second edition, revised and expanded*, Marcel Dekker, Inc. and Sun Microsystems, Inc.
35. Herrero P., Sainz M. A., and Vehí J. (2005). *Quantified Set Inversion Algorithm with Applications to Control*, Realiable Computing, Vol. 11, pp. 369-382.
36. Herrero P. (2006). *Quantified real constraint solving using modal intervals with applications to control*, Doctoral Thesis. University of Girona (Spain).
37. Hertling P.(2001). *A limitation for underestimation via Twin Arithmetic*, Realiable Computing, Vol. 7, pp. 157-169.
38. Isidori A.(1985). *Nonlinear Control Systems*, Springer-Verlag Berlin.
39. Jaulin L., Walter E. (2004). *Set inversion via interval analysis for nonlinear bounded-error estimation*, Automatica, Vol. 32(8), pp. 1053-1064.
40. Kreinovich V., Nesterov V.M., and Zheludeva N.A., (1996). *Interval Methods That Are Guaranteed to Underestimate (and the Resulting New Justification of Kaucher Arithmetic)*, Realiable Computing, Vol. 2(2), pp. 119-124.
41. Lévine, J., Lottin J., and Ponsart J.C. (1996). *A nonlinear approach to the control of magnetic bearings*, IEEE Trans. Control Systems Technol., Vol (5), pp. 524-544.
42. Lévine, J., (1999). *Are there new industrial perspectives in the control of mechanical systems?.* In P. Frank (Ed) Advances in Control (Highlights of ECC'99) (London:Springer-Verlag), pp. 197-226.
43. Malan S., Milanese M. and Taragna M. (1997). *Robust analysis and design of control systems using interval arithmetic*, Automática, Vol. 33(7), pp. 1363-1372.
44. Marquez H. J. (2003). *Nonlinear Control Systems Analysis and Design*, Wiley-InterScience.
45. Martin, Ph., (1993). *Endogenous feedbacks and equivalence*, Mathematical Theory of Networks and Systems, Regensburg Germany.
46. Martin, P., Devasia S., and Paden, B. (1994). *A different look at output tracking- Control of a VTOL aircraft*, Automatica, Vol 32(1), pp. 101-107.
47. Martin, P., (1996a). *Aircraft control using flatness*, in Multiconference on Computational in Systems Applications, pp. 194-199.
48. Martin, P., and Rouchon P., (1996b). *Flatness and sampling control of induction motors*, in Workshop IFAC Sym.
49. Martin, P., Murray, R., and Rouchon, P., (1997). *Flat systems*, Plenary Lectures and Minicourses of the European Control Conference, Chapter 8, pp. 211-264, G. Bastin and M. Gevers (Eds.)
50. Martin, P., Murray, R., and Rouchon, P., (2000). *Flat systems, equivalence and feedback.* In A. Baños, F. Lamnabhi-Lagarrigue and F. Montoya (Eds) Advances in the Control of Nonlinear Systems, Vol. 264 of Lecture Notes Control Information Science (London:Springer), pp. 5-32.
51. Moore R. E. (1966). *Interval Analysis. Prentice Hall*, New Jersey (USA).
52. Moore R. E. (1962). *Interval arithmetic and automatic error analysis and digital computing*, PhD thesis, Stanford University.

53. Moore R. E. (1979). *Methods and applications of interval analysis*, SIAM studies in applied mathematics, Philadelphia, Siam.
54. Faiz N., and Agrawal K.(1979). *Trajectory planning of differentially flat systems with dynamics and inequalities*, Jornal of Guidance, Control, and Dynamics, Vol. 24, No. 2, March-April, pp. 219-227.
55. Nijmeijer, H., and Van der Shaft A. (1990). *Nonlinear Dynamical Control Systems*, Springer, Berlin.
56. Oldenburg J., and Marquardt W. (2002). *Flatness and higher order differential model representations in dynamic optimization*, Computers and Chemical Engineering, Vol. 26, 385-400.
57. Picó-Marco E. (2004). *Nonlinear robust Control of Biotechnological processes, Tesis Doctoral, Valencia Technical University*; Departamento de Ingeniería de Sistemas y Automática.
58. Radhakrishnan Mahadevan, Sunil K. Agrawal, Francis J. and Doyle III. (2001). *Differential flatness based nonlinear predictive control of fed-batch bioreactors*, Control Engineering Practice, Vol. 9, pp. 889-899.
59. Rauh, A., Kletting M., Aschemann H., Eberhard P. (2005). *Robust Controller for Bounded State and Control Variables and Uncertain Parameters Using Interval Methods*, 2005 International Conference on Control and Automation (ICCA2005) June 27-29, 2005, Budapest, Hungary pp. 777-782.
60. Ratschan, S. (2003). *Solving Existentially Quantified Constraints with One Equality an Arbitrarily Many Inequalities*, in: Rossi, F. (ed.), Procceding of the Ninth International Conference on Principles and Practice of Constraint Programming, Springer, pp. 615-633.
61. Ratschan, S., and Vehí, J., (2004). *Robust Pole Clustering of Parametric Uncertain Systems Using Interval Methods*, in: Colaneri, P. (ed.), Robust Control design 2003-Proceedings of the 4th IFAC Symposium, Elsevier Science.
62. Remei C.P. (2005). *Análisis intervalar modal: Su construcción teórica, implementación y posibilidades de aplicación a la simulación y control*, tesis doctoral, barcelona, españa.
63. Rouchon, P., Fliess M., Lévine J., and Martin P. (1992). *Flatness and motion planning with n trailers*, Proc, European Control Conf., pp. 1518-1522.
64. Rouchon, P., Fliess M., Lévine J., and Martin P. (1993). *Flatness, motion planning and trailer systems*, proc. IEEE Control and Decision Conf., pp. 2700-2705.
65. Rouchon, P., (1996). *Vibrational control and flatness of chemical reactors*, in Multiconference on Computational Engineering in Systems Applications, pp. 211-212.
66. Rothfuss, R., Rudolph J., and Zeitz M. (1996). *Flatness based control of a nonlinear chemical reactor model*, Automatica Vol., 32(10), pp. 1433-1439.
67. Sainz M.A, Herrero P., Armengol J., and Vehí J. (2008). *Continuous minimax optimization using modal intervals*, ScienceDirect J. Mathematical Analysis and Applications. Vol. 339, pp.18-30.
68. Shary, S. P. (2002). *A New Technique in Systems Analysis under Interval Uncertainty and Ambiguity*, Realible Computing, Vol. 8(5), pp. 321-418.
69. Sira-ramírez, H. (2000). *Sliding mode control of PPR mobile robot with a flexible join.* In A. Isidori, F. Lamnabhi-Lagarrigue and W. Respondek (Eds) Nonlinear Control in the year 2000 (Vol. 2), Vol. 259 of Lecture Notes Control Information Science (London:Springer), pp. 421-442.

70. Sira-ramírez, H., and Agrawal S.K. (2004). *Differentially flay systems.* Marcel Dekker, Inc. New york.
71. Slotine, Jean-Jacques E., and Weiping Li. (1991). *Applied nonlinear control* Prentice Hall, New Jersey.
72. Weerasooriya S., and El-sharkawi M.A. (1991). *Identification and control of a DC motor using back-propagation neural networks.* IEEE Transaction on Energy Conversion, Vol. 6, No. 4., pp. 663-669.
73. Vehí J. and Sainz M. (1999). *Modal intervals. Application of interval analysis to systems and control*, SIGLA/X, Girona, Spain., pp. 157-221.
74. SIGLA/X. *Modal intervals. Application of interval analysis to systems and control*, Proceedings of MISC 99, pp. 157-227.
75. Terano T., Asai K., and Sujeno M. (1992). *Fuzzy Systems Theory and its Applications.*, Academic Press.
76. Tsang E.P.K. (1993). *Foundations of Constraint Satisfaction*, Academic Press, London and San Diego. ISBN 0-12-701610-4.
77. van Nieuwstadt, M., Rathinam M., and Murray R. M., (1994). *Differential flatness and absolute equivalence*, Proc. IEEE Control and Decision Conf., pp. 326-333.
78. van Nieuwstadt, M. J., and Murray R. M., (1998). *Real-time trajectory generation for differentially flat systems*, Int. Journal Nonlinear Control, Vol. 8, pp.995-1020.
79. Vehí J., Sainz M., Rodellar J., and Armengol J. (2000). *Análisis of the robustness of predictive controllers via modal intervals*, Realible computing, Vol. 6(3), pp. 30-52.
80. Vehí J., Sainz M. (1999). *Necessary and Sufficient Conditions for Robust Stability Using Modal Intervals*, IEEE pp. 673-676.
81. Vehí J. (1998). *Analysis and Design of Robust Controllers by means of Modal Intervals*, PhD thesis, Universidad de Girona. Spain. (in catalan).
82. Walter E., and Jaulin L. (1994). *Guaranteed Characterization of stability domains via set inversion*, IEEE Trans. On Automatic Control, AC-35, pp. 835-841.
83. Zadeh L.A. (1978). *Fuzzy Sets as a Basic for Theory of Possibility, Fuzzy Sets and Systems*, pp. 3-28.
84. Zadeh L.A. (1999). *Fuzzy Sets as a Basic for Theory of Possibility, Fuzzy Sets and Systems 100 Supplement*, North Holland, pp. 9-34 .
85. Zoran V., Ljubomir K., Dali D., and Sejid T. (2003). *Nonlinear Control Systems, Control Engineering Series*, Marcel Dekker, Inc.