Monotone Modules Approximation

Monotone Modules Approximation

Simona Constantinescu

September 3, 2011

Contents

Acknowledgements

This thesis, on monotone systems approximations, represents the most important project developed during my ETH Master studies. As the topic of monotone systems was of high interest for me ever since I started the Master in Zurich, I personally see this project as one integrating part my the main overall focus on the theoretical analysis of dynamical systems. In this aspect, for the present master thesis project and not only, I firstly send my gratitude towards my advisors, Hans-Michael Kaltenbach and Joerg Stelling, for their constant support, extremely valuable feedback and great discussions. I greatly appreciate the highly competitive, as well as relaxing, work atmosphere that Michael always establishes, as well as his exact, yet full of humour, way of explaining things. I did learn a lot about monotone sysems in the past two years and also a lot on how to write a scientific paper in this last month, having him as an example.
I am very grateful to my professor Joerg Stelling for always finding time to offer me his great feedback on all the projects I have been working on. From him I have learnt that, in systems biology, all topics are interconnected, you just have to know which analysis perspective to employ. Also, he is the ideal example about how to in detail and with highest quality study all these interconnected parts: from experimental biology to pure mathematics.
Finally, I could not have done any of the things I accomplished in these 2 Zurich years (and not only) without the support and equilibrium that my family, my boyfriend and my friends always surround me with. They constantly prove me, without any formal definitions, but just by simple counterexamples, that not all the things in this Universe are relative.

Introduction

The present thesis brings together two very promising topics in today's theoretical approaches in studying biological systems: *monotone systems* and *system approximation*. Informally, a monotone system is characterized by a robust response to external perturbations, in the sense of consistently - signed net changes in the concentration values of its species. On the other hand, approximating a system informally signifies finding a simpler system, which is similar enough to the original one such as to allow the two to be used interchangeably.

Monotone Systems

Unlike systems from applied mathematics or engineering, the biological systems are strongly governed by stochasticity and noise. Moreover, the lack of complete understanding of the biological process itself, originating mainly in the difficulty to reliably and in detail observe it, induces an extremely large degree of uncertainty in the parameter estimates for this type of systems. From a global systems biology perspective, this translates into a questionable reliability of the mathematical models proposed to describe the biological phenomena. Therefore, modeling techniques which overcome the parameter uncertainty issue possess a clear advantage from the reliability viewpoint. Monotone systems represent such a technique.

In order to offer an intuitive explanation on what the *robust consistently signed concentration change* means in the framework of monotone systems, lets briefly look at a concrete example. Assume the graph from the figure 1a is an abstraction of the interactions which take place in a very simple biological network: the 4 vertices of the graph are metabolites which are involved in a series of biochemical reactions. These metabolites, together with their interactions, represent a biochemical network. A directed edge exists between the species A and B whenever the concentration of A directly influences the concentration of B. In case this influence is positive, i.e. an increase in concentration of A brings about an increase in the concentration of B, the edge is assigned a $+$ sign (and, equivalently, a $-$ sign in case the influence is negative). For the sake of simplicity, let us assume that the system is as such that we can always fully state that the influence between two vertices, if existent, is either positive or negative.

Now assume that, due to an external perturbation, the concentration of species A has suddenly decreased. We are interested in knowing whether *we can predict the influence of an external perturbation on all the other species in the network, without knowing the exact values of the network's parameters?*. On the example at hand, if the concentration of A has suddenly decreased, then the concentration of B will obviously decrease as well, while the one of C will increase. More interesting is what will happen with D, as there exist 3 ways of getting from A to D. On the diagonal edge, the effect that a decrease in the concentration of A has on D is positive, hence D will decrease as well. On the path $A - B - D$, the effect is still positive, therefore the prediction is still a

decrease. Same happens on the path $A - C - D$. Therefore, we can conclude in this case that, when A decreases, D will decrease as well. Or, in other words, we can indeed predict the net effect that a sudden change in the concentration of one species will have on all the other species without using the exact parameter values of the network.

Now, lets assume that the graph of the network is slightly changed, i.e. the species A influences the species C in a positive rather than negative way, as in figure 1b. In this case, we cannot predict the effect a perturbation in A has on D anymore, as the paths leading to D have different signs. For making such a prediction, we will need to know the exact parameter values of the network.

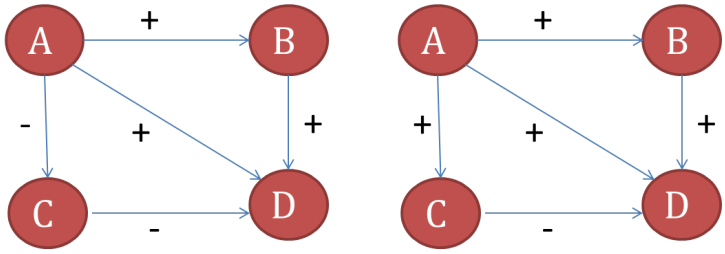
In this small example, the first system is monotone, while the second is not. This is an intuitive explanation of the claim that monotone systems are characterized by consistent response to perturbations

System Approximation

However, working with complete biological systems is not a feasible task, due to the very high number of species and the complexity of their interactions. Therefore, modularizations seems to be the only feasible approach in handling large systems. Particularly appealing is the modularization of a system into monotone modules, which possess the nice property exemplified here. As first modularization attempts have been conducted in this sense, the next arrising question is how one can make use of the monotone modules in predicting the dynamical behavior of the overall system and make use of their properties? The answer to this question is *system approximation*

Formulated as such, replacing a system through simpler, yet similar enough ones, is undoubtedly not a clear task. Notions as "simple or "similar enough need to be formally included through the theoretical framework of normed vector spaces and measurable functions which will be introduced in Chapter 3. The defined norms will basically measure the difference between the output trajectories of the system and, based on this quantitative information, rank them according to their similarity to the original one.

Various system approximation methods have been applied on monotone systems, the main difference between them being their empirical or theoretical approach. The results of these approximations will be discussed in this thesis.



(a) System1 (b) System2

Figure 1: Example networks

.

Chapter 1

Theory

The main concept lying at the basis of most of the theoretical and computational studies employed in the present thesis is the one of *dynamical systems*. Roughly speaking, a dynamical system depicts the functional time-dependency of a given set of variables in a well-defined geometrical space. A particular class of dynamical systems, often used in modeling in systems biology, are the ODE systems. In the ODE systems framework, the rate of change of the variables with respect to time is functionally depending on the system's variables. In this chapter, we will discuss in detail the basic concepts of (nonlinear and linear) ODE systems, as well as other closely related theoretical concepts, which will be relevant in the later analysis of these systems.

1.1 Dynamical Systems

Consider the following ordinary differential equations (ODE) system:

$$S : \begin{cases} \dfrac{\mathrm{d}x_1(t)}{\mathrm{d}t} = f_1(x_1(t), \ldots, x_n(t)) \\ \vdots \\ \dfrac{\mathrm{d}x_n(t)}{\mathrm{d}t} = f_n(x_1(t), \ldots, x_n(t)) \end{cases}$$

or, in more compact notation:

$$S : \frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t)) \tag{1.1}$$

where the measurable function $x : \mathbb{R}_+ \to \mathbb{R}^n$ is the *state* of the system and $x_0 = x(0)$ is its *initial condition*. We will denote by $\phi(t, x_0)$ the solution to this initial value problem. $\mathbb{R}_+$ represents the set of real, non-negative numbers.

Even if system 1.1 is often used in modeling, it has certain drawbacks when

employed in representing natural processes. More specifically, it cannot capture at least two essential features of this type of processes: *external influences* and *interconnections*. A natural conceptual extension of system 1.1, which also handles these two issues, is the *controlled* or *I/O ODE system* Σ:

$$\Sigma : \begin{cases} \dfrac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t), u(t)) \\ y(t) = g(x(t), u(t)) \end{cases} \tag{1.2}$$

with *state* $x : \mathbb{R}_+ \to \mathbb{R}^n$, *input* $u : \mathbb{R}_+ \to \mathbb{R}^m$, *output* $y : \mathbb{R}_+ \to \mathbb{R}^p$ and *initial condition* $x_0 = x(0)$. x, u and y are all measurable functions and will be from now on called *signals*. We will define the new *initial value problem* as being the ODE system 1.2, together with the initial condition x_0 and the input u. The solution to this initial value problem will be denoted by $\phi(t, x_0, u)$. The system form 1.2 is also known as a *state space representation* or, alternatively *state space realization*.

Note: In the present framework, all parameters of the system are assumed to be fixed. However, if this was not the case, the system 1.2 would have had the following new form, with P being the parameter vector:

$$\Sigma : \begin{cases} \dfrac{\mathrm{d}x(t)}{\mathrm{d}t} &= f(x(t), u(t), P) \\ y(t) &= g(x(t), u(t), P) \end{cases}$$

1.2 Linear Systems

A particular case of the ODE system 1.2 is the *linear ODE system* ([16]). A linear time - varying ODE system has the following form:

$$G : \begin{cases} \dfrac{\mathrm{d}x(t)}{\mathrm{d}t} &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{cases} \tag{1.3}$$

with, as before, $x : \mathbb{R}_+ \to \mathbb{R}^n$, $u : \mathbb{R}_+ \to \mathbb{R}^m$, $y : \mathbb{R}_+ \to \mathbb{R}^p$ and the following matrix - valued measurable functions: $A : \mathbb{R}_+ \to \mathbb{R}^{n \times n}$, $B : \mathbb{R}_+ \to \mathbb{R}^{n \times m}$, $C : \mathbb{R}_+ \to \mathbb{R}^{p \times n}$ and $D : \mathbb{R}_+ \to \mathbb{R}^{p \times m}$.
In case the matrix coefficients of the states and input are not time - dependent, system 1.3 becomes a linear time - invariant (LTI) system:

$$G : \begin{cases} \dfrac{\mathrm{d}x(t)}{\mathrm{d}t} &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{cases} \tag{1.4}$$

1.2.1 Time-domain Analysis

When knowing the initial condition x_0 and the input $u(\tau)$, for $\tau \in [0, t]$, the *full state response* $x(t)$ and the *full output response* $y(t)$ can be uniquely determined, $\forall t > 0$.

The full state response has the following form (for a complete derivation, see [24]):

$$x(t) = e^{At}x_0 + e^{At}\int_0^t e^{-A\tau}Bu(\tau)d\tau \tag{1.5}$$

where, if M is a nxn square real matrix, e^M is called the *matrix exponential of* M and it is a nxn square real matrix defined as:

$$e^M := \sum_{k=0}^{\infty} \frac{1}{k!}M^k$$

The expression 1.5 consists of two components: the first term is only dependent on the initial condition x_0 and it represents the *homogeneous response*, $x_h(t) = e^{At}x_0$, i.e. the solution to the system when assuming zero input. The second term is called the *forced response* and it represents the solution to the system when assuming zero initial condition.

The output response of the system, y, can be derived by substituting the state response into the algebraic system output equation:

$$y(t) = Cx(t) + Du(t) \tag{1.6}$$

By substituting $u(t) = 0$ in 1.6, we obtain the following form for the *homogeneous output response*:

$$y_h(t) = Ce^{At}x_0 \tag{1.7}$$

while after replacing the forced state response in 1.6, we obtain the *forced output response*:

$$y_f(t) = C\int_0^t e^{A(t-\tau)}Bu(\tau)d\tau + Du(t) \tag{1.8}$$

By combining 1.7 and 1.8, the output response of the LTI system 1.4 is:

$$y(t) = Ce^{At}x_0 + C\int_0^t e^{A(t-\tau)}Bu(\tau)d\tau + Du(t) \tag{1.9}$$

The forced output response and, equivalently, any LTI system for which the initial condition is set to zero, is equivalently described by the convolution integral as such:

$$y(t) = (g_c * u)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} g_c(t-\tau)u(\tau)d\tau \tag{1.10}$$

where g_c is called the *impulse response function* of the system and it represents the value of the output when the input is the *Dirac delta function* δ, i.e. a distribution satisfying the following properties:

$$\delta(t) = \begin{cases} +\infty, & t = 0 \\ 0, & t \neq 0 \end{cases}$$

and

$$\int_{-\infty}^{\infty} \delta(x)dx = 1$$

Note: As here we restricted our study to signals defined on the non-negative domain R_+, i.e. signals u for which $u(t) \equiv 0$, $\forall t < 0$, the convolution integral representation in 1.10 is equivalent to:

$$y(t) == \int_0^{\infty} g_c(t-\tau)u(\tau)d\tau \tag{1.11}$$

Any linear time-invariant system can be fully characterized by its impulse response function, that is, for any input function, the output can be calculated in terms of the input and the impulse response. More specifically, as it can be deduced from the equations 1.9 and 1.10, the impulse response function has the following matrix form, $\forall t \geq 0$:

$$g_c(t) = Ce^{At}B + Dt$$

1.2.2 Frequency-domain analysis

In the study of signals and systems, complementary to the *time domain analysis*, where functions are time-dependent, the *frequency domain analysis* can also be employed. In control theory, among the most common methods of transforming the time domain to the frequency domain are the *Laplace transform* ([30]) and the *Fourier transform*.

Definition 1.2.1. *The* Laplace transform *of a function* $f : \mathbb{R}_+ \to \mathbb{R}$ *is the function* F*:*

$$F(s) = \mathcal{L}\{f\}(s) := \int_0^{\infty} e^{-st} f(t)dt$$

where $s = \sigma + i\omega \in \mathbb{C}$*.*

Definition 1.2.2. *The* Fourier transform *of a function* $f : \mathbb{R}_+ \to \mathbb{R}$ *is the function* $\hat{f}$*:*

$$\hat{f}(s) = \mathcal{F}\{f\}(s) := \frac{1}{2\pi} \int_0^{\infty} e^{-i\omega t} f(t)dt$$

where $s = i\omega \in \mathbb{C}$*.*

The Fourier transform is therefore equivalent to evaluating the Laplace transform exclusively on the imaginary axis, i.e. for $s = i\omega$. From reasons of consistency with the control theory literature ([14], [11], [24]), we will, from now on, use the notation $F(j\omega)$ to denote the Fourier transform $\hat{f}(s)$, for $s = i\omega$, replacing the imaginary unit i with j. Also note that the extension of definitions 1.2.1 and 1.2.2 to the codomain $\mathbb{R}^n$ is done component-wise.

Applying the Laplace transform on the time-domain relationship 1.10 and using its properties ([30]) yields:

$$y(t) = (g * u)(t) \Rightarrow \mathbf{Y}(s) = \mathbf{G}(s) \cdot \mathbf{U}(s) \tag{1.12}$$

where $\mathbf{Y}(s) = \mathcal{L}\{y\}(s)$; $\mathbf{U}(s) = \mathcal{L}\{u\}(s)$ and $\mathbf{G}(s) = \mathcal{L}\{g\}(s)$.

The Laplace transform of the impulse response g_c of an LTI system G is caled the *transfer function* of the system: $\mathbf{G} : \mathbb{C} \to \mathbb{C}^{pxm}$

$$\mathbf{G}(s) = \int_0^\infty g_c(t)e^{-st}dt$$

and, in different notation:

$$\mathbf{G}(s) = \frac{\mathbf{Y}(s)}{\mathbf{U}(s)} \tag{1.13}$$

Using the properties of the Laplace transform ([16], [30]) in 1.13 an equivalent expression for the transfer function of the system 1.4 is:

$$\mathbf{G}(s) = C(sI - A)^{-1}B + D$$

As the codomain of $\mathbf{G}$ is $\mathbb{C}^{pxm}$, we will also refer to $\mathbf{G}$ as the *transfer function matrix* of the system 1.4. Moreover, any system described by a transfer function matrix is linear and time - invariant.

1.3 Stability

Given the (nonlinear) ODE system with inputs and outputs 1.2:

Definition 1.3.1. *The pair* $(\bar{x}, \bar{u}) \in (\mathbb{R}^n, \mathbb{R}^m)$ *is called an* equilibrium point/equilibrium input pair *if*

$$f(\bar{x}, \bar{u}) = 0_n.$$

If, given a fixed initial condition, the system 1.2 admits an equilibrium point/equilibrium input pair, we will say that the system (or its output signal) *reached steady-state*. The existence and characteristics of the equilibrium points play a significant role in the analysis of a system. They form the basis of what is called *stability analysis*, offering insights into the system's qualitative dynamical behavior. Especially in the monotone systems framework, equilibrium points are of central importance, as most of the existent results concern stability aspects ([2],[4],[3],[5]).

The meaning of the equilibrium point/equilibrium input pair is that, if the system is started at the initial condition $x(0) = \bar{x}$ and the input $\bar{u}$ is applied, then the system will not deviate from the initial state throughout its evolution, i.e. $\phi(t, \bar{x}, \bar{u}) = \phi(0, \bar{x}, \bar{u}), \forall t \geq 0$. Moreover, if under the initial condition x_0 and the input $\bar{u}$, the system reaches the state $\bar{x}$, i.e. $\exists T$ s.a. $\phi(T, x_0, \bar{u}) = \phi(0, \bar{x}, \bar{u})$, then the system will not leave this state anymore, i.e. $\phi(t, x_0, \bar{u}) = \phi(T, x_0, \bar{u}) = \phi(0, \bar{x}, \bar{u}), \forall t \geq T$.

Without going into the formal details of stability in the case of nonlinear systems (in this aspect, see [34]), we will here just briefly discuss this concept in an informal manner, with special attention given to the LTI case ([21]).
One important distinction regarding the equilibrium points of a system concerns

the *convergence properties* of the *nearby* or *close* (with respect to a given metric) solutions to the equilibrium point. More specifically, an equilibrium point is called *unstable* if the output (or state) trajectory cannot be kept close enough to it by starting sufficiently close and under the influence of an input which is sufficiently close to the equilibrium input. On the other hand, the equilibrium point is called *stable* if the trajectories can be kept as close as desired, by starting with an initial condition sufficiently close and under the influence of an input sufficiently close to the equilibrium input. In addition to this, the equilibrium point is called *asymptotically stable* if nearby solutions converge to it as well and *exponentially asymptotically stable* this happens at an exponential rate.
The second distinction is done regarding the starting point, i.e. the initial condition x_0. A *local* equilibrium point implies that the convergence properties hold provided that we start close enough, whereas *global* requires that these properties hold irrespective of where we start.
Due to the fact that, in the case of an LTI system, a close algebraic formula exists for the state and output trajectories of a system (equations 1.5 and 1.9), the stability of such a system can be analyzed through a simple algebraic test concerning solely the eigenvalues of the matrix A. For a proof of the following theorem, see [21].

Theorem 1.3.2. *For LTI systems, the following statements are equivalent:*

(a) The equilibrium solution is algebraically stable.

(b) The equilibrium solution is exponentially stable.

(c) $\forall \lambda_i \in eig(A)$, $Re(\lambda_i) < 0$.

where by eig(A) we mean the set of eigenvalues of the matrix A.

Such a matrix, whose eigenvalues have real negative part, is called *Hurwitz*. In case this happens for an LTI system G, we will informally refer to the system as being *stable*.

1.4 Linearization

As in the case of nonlinear systems, there is no way of calculating a closed form for the state and output responses ([34]), the algebraic test from theorem 1.3.2, as well as many other theoretical results ([16]) cannot be applied. This is the main justification behind studying nonlinear systems through linear surrogates. In this aspect, the most popular procedure employed is called *system linearization*.

In the case of a nonlinear system of equations, i.e. a multi-dimensional nonlinear function, the linearization procedure amounts to approximating the nonlinear system through a linear one, aslo called *the linearization of the nonlinear system*. The linear system is uniquely determined by a fixed multi - dimensional point, called *operating point*, and the matrix of first - order partial derivatives

of the function with respect to the system's states and inputs, evaluated at the operating point. The solution of the linearization will coincide with the one of the nonlinear system only when evaluated at the operating point.

In principle, the steady-state behavior need not always be of highest interest when analyzing dynamical systems. Not any system necessarily reaches a steady state or, depending on the application, the long - term behavior of the system might not be the main focus. However, in the present case, monotone systems almost always reach an asymptotically stable steady state ([32]). Therefore, when aiming at approximating monotone systems, maintaining their steady - state behavior is of particular interest.
This is why, in the present framework, we linearized the systems around their steady states.

1.4.1 Deviation variables

As described in [27], the mathematical formalism employed is defining *deviation variables* for both the system states and the input.

$$\delta_x(t) := x(t) - \bar{x};$$
$$\delta_u(t) := u(t) - \bar{u};$$

Re-writing the state - dependency from system 1.2 in the new deviation coordinates simply shifts the origin of the system, which is now the equilibrium pair:

$$\frac{\mathrm{d}\delta_x(t)}{\mathrm{d}t} = f(\delta_x(t) + \bar{x}, \delta_u(t) + \bar{u})$$

Now, performing a first-order Taylor expansion on the right-hand side:

$$\frac{\mathrm{d}\delta_x(t)}{\mathrm{d}t} \approx f(\bar{x}, \bar{u}) + \left.\frac{\partial f}{\partial x}\right|_{\substack{x=\bar{x};\\ u=\bar{u}}} \delta_x(t) + \left.\frac{\partial f}{\partial u}\right|_{\substack{x=\bar{x};\\ u=\bar{u}}} \delta_u(t)$$

and taking into account that, since $(\bar{x}, \bar{u})$ is an equilibrium pair, $f(\bar{x}, \bar{u}) = 0$, one obtains the new differential equation, with the deviation variables as unknows:

$$\frac{\mathrm{d}\delta_x(t)}{\mathrm{d}t} \approx \left.\frac{\partial f}{\partial x}\right|_{\substack{x=\bar{x};\\ u=\bar{u}}} \delta_x(t) + \left.\frac{\partial f}{\partial u}\right|_{\substack{x=\bar{x};\\ u=\bar{u}}} \delta_u(t) \tag{1.14}$$

As the Taylor expansion was trimmed after the first order term and the higher order terms were neglected, 1.14 represents *an approximation* of the rate of change of the system's states, f, in system 1.2. The behavior of the linear system is similar to the behavior of the nonlinear one as long as the deviation variables are close enough to the origin pair $(0_n, 0_m)$ (i.e. the states of the system are close enough to the equilibrium states and the input is close enough to the equilibrium input).
Equation 1.14 represents a linear, time - invariant differential equation, since,

in that representation, the coefficients of the functions δ_x and δ_u are constant matrices. The same expression as for the differential equation from the system 1.4 can be recovered by:

$$A := \left.\frac{\partial f}{\partial x}\right|_{\substack{x=\bar{x};\\ u=\bar{u}}} \in \mathbb{R}^{n\times n}; \qquad B := \left.\frac{\partial f}{\partial u}\right|_{\substack{x=\bar{x};\\ u=\bar{u}}} \in \mathbb{R}^{n\times m};$$

$$\frac{\mathrm{d}\delta_x(t)}{\mathrm{d}t} \approx A\delta_x(t) + B\delta_u(t)$$

In order to obtain comparable outputs for both systems, one needs to construct the following output for the linearization:

$$y'(t) = g(\delta_x(t) + \bar{x}, \delta_u(t) + \bar{u})$$

If g is a linear function as well, one recovers the linear system 1.4.

1.4.2 Example

We will now show an example of the linearization routine, as theoretically described in this section. For consistency, we chose an example system which will be encountered throughtout the next chapters as well, as it is one of the monotone example systems implemented for testing various approximation algorithms. The technical details on the form of the system, as well as its biological interpretation, can be found in the chapter on *Monotone Examples*, where the system is referred to as *Example1*. For clarity however, we will show here the I/O ODE form of the system:

$$\begin{cases} \dfrac{\mathrm{d}p}{\mathrm{d}t} = K_i u(t) - K_e p(t) - a_2 p(t) \\ \dfrac{\mathrm{d}q}{\mathrm{d}t} = K_e p(t) - K_i q(t) - a_3 q(t) + r(t)^2 \\ \dfrac{\mathrm{d}r}{\mathrm{d}r} = p(t)^2 - a_1 r(t) \\ y(t) = q(t) \end{cases} \tag{1.15}$$

The main purpose of employing this example here is showing how different outputs the linearization routine can produce, when changing the operating equilibrium point. The figure 1.1 shows the comparison between the output trajectory of the nonlinear system and two linearization trajectories, done around different operating point. Even if, at the moment, no vector norm is introduced in order to formally compare the signals, we will just make a visual argument, claiming not only that the trajectory linearized around the steady state is much more similar to the original system than the ones linearized around a non-steady-state point, but also that the further the value of the operating point is

from the steady-state value, the further the linear output trajectory is from the nonlinear one. Note, however, that the equilibrium input is constant.

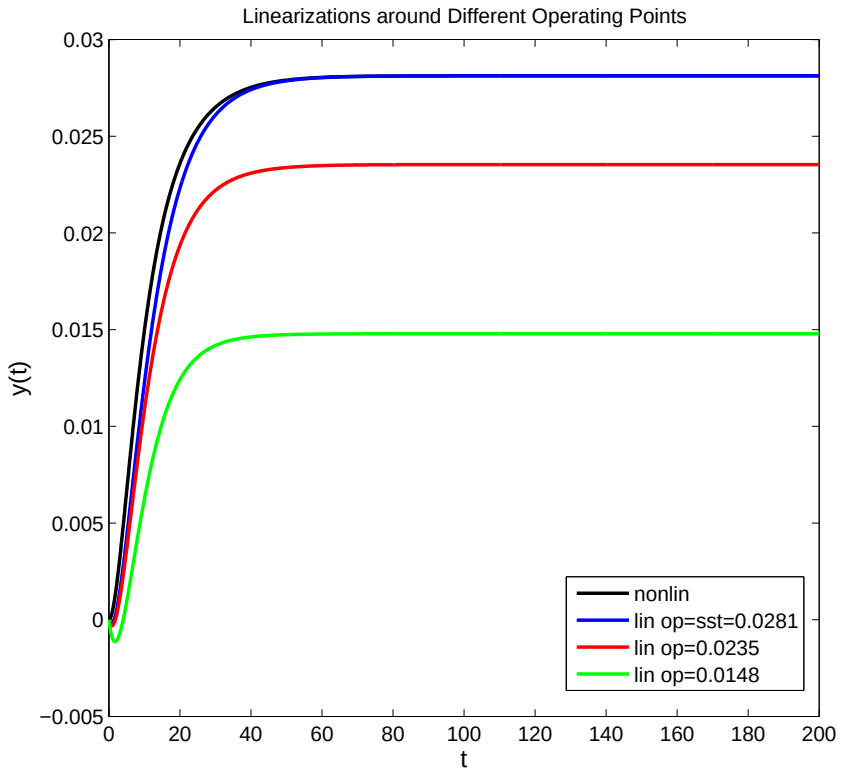


Figure 1.1: Simulated output for system 1.15 under the following parameter values: $a_1 = a_2 = a_3 = K_i = K_e = 0.1$, initial condition $(p(0), q(0), r(0)) = (0, 0, 0)$ and constant input $u(t) \equiv 0.1$. *nonlin* represents the output of the original system, *lin op=sst* represents the trajectory of the linearization done around the steady state, while the other two curves represent trajectories of linearizations done around the non-steady-state specified operating points. The similarity between the nonlinear outputs and the linear ones is the highest when the linearization is done around the steady state.

Chapter 2

Monotone Systems

2.1 Biochemical Reaction Networks

Even if started as a purely mathematical concept, monotone systems have evolved to be widely applicable when studying biochemical reaction networks.

A biochemical reaction network is identified through its participating *species* $\mathbf{S} = \{S_1, S_2, \ldots, S_n\}$ and by the way in which the species interact, i.e. the *reactions* $\mathbf{R} = \{R_1, R_2, \ldots, R_r\}$, formally defined through:

$$R_j : \sum_{i=1}^{n} \alpha_{ji} S_i \rightarrow \sum_{i=1}^{n} \beta_{ji} S_i$$

α_{ji} and β_{ji} are the *substrate* and *product molecularities* ([17]). The net production of species i due to reaction j is described by the *stoichiometries* $N_{ij} := \beta_{ji} - \alpha_{ji}$. For $1 \leq i \leq n$ and $1 \leq j \leq r$, the stoichiometries form what is called *the stoichiometry matrix* $\mathbf{N}$. In order to model this network as a dynamical system, each species will be represented by its time - dependent concentration $x_i(t)$. The vector $x = (x_1(t), \ldots, x_n(t))^T$ is the *state* of the system.

To each reaction R_j, a *rate equation* $v_j(x) : \mathbb{R}^n_+ \rightarrow \mathbb{R}$ is assigned, describing the velocity of the reaction. Let the vector $v(x) = (v_1(x), \ldots, v_r(x))^T$ denote the *rate vector* of the system. We will here work with rate equations only belonging to the class N1C, meaning that an increases in the concentration of a species cannot decrease the rate of any reaction and that a species not participating in a reaction does not influence its rate ([17]). The N1C class contains the mass - action, Michaelis - Menten and Hill - type rate laws. Also, in a network whose reactions are N1C, the stoichiometric matrix completely defines its topology.
As described above, the temporal dynamics of the network is given by the (nonlinear) uncontrolled ODE system:

$$\frac{dx}{dt} = Nv(x)$$

However, as biefly argued in the previous chapter as well, input and output channels are essential in modeling natural processes, as they represent the interconnections of the system with its exterior. More specifically, in case of a biochemical reaction network, the *inputs* u represent time-dependent external influences, such as the changing concentration of an external attractant or repellent, while the *outputs* y represent the "measurable" behavior of the system, such as (most commonly) GFP expression.
If modeled as an I/O system, the temporal dynamics of a biochemical reaction network is given by the following (nonlinear) controlled ODE system:

$$\frac{dx}{dt} = Nv(x(t), u(t)) \qquad y(t) = g(x(t), u(t))$$

2.2 Monotonicity

The main feature of monotone systems is their robust response to perturbations. The net effect (increasing or decreasing) that a sudden change in the concentration of one species has on any other species can be predicted without the exact knowledge of the parameters or the rate laws of the network. However, the existence of gradients and the proper use of such terms as *increasing* or *decreasing* silently implies working inside an ordered space, which needs to be mathematically formalized.
Informally, a system is said to be monotone if a partial ordering on the initial conditions and inputs is preserved on the states. Formally, defining a monotone system makes use of more advanced mathematical concepts. For the autonomous systems case, the main results in monotone systems theory belong to Hirsch and Smith ([31],[22]). These results were extended to controlled I/O case by Sontag ([33], [6]).

2.2.1 Dynamical Systems Perspective

Definition 2.2.1. *A* real Banach space $\mathbb{B}$ *is a complete normed vector space defined over the field of real numbers, i.e. a vector space* $(\mathbb{B}, \mathbb{R})$ *equipped with a norm* $||*||$*, in which any Cauchy sequence with respect to* $||*||$ *is also convergent, i.e. has a limit in* $\mathbb{B}$*.*

Definition 2.2.2. *K is called a* cone *of the real Banach space $\mathbb{B}$ if $K \neq \emptyset$, $K \subseteq \mathbb{B}$ and $\forall \alpha \in \mathbb{R}_+$, $\alpha K \subseteq K$. The cone K is said to be* positive *if it includes the null vector.*

Definition 2.2.3. *A* binary relation *R is an ordered triple (X, Y, G), where X and Y are two nonempty sets and G is a subset of the Cartesian product $X \times Y$. For any pair $(x, y) \in R$, i.e. $x \in X$, $y \in Y$ and $(x, y) \in G$, it can be stated that x* is related to *y and it will be denoted by xRy.*

Definition 2.2.4. *An* order $\succeq$ *is a binary relation (X, X, G) satisfying the following properties, $\forall x, y \in X$:*

1. $a \succeq a$ *(reflexivity);*

2. *if* $a \succeq b$ *and* $b \succeq a$*, then* $a = b$ *(antisymmetry);*

3. *if* $a \succeq b$ *and* $b \succeq c$*, then* $a \succeq c$ *(transitivity).*

The order is said to be total *if* $G = X \times X$*, i.e. any two elements of* X *need to be related. Otherwise, the order is said to be* partial.

Definition 2.2.5. *An* ordered Banach space *is a pair* $(\mathbb{B}, K)$*, together with the partial order* $\succeq$*, defined through:*

$$x_1 \succeq x_2 \Leftrightarrow x_1 - x_2 \in K$$
$$x_1 \succ x_2 \Leftrightarrow x_1 \succeq x_2 \text{ and } x_1 \neq x_2$$

As we will only work with real multi-dimensional functions, we will from now on restrict the ordered Banach space $\mathbb{B}$ to be equal to $\mathbb{R}^n$ in the case of the state vector domain, $\mathbb{R}^m$ for the input set and $\mathbb{R}^p$ for the output set.

Definition 2.2.6. *The uncontrolled dynamical system 1.1 is said to be monotone with respect to a partial order* $\succeq$ *defined on* $\mathbb{R}^n$ *if for any initial conditions* x_1 *and* x_2*:*

$$x_1 \succeq x_2 \Longrightarrow \phi(t, x_1) \succeq \phi(t, x_2), \forall t > 0$$

where $\phi(t, x_i)$ *is the time-dependent solution of the ODE system 1.1.*

For the case of controlled systems with inputs and outputs, definition 2.2.6 becomes:

Definition 2.2.7. *The controlled dynamical system 1.2 is said to be monotone with respect to three partial orders* $\succeq_x$ *defined on* $\mathbb{R}^n$*,* $\succeq_u$ *defined on* $\mathbb{R}^m$ *and* $\succeq_y$ *defined on* $\mathbb{R}^p$ *if for any initial conditions* x_1 *and* x_2*, any inputs* u_1 *and* u_2*:*

$$x_1 \succeq_x x_2 \text{ and } u_1(t) \succeq_u u_2(t) \Longrightarrow \phi(t, x_1, u_1) \succeq_x \phi(t, x_2, u_2), \forall t > 0$$

where $\phi(t, x_1, u_1)$ *is the time-dependent solution of the ODE system 1.1.*

If there exist no input nor output space, one recovers the definition of monotonicity for uncontrolled systems 2.2.6.

By choosing the corresponding cones with respect to which the partial orders are defined to be of the form $K_x = \{x \in \mathbb{R}^n | e_n^T x \geq 0\}$, $K_u = \{u \in \mathbb{R}^m | e_m^T u \geq 0\}$ and $K_y = \{y \in \mathbb{R}^p | e_p^T y \geq 0\}$, where $e_i = \{-1, 1\}^i$, for $i \in \{n, m, p\}$, one can identify whether a system is monotone or not by employing graph - theoretical tools. Therefore, we will restrict our study from now on to partial orders defined on these type of cones, called *orthant cones*. Moreover, from convenience and without the risk of confusion, all partial orders will be denoted by $\succeq$.

2.2.2 Graph Theory Perspective

The *species graph* associated with the uncontrolled system 1.1 is the directed signed graph $G = (V, E, w)$ ([36]). V is the set of vertices, with $|V| = n$, representing the biological species. We will denote its elements by v_i, with $1 \leq i \leq n$ and the directed edge connecting vertices v_i and v_j by e_{ij}. The set of edges E of the graph is defined on the basis of the corresponding entries of the following matrix, called *the Jacobian* of the system:

$$\mathbf{J} : \mathbf{j}_{ij} = \frac{\partial f_i}{\partial x_j}, \forall\ 1 \leq i, j \leq n$$

As long as the sign of the (i, j)-th entry doesn't vanish identically, $\forall x \in \mathbb{R}^n_+$, a directed edge e_{ij} between the vertices v_j and v_i exists. In this case, if the sign of the (i, j)-th entry is constant, $\forall x \in \mathbb{R}^n_+$, a *weight* is assigned to the edge e_{ij}: $w(e_{ij}) = sgn(\mathbf{j}_{ij})$. If the sign is not constant, then, in the first stage, the edge e_{ij} is assigned both a $(+)$ and a $(-)$. This assignment however doesn't properly define a weighting function $w : E \to \{-1, 1\}^{|E|}$. This conflict is solved by the introduction of an additional vertex v, conntected only with vertices v_i and v_j, by edges e_{iv} and e_{vj}, with $w(e_{iv}) = +1$ and $w(e_{vj}) = +1$. The edge e_{ij} is now assigned a constant sign, i.e. $w(e_{ij}) = -1$ and the weighting function is properly defined (see figure 2.1 for more details on the additional vertex introduction procedure. Note, however, that in the figure 2.1, a different conflict is solved, which appears after the weights on the directed graph have been properly defined). In the case of continuous systems (such as 1.1), the diagonal entries of the Jacobian, i.e. *self - loops* in the graph, are ignored when constructing the species graph (see [32] for additional details).

In case of the controlled system 1.2, the *species graph* is extended as to include additional vertices for each input and output channel. Therefore, the graph will have $(m + n + p)$ vertices. The edges connecting the input u_j and output y_j channels, as well as their weights, will be constructed on the basis of the following partial derivatives:

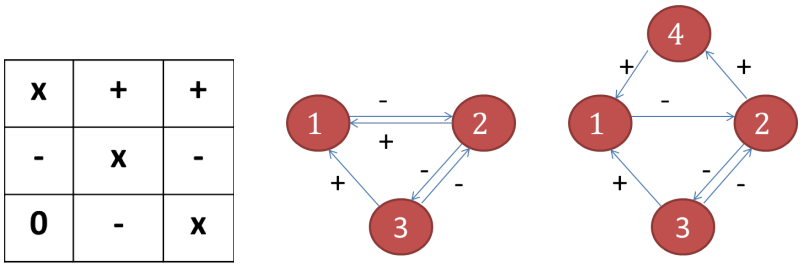
$$\frac{\partial f_i}{\partial u_j} \quad \text{and} \quad \frac{\partial f_i}{\partial y_j}$$

in the same manner as for the uncontrolled system. The direction of the new edges is always the same: starting from the input vertices and directing towards the species vertices or starting from the species vertices and directing towards the output vertices. There are no edges directed towards the input vertices, as well as no edges starting from the output vertices.

In identifying monotonicity, the graph theoretical criterion involves two equally important concepts: a *consistent spin assignment*, defined on a directed signed graph, and a *negative cycle*, defined on an undirected signed graph $G^u = (V, E^u, w^u)$.

Definition 2.2.8. *Given a signed graph $G = (V, E, w)$, a* spin assignment *is a function which assigns, to each vertex, a sign, either + or -:*

$$\sigma : V \to \{-1, 1\}^{|V|}$$



(a) (b) (c)

Figure 2.1: Additional vertex introduction: the conflict solving for obtaining unique edge weights in the species graph, exemplified on a fictive 3-species system. Subfigure 2.1a represents the corresponding signs of the Jacobian matrix of the fictive system; a + signifies a positive influence, - a negative one, while *0* means that there is no influence between the two species. As the diagonal entries can be ignored, they are not depicted. Subfigure 2.1b represents the initial corresponing species graph, with edges having conflicting signs. Subfigure 2.1c is the resolved directed graph, where, if existent, the edges between two vertices have same signs, regarless of their orientation. This directed graph is used as basis for constructing the undirected counterpart of the species graph.

The spin assigments terminology is borrowed from statistical mechanics. For more information on the relation to Ising spin-glass models, see [32].

Definition 2.2.9. *Given a signed graph $G = (V, E, w)$ and a spin assignment σ, one says that σ is a* consistent spin assignment *for G iff, $\forall i, j \in V$ and $e_{ij} \in E$, $w(e_{ij})\sigma(i)\sigma(j) = 1$.*

The following term was introduced by Harary in ([15]):

Definition 2.2.10. *A graph G for which a consistent spin assignment exists is said to be* balanced*.*

Definition 2.2.11. *Given a graph $G' = (V', E')$, a* cycle *is a sequence of vertices $C = (v'_1, v'_2, \ldots, v'_c)$, where $\exists e'_{i,i+1}, \forall 1 \leq i < c$, $v'_1 = v'_c$ and $|\{(v'_1, v'_2, \ldots, v'_{c-1})\}| = c - 1$.*

A cycle C represents a closed path along the edges of a graph, with no repeating vertices other than the starting and the ending one and no repeating edges. When referring to an *undirected cycle*, the orientation of the edges will not be taken into consideration, even if the underlying graph is directed.

Definition 2.2.12. *Given a cycle $C = (v'_1, v'_2, \ldots, v'_c)$ of the signed graph $G' = (V', E', w')$, the* sign of the cycle C *is $sgn(C) = \prod_{i=1}^{c-1} w'(e'_{i,i+1})$.*

The sign of a cycle simply reperesents the product of the weights of its edges. In the case of the species graph, as for any edge (i, j), $|w(e_{ij})| = 1$, the sign of

any cycle C will also satisfy $|sgn(C)| = 1$. Hence, we will only refer to either *positive* or *negative* cycles in the directed species graph $G = (V, E, w)$.

Even if the species graph is a directed one, the graph - theoretical criterium for detecting monotonicity in a system makes use of the undirected version of G, which we will call G^u. In transforming the directed graph into its undirected counterpart, the edges loose their orientation, i.e. $e^u_{ij} = e^u_{ji}$. More specifically, we can here distinguish three cases:

1. Only one of the edges e_{ij} or e_{ji} exists.
   In this case, in the undirected graph G^u, we will say that vertices v_i and v_j are connected through the edge e_{ij} or, equivalently, e_{ji}, as the directed edge looses its orientation. Moreover, the weight of the new edge: $w^u(e^u_{ij})$ will be equal to the one of the existent edge in the directed version of the graph.

2. Both edges e_{ij} or e_{ji} exist and $w(e_{ij}) = w(e_{ji})$.
   This case is treated equivalently to the first case. As both edges have the same weight, there will be no conflict and, in the undirected graph, the new edges will have the same sign as the two edges from the directed version.

3. Both edges e_{ij} or e_{ji} exist and $w(e_{ij}) \neq w(e_{ji})$.
   In this case, a conflict appears, as it is unclear which sign should be assigned to the edge from the undirected graph. The conflict is solved through the same trick of additional vertex introduction (see figure 2.1 for additional explanations). In this way, each edge has an unique weight assigned and the signed graph (G^u, E^u, w^u) is properly defined.

The following property (initially proven by Harrary in [15] and then by Sontag in [32]) shows the equivalence between the balancing property of a graph and its cycles structure:

Proposition 2.2.13. *An undirected graph $G^u = (V, E^u, w^u)$ is balanced iff it has no negative cycles.*

Proof.
Necessity: Assume that the undirected signed graph $G^u = (V, E^u, w^u)$ is balanced, i.e. it admits a consistent spin assignment σ. Then, for any undirected cycle C_i: $C_i = (v_{1i}, v_{2i}, \ldots, v_{ci})$, with $v_{ci} = v_{1i}$, one has that:

$$sgn(C_i) = \prod_{j=1}^{c-1} w^u(e^u_{ji;(j+1)i}) = \prod_{j=1}^{c-1} \frac{1}{\sigma(v_{ji})\sigma(v_{(j+1)i})} = \prod_{j=1}^{c} \frac{1}{\sigma(v_{ji})^2} = 1$$

The above expression holds because all the spin assignments of vertices are counted twice. And, as they were assigned either a $+1$ or a -1, the square of each individual assignment will equal 1, hence the product of squares of the assignments for all vertices will equal 1 as well.

Sufficiency: Assume that the undirected signed graph $G^u = (V, E^u, w^u)$ has no negative cycles, i.e. any two paths between any two vertices have the same sign. Assume as well, without loss of generality, that the graph is connected (in case it is not, same procedure is repeated for all the connected components separately). Pick any random vertex to start with, say $v_1 \in V$ and assign it with the value $\sigma(v_1)$, either $+1$ or -1. Then continue with all the vertices v_i with which v_1 is connected and assign them the function $\sigma(v_i) = \dfrac{1}{w(e^u_{v_1 v_i})\sigma(v_1)}$. Continue the procedure recursively for all the neihbors v_i and their neighbours, as long as there remains any unlabeled vertex. The connectivity of the graph ensures that each vertex will be visited. The fact that any two paths between any two vertices have same parity ensures that any vertex v will be uniquely assigned a value $\sigma(v)$. □

Note: In [32], proposition 2.2.13 is stated for a *directed* graph. However, the negative cyles are always considered to be undirected. Therefore, as one makes no use of the edge orientation in the graph, we here formulated the equivalence property directly in terms of the undirected graph $G^u = (V, E^u, w^u)$.
However, for identifying whether a graph is balanced, Harary introduced, in [15], a linear running-time dynamic-programming algorithm, based on the idea described in the proof. For a balanced connected graph, two consistent spin assignments exist, one being the reverse of the other (each vertex sign is reversed) [32]. Equivalently, this algorithm can be used for checking monotonicity.
Hence, the following proposition, representing a graph-theoretical criterium for identifying monotonicity, is a just consequence of the individual results presented in this chapter.

Proposition 2.2.14. *A system is monotone iff its species graph is balanced or, equivalently, iff its species graph has no negative undirected cycles.*

2.3 Properties of monotone systems

Monotone systems show restricted dynamical behavior. For instance, in the autonomous case, stable oscillations are ruled out ([22], [31]) and, whenever there exists a single steady state, under mild assumptions on the parameter values and on the boundness of solutions, which are almost always satisfied in biological networks, any solution converges to that steady state ([32]). This monostationarity result is often considered to be the first important result in the field of autonomous monotone systems and it belongs to Dancer ([10]) for the discrete case. The extension to the continuous case belongs to Enciso ([12]). Moreover, as stated by Hirsch, the existence of attractive periodic solutions is precluded for uncontrolled monotone systems. For a proof on the following theorem, see ([22]):

Theorem 2.3.1. *Let $K \subseteq \mathbb{R}^n$ be a closed cone with nonempty interior, i.e. a cone closed under convex combinations. If the system 1.1 is monotone with respect to K, then it has no attractive periodic solutions.*

The following definitions and propositions, introduced by Angeli and Sontag in [2], are useful in describing the dynamical behavior of controlled monotone systems. See [2] for the proofs of all the propositions stated here.

Definition 2.3.2. *A controlled dynamical system of the form 1.2 is endowed with the* static input-to-state characteristic*:*

$$\kappa_x(\cdot) : \mathbb{R}^m \to \mathbb{R}^n$$

if, for each constant input $\bar{u}$, there exists a (necessarily unique) globally asymptotically stable equilibrium.

In other words, the input-to-state characteristic represents the unique mapping between the value of the constant input $\bar{u}$ and the value of the (globally asymptotically stable) equilibrium point of the system, correspnding to $\bar{u}$. Note that the initial condition is assumed to be fixed.

Definition 2.3.3. *If the controlled dynamical system of the form 1.2 admits an input-to-state characteristic and if the function g is continuous, then it exists the* static input-to-output characteristic*:*

$$\kappa_y(\bar{u}) := g(\kappa_x(\bar{u}))$$

The input-to-output characteristic hence represents the unique mapping between the value of the constant input $\bar{u}$ and the value of the output function, evaluated at the (globally asymptotically stable) equilibrium point of the system, corresponding to $\bar{u}$.

Proposition 2.3.4. *If the controlled dynamical system of the form 1.2 is monotone and it admits a static input-to-state characteristic κ_x, then κ_x must be non-decreasing with respect to the orders in question, i.e. $\bar{u} \succeq \bar{v} \Rightarrow \kappa_x(\bar{u}) \succeq \kappa_x(\bar{v})$, where $\bar{u}$ and $\bar{v}$ are two constant inputs $\in \mathbb{R}^m$.*

Proof. Let x_1 be any initial condition. The system 1.2 is monotone (definition 2.2.7), therefore it hold that:

$$\phi(t, x_1, \bar{u}) \succeq \phi(t, x_1, \bar{v}), \quad \forall t \geq 0$$

If letting $t \to \infty$, then one recovers the steady - state value, or, in other words:

$$\kappa_x(\bar{u}) \succeq \kappa_x(\bar{v})$$

□

Definition 2.3.5. *The order $\succeq$ defined on $\mathbb{R}^n$ is* bounded *if the following two properties hold:*

1. *For each bounded subset $S \subseteq \mathbb{R}^n$, there exist two elements $a, b \in \mathbb{R}^n$ such that $S \subseteq [a, b] = \{x \in \mathbb{R}^n | b \succeq x \succeq a\}$;*

2. *For each $a, b \in \mathbb{R}^n$, the set $[a, b]$ is bounded.*

Property 1 from definition 2.3.5 is equivalent to the cone K having nonempty interior, while property 2 holds if the cone $K \in \mathbb{R}^n$ is positive (see definition 2.2.2). Therefore, it is evident that the orders defined on orthant cones are bounded. The following proposition guarantees bounds on state and output trajectories of a monotone I/O system, under bounded inputs:

Proposition 2.3.6. *Consider a monotone system defined on an orthant cone in $\mathbb{R}^n$ which is endowed with a static inputstate characteristic, and suppose that the order on the state space $\mathbb{R}^n$ is bounded. Pick any input u whose values lie in a closed interval $\in \mathbb{R}^m$. Let $x(t) = \phi(t, x_1, u)$ be any trajectory of the system corresponding to the input u and initial condition x_1. Then, $x(t)$ is a bounded subset of $\mathbb{R}^n$, $\forall t \geq 0$.*

Proposition 2.3.7. *If 1.1 is a monotone system which is endowed with a static inputstate characteristic κ_x, then κ_x is a continuous function. Moreover, for each $\bar{u} \in \mathbb{R}^m$ and $\bar{x} = \kappa_x \bar{u}$, the following properties hold:*

1. *For each neighborhood P of $\bar{x} \in \mathbb{R}^n$, there exist a neighborhood P_0 of $\bar{x} \in \mathbb{R}^n$ and a neighborhood Q_0 of $\bar{u} \in \mathbb{R}^m$ such that $\phi(t, x_1, u) \in P$, $\forall t \geq 0$, $\forall x_1 \in P_0$ and all inputs u such that $u(t) \in Q_0$, $\forall t \geq 0$.*

2. *If, in addition, the order on the state space $\mathbb{R}^n$ is bounded, then, for each input u whose values $u(t)$ lie in some interval $[c, d] \in \mathbb{R}^m$ and with the property that $u(t) \to \bar{u}$ and all initial states $x_1 \in \mathbb{R}^n$, necessarily $x(t) = \phi(t, x_1, u) \to \bar{x}$ as $t \to \infty$.*

The first part of proposition 2.3.7 extends the boundness of trajectories for I/O monotone systems and basically states that there can be found neighborhoods for initial conditions and inputs such that the trajectories of the system as kept as close as possible to their steady states, for all times. Moreover the second part of the proposition states that, under a converging input and a bounded order, all trajectories will ultimately converge to the existent steady state (monostability).

Chapter 3

Performance Evaluation

Given a dynamical system, the goal of the *system approximation* procedure is to find other systems with "similar" dynamical behavior. In this direction, a theoretical framework needs to be defined, which should not only quantify "similarity", but also, based on it, rank the systems. Essentially, a quantitative score is assigned to each approximation, representing how close it is to the original system. However, some basic principles need to be taken into account when performing the assignment procedure. In case these principles are followed, the assignment procedure is called *a norm*.

Recall that, given a vector space $\mathbf{V}$ over the field of complex numbers $\mathbb{C}$, a *norm* is a function $||\cdot|| : \mathbf{V} \to \mathbb{R}$ with the following properties, satisfied $\forall a \in \mathbb{C}$ and $\forall\ \mathbf{u}, \mathbf{v} \in \mathbf{V}$:

1. $||a \cdot \mathbf{v}|| = |a| \cdot ||\mathbf{v}||$ *(positive homogenity);*
2. $||\mathbf{u} + \mathbf{v}|| \leq ||\mathbf{u}|| + ||\mathbf{v}||$ *(triangle inequality);*
3. if $||\mathbf{v}|| = 0$ then $\mathbf{v} = \mathbf{0}$ *(separating points).*

A simple consequence of the first property is that $||\mathbf{0}|| = 0$, which, combined with the triangle inequality, results in the fourth norm property:

$$||\mathbf{v}|| \geq 0 \text{ (non-negativity).}$$

Definition 3.0.8. *A* seminorm *is a norm for which the* 3^{rd} *property, the separating points, is not satisfied.*

The (semi)norms used for evaluating the performance of the approximated systems with respect to the original one will be divided into two classes ([14],[11]), depending on whether the analysis is done on outputs produced under any input of choice (*signal norms*) or on outputs produced under certain input signals and, in some cases, even under a wider input range (*system norms*).

3.1 Signal Norms

3.1.1 The Error Signal

Consider two ODE systems, Σ and $\hat{\Sigma}$, with same number of input and output channels:

$$\Sigma = \begin{cases} \dfrac{\mathrm{d}x(t)}{\mathrm{d}t} &= f(x(t), u(t)) \\ y(t) &= g(x(t), u(t)) \end{cases} \tag{3.1}$$

with $x : \mathbb{R}_+ \to \mathbb{R}^n$, $u : \mathbb{R}_+ \to \mathbb{R}^m$ and $y : \mathbb{R}_+ \to \mathbb{R}^p$;

$$\hat{\Sigma} = \begin{cases} \dfrac{\mathrm{d}\hat{x}(t)}{\mathrm{d}t} &= \hat{f}(\hat{x}(t), \hat{u}(t)) \\ \hat{y}(t) &= \hat{g}(\hat{x}(t), \hat{u}(t)) \end{cases} \tag{3.2}$$

with $\hat{x} : \mathbb{R}_+ \to \mathbb{R}^{n'}$, $\hat{u} : \mathbb{R}_+ \to \mathbb{R}^m$ and $\hat{y} : \mathbb{R}_+ \to \mathbb{R}^p$;
As previously mentioned, the measurable functions x, u, y, $\hat{x}$, $\hat{u}$, $\hat{y}$ are called the *signals* of the systems.

Definition 3.1.1. *Given two systems Σ (3.1) and $\hat{\Sigma}$ (3.2) and an input signal $u : \mathbb{R}_+ \to \mathbb{R}^m$, the* error signal $e_{\Sigma,\hat{\Sigma},u} : \mathbb{R}_+ \to \mathbb{R}^p$ *is:*

$$e_{\Sigma,\hat{\Sigma},u}(t) = |y(t) - \hat{y}(t)| \tag{3.3}$$

The error signal, depending on the two systems and on the input u, represents the absolute value of the difference between the two outputs, produced under the same input u. For $p > 1$, the difference between the outputs, as well as the absolute value, are done component-wise. In comparing the outputs produced by the two systems, the following signal norms will be applied on the error signal.

3.1.2 The L_p norms

In order to remain consistent with the literature, we will here use the notation p when referring to the most general form of what, in functional analysis, is called the L_p *norm*. Even if p is generally used throughout this thesis as being the dimension of the output space, any confusion will be avoided, as the signification of the notation will always be clear from the context.

Definition 3.1.2. *The* pseudo-Euclidean norm *of a signal $s : \mathbb{R}_+ \to \mathbb{R}^n$, $s(t) = (s_1(t), \ldots, s_n(t))$ is:*

$$|s(t)|_2 := \sqrt{s_1^2(t) + \ldots + s_n^2(t)}$$

Definition 3.1.3. *The* L_p-norm *of a signal $s : \mathbb{R}_+ \to \mathbb{R}^n$ is, for $p \neq 0$:*

$$||s||_p := \left(\int_0^\infty |s(t)|_2^p dt \right)^{1/p} \tag{3.4}$$

Definition 3.1.4. *The* finite - horizon L_p-norm *of a signal* $s : \mathbb{R}_+ \to \mathbb{R}^n$ *is, for* $p \neq 0$*:*

$$||s||_{p,[0,T]} := \left(\int_0^T |s(t)|_2^p dt \right)^{1/p} \tag{3.5}$$

The integrals are properly defined, as the signals are (Lebesque) measurable functions.
We will also use the following norm of the signal s, as the counterpart of the norm 3.5 with respect to the norm 3.4:

$$||s||_{p,[T,\infty]} := \left(\int_T^\infty |s(t)|_2^p dt \right)^{1/p}$$

.

Depending on the value of p in definitions 3.1.3 and 3.1.4, the L_p-norms have different interpretations and extract different information about the error signal and, equivalently, about how similar the compared signals are. Hence, the most encountered norms when studying signals ([14],[11]) are:

1. The L_1-norm of a signal s represents the integral of its absolute value:

$$||s||_1 := \int_0^\infty |s(t)|_2 dt$$

   When being applied to the error signal, the L_1-norm computes the total deviation between the two outputs.

2. The L_2-norm of a signal s represents the square root of the integral of $s(t)^2$:

$$||s||_2 := \Big(\int_0^\infty |s(t)|_2^2 dt \Big)^{1/2} \tag{3.6}$$

   In a more physical interpretation, assuming that the error signal $e(t)$ represents a voltage or a current, then the total energy associated with the signal equals ${||e||_2}^2$.

3. The L_∞-norm of a signal s represents the supremum of its absolute value:

$$||s||_\infty := \sup_{t \geq 0} |s(t)|_2$$

   When being applied to the error signal, the L_∞-norm computes the maximum deviation between the two compared outputs.

Note: The finite time horizon L_1, L_2 and L_∞ norms are obtained by replacing the corresponding values of p in definition 3.1.4.

3.1.3 The L_2 Frequency Norm

Definition 3.1.5. *The* Singular Value Decomposition *(*SVD*) of $A \in \mathbb{C}^{mxn}$ is the set of matrices (U, V, Σ) such that $A = U\Sigma V^H$, where X^H denotes the conjugate transpose of the matrix X, U is an mxm unitary matrix, i.e. $U^H U = UU^H = I_m$, V is an nxn unitary matrix and Σ is an mxn diagonal matrix with nonnegative real numbers σ_i on the diagonal. Without loss of generality, we can assume that $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_p \geq 0$, where $p = min(m, n)$. σ_i are called the* singluar values *of the matrix A.*

As it is proven in most modern linear algebra books, any matrix $A \in \mathbb{C}^{nxm}$ admits a SVD.

Definition 3.1.6. *Let $A \in \mathbb{C}^{mxn}$. The* Frobenius norm *(or the* Hilbert - Schmidt norm*) of A, $||\cdot||_F$ is defined as:*

$$||A||_F := \sqrt{tr(A^H A)} = \sqrt{\sum_{i=1}^{min\{m,n\}} \sigma_i^2}$$

where σ_i are the singular values of A.

In analogy with the L_2-norm in the time domain (expression 3.4), we define a L_2-norm on the Laplace - transformed signal on the imaginary axis $\mathbf{U}(s)$, where $s = j\omega$:

$$||\mathbf{U}||_2 := \Big(\int_0^\infty ||\mathbf{U}(j\omega)||_F^2 \Big)^{1/2} = \Big(\int_0^\infty \mathbf{U}(j\omega)^H \hat{U}(j\omega) \Big)^{1/2} \tag{3.7}$$

Parseval's theorem ([14],[11]) states that the time - domain and the frequency domain L_2-norms are equal. Even though norms in the frequency domain can be defined for other values of p than 2, Parseval's identity doesn't exist for any $p \neq 2$:

$$||u||_2 = ||\mathbf{U}||_2$$

3.1.4 The Finite Horizon Average Norm

In addition to the classical L_p norms, we will here introduce a new measure, which we will refer to as *the finite horizon average norm*. This measure computes the total deviation between the two signals (i.e. the finite horizon L_1 norm), divided by the lengh of the horizon interval. As we only work with signals starting from 0, we restricted the use of the norm on the interval $[0, T]$, where T is the upper limit of the finite horizon time. However, same results hold when defining the norm over any real fixed positive interval $[a, b]$. Moreover, we restircted the domain of the functions of interest to $\mathbb{R}_+^n$, as the finite horizon average norm presented here will be applied only on error signals, which are always non-negative (see definition 3.1.1).

Definition 3.1.7. *Given $a, b \in \mathbb{R}$, with $b > a$, the* average *of a function $f : \mathbb{R}_+ \to \mathbb{R}_+^n$ on the interval $[a, b]$ is the function $f_{\text{avg,a,b}} : \mathbb{R}_+^n \to \mathbb{R}_+^n$:*

$$f_{\text{avg,a,b}}(f) := \begin{cases} 0 & \text{for } a = b \\ \dfrac{\int_a^b f(x)dx}{b-a} & \text{otherwise} \end{cases} \tag{3.8}$$

Proposition 3.1.8. *The function $f_{\text{avg,0,T}}$ is a* norm *on the vector space $\mathbb{R}_+^n$ for fixed $T > 0$.*

Proof. Let $a \in \mathbb{R}_+$, $f, g : \mathbb{R}_+ \to \mathbb{R}_+^n$ and $T_1, T_2 \in \mathbb{R}_+$.

1. The positive homogenity property is satisfied:

$$f_{\text{avg,0,T}}(af) = \frac{\int_0^T (af)(x)dx}{T} = a\frac{\int_0^T af(x)dx}{T} = af_{\text{avg,0,T}}(f)$$

2. The triangle inequality property is satisfied:

$$\begin{aligned} f_{\text{avg,0,T}}(f+g) &= \frac{\int_0^T (f+g)(x)dx}{T} = \frac{\int_0^T (f)(x)dx}{T} + \frac{\int_0^T (g)(x)dx}{T} \\ &= f_{\text{avg,0,T}}(f) + f_{\text{avg,0,T}}(g) \end{aligned}$$

3. The separating points property is satisfied, $\forall T \neq 0$:
   If $T \neq 0$ and $f_{\text{avg,0,T}}(f) = 0_n$, then, as $f \in \mathbb{R}_+^n$, it follows that $f = 0_n$

□

3.1.5 The Stability Seminorm

The signal norms introduced until now do not offer any information on the *stability* properties of the signals being compared. As presented in Chapter 2, under mild assumptions, monotone systems admit a single stable steady state (see proposition 2.3.7 or the paper [2] for more details). Therefore, it makes sense that capturing stability properties into the definition of the norms used for monotone systems evaluation would offer more insight into the intrinsic dynamical behavior of monotone systems *per se*.
It is undoubtedly not an easy task to mathematically formalize the "core characteristics" of monotone systems and to formally introduce them into the evaluation procedure. Moreover, a necessary condition in designing a new ranking system is assuring that the new measure satisfies the *(semi)norm* requirments, especially the *triangle inequality*, which considerably restricts the space of available measures.
Unlike the L_p norms, the *stability seminorm* is more specific about the qualitative behavior of the monotone system to be approximated. The measure is only defined in the case in which the monotone system reaches steady state. Once

this requirment is fulfilled, it weights differently the scores of the individual L_p norms before and after the monotone system reached steady state.
The idea behind this norm was to introduce the steady state offset of the two signals into the ranking procedure, in case the approximated system reaches steady state as well. Even if, from a theoretical viewpoint, any signal norm could have been used on comparing the two signals before they reach steady state, we restriced ourselves to L_∞ in order to maintain the homogenity of the two quantities being weighted. More specifically, assume that the finite horizon L_1 norm, which computes the area under the error signal until a given point T, is used for evaluating the signals before both of them reach steady state and that the L_∞ norm (steady-state offset, in this case) is used for evaluating them after they reached steady state. Even if both norms are R_+ valued, computing an area is qualitatively different than computing a maximum deviation. This fact would positively bias the importance that the prior-steady-state behavior has on the norm measure.
All the monotone examples used in this thesis reached steady state, therefore they were evaluated through the stability norm as well.

Definition 3.1.9. *Let $||\cdot||$ be a signal norm, $||\cdot|| : \mathbf{V} = (\mathbb{R}^n, \mathbb{R}) \to \mathbb{R}_+$. w_1 and $w_2 \in \mathbb{R}_+$ such that $w_1 + w_2 = 1$. Let $T \in \mathbb{R}_+$. The function $||\cdot||_S : V \to \mathbb{R}_+$ is defined as such, $\forall \mathbf{v} \in \mathbb{R}^n$:*

$$||\mathbf{v}||_S := w_1 * ||\mathbf{v}||_{[0,T]} + w_2 * ||\mathbf{v}||_{[T,\infty]} \tag{3.9}$$

Proposition 3.1.10. $||\mathbf{v}||_S : V \to \mathbb{R}_+$ *is a* seminorm *on the vector space* $(\mathbb{R}^n, \mathbb{R})$

Proof. Let $\mathbf{v}, \mathbf{x} \in \mathbb{R}^n$ and $a \in \mathbb{R}$. Expression 3.9 satisfies all the requirements of a norm:

1. The positive homogenity property is fully satisfied:
   $||a \cdot \mathbf{v}||_S = w_1 \cdot ||a \cdot \mathbf{v}||_{[0,T]} + w_2 \cdot ||a \cdot \mathbf{v}||_{[T,\infty]} = a \cdot w_1 \cdot ||\mathbf{v}||_{[0,T]} + a \cdot w_2 * ||a\mathbf{v}||_{[T,\infty]} = a \cdot ||\mathbf{v}||_S$

2. The triangle inequality if fully satisfied:
   $||\mathbf{v} + \mathbf{x}||_S = w_1 \cdot ||\mathbf{v} + \mathbf{x}||_{[0,T]} + w_2 \cdot ||\mathbf{v} + \mathbf{x}||_{[T,\infty]} \leq w_1 \cdot (||\mathbf{v}||_{[0,T]} + ||\mathbf{x}||_{[0,T]}) + w_2 \cdot (||\mathbf{v}||_{[T,\infty]} + ||\mathbf{x}||_{[T,\infty]}) = (w_1 \cdot ||\mathbf{v}||_{[0,T]} + w_2 \cdot ||\mathbf{v}||_{[T,\infty]}) + w_2(||\mathbf{x}||_{[0,T]} + ||\mathbf{x}||_{[T,\infty]}) = ||\mathbf{v}||_S + ||\mathbf{x}||_S$

3. The separating points is not fully satisfied:
   $||a \cdot \mathbf{v}||_S = \mathbf{0} \Rightarrow w_1 \cdot ||\mathbf{v}||_{[0,T]} + w_2 \cdot ||\mathbf{v}||_{[T,\infty]} = \mathbf{0}$. As $w_1 + w_2 = 1 \Rightarrow$ at least one of w_1 and w_2 is $\neq 0$. Without loss of generality, assume therefore that $w_1 \neq 0$. Dividing by $w_1 \Rightarrow ||\mathbf{v}||_{[0,T]} + \frac{w_2}{w_1} \cdot ||\mathbf{v}||_{[T,\infty]} = \mathbf{0} \Rightarrow ||\mathbf{v}||_{[0,T]} = \mathbf{0}$ and $w_2 = 0$ or $||\mathbf{v}||_{[T,\infty]} = \mathbf{0}$. In case $||\mathbf{v}||_{[T,\infty]} = \mathbf{0} \Rightarrow \mathbf{v} = \mathbf{0}$. In case $w_2 = 0$, then $w_1 = 1$ and $||\mathbf{v}||_S = ||\mathbf{v}||_{[0,T]}$.

□

When being applied to evaluating the performance of monotone systems approximations, the positive parameter T was set to be the time when the monotone system reaches steady state (in case it does).

3.1.6 Discretization

As the performance evaluation of the approximation algorithms was done computationally, the theoretical measures defined in this chapter were adjusted to fit into the computational framework. Two qualitatively different distinctions should be made between the theoretical definition and the practical computation of the above-defined norms:

1. In the computational environment, all the norms were evaluated on a finite time horizon. This makes the values of all norms dependent on the maximal time T chosen in definition 3.1.4.

2. In the computational environment, signals were represented as time-dependent vectors. In order to maintain the equivalence between the theoretical finite time horizon norms and their computational approximation, certain requirments needed to be fulfilled:

   - The discretization of the signal has to be an accurate enough representation of the continuous signals. Therefore, the sampling needs to be done with a high - enough frequency.
   - The signals need to be comparable, i.e. the vector spaces on which the time-horizon norms are defined should have the same dimensions. Therefore, the finite time horizon needs to be kept constant between signals to be compared and, in the same time, the sampling needs to be done at equal intervals.

Moreover, a widely encountered situation is the one in which the system doesn't actually reach steady state but, instead, it asymptotically converges towards it. In practice, we would in this case consider that the system did reach its steady state, as soon as its output doesn't change with more than a pre-defined error bound for longer than a pre-defined time bound.

3.2 System Norms

Unlike the signal norms, which provide information on the closeness of the systems' outputs produced under any chosen input, the system norms provide information on the systems' output produced either under a specific input (the H_2 norm) or under a wider input range (the H_∞ norm). Therefore, the signal norms are input-dependent, whereas the system norms are not.

We will restrict our discussion to SISO (single input/single output) LTI systems. For a generalization of the system norms to the MIMO (multiple input/multiple output) case, see [14].

3.2.1 H_2 - norm

Stochastic Interpretation of the H_2-norm

From a stochastic system perspective, the H_2-norm of the LTI system with the state space representation 1.4 and with transfer function matrix $\mathbf{G}$ equals the root-mean-square (RMS) of the impulse response of the system. It measures the steady-state covariance (or power) of the output response to an unit variance white noise input ([14]). More specifically, if $y(t) = (g_c * u)(t)$ and the input u is:

$$u(t) = \begin{cases} \text{a unit variance white noise process, } t \in [0, T] \\ 0, \text{ otherwise} \end{cases} \tag{3.10}$$

then the *finite - horizon H_2-norm* of the system $\mathbf{G}$ is defined by:

Definition 3.2.1.

$$||\mathbf{G}||^2_{2,[0,T]} = \varepsilon \left\{ \frac{1}{T} \int_0^T y^H(t) y(t) dt \right\} \tag{3.11}$$

where ε is the expectation operator and y^H represents the Hermitian (conjugate transpose) matrix of the output matrix y.

Substituting

$$y(t) = \int_0^\infty g_c(t - \tau) u(\tau) d\tau$$

into equation 3.11 and using the statistical properties of the unit variance white noise ([14]), one obtains that:

$$||\mathbf{G}||^2_{2,[0,T]} = \int_{-T}^{T} \text{tr}\left(g_c(t) g_c^H(t) \right) dt - \frac{1}{T} \int_0^T \text{tr}\left(g_c(t) g_c^H(t) + g_c(-t) g_c^H(-t) \right) t dt$$

If the impulse response matrix $g_c(t)$ of the system (relation 1.2.1) is such that the integrals remain bounded as $T \to \infty$, the following definition makes sense:

Definition 3.2.2. *The* infinite-time horizon H_2-norm *of the system* $\mathbf{G}$ *is*

$$||\mathbf{G}||_2^2 = \int_{-\infty}^{\infty} \text{tr}\left(g_c(t) g_c^H(t) \right) dt = \int_{-\infty}^{\infty} ||g_c(t)||_F dt \tag{3.12}$$

Parseval's theorem states the equivalence between the 2-norm of a signal in the time domain and its 2-norm in the frequency domain ([14]). By applying this theorem to the system's impulse response $g_c(t)$ and to its transfer function $\mathbf{G}$ (which is the Laplace transform of the impulse response; see equation 1.13), we obtain that:

$$||\mathbf{G}||_2^2 = \int_{-\infty}^{\infty} ||\mathbf{G}(j\omega)||_F d\omega = \int_{-\infty}^{\infty} \text{tr}\left(\mathbf{G}(j\omega) \mathbf{G}^H(j\omega) \right) d\omega \tag{3.13}$$

where $\mathbf{G}(j\omega)$ is the Fourier transform of $g_c(t)$.

State-space computation of the H_2 norm

As described in [35], rather than evaluating the integrals 3.12 or 3.13 directly, the H_2 norm of a transfer function can be computed on the basis of the state-space representation of the LTI system $\mathbf{G}$. However, in order to do this, additional constraints need to be imposed on the form of the system.

Proposition 3.2.3. *Let* $\mathbf{G}$ *be a LTI system with a state-space representation of the form 1.4.* $||\mathbf{G}||_2$ *is finite iff* $\mathbf{G}$ *is stable and* $D = 0_{p \times m}$

For a complete proof of the above proposition, see [14].

As we are only interested in finite values of $||\mathbf{G}||_2$, let from now on $\mathbf{G}$ be a stable LTI system, with the D matrix equally vanishing.
By Parseval's theorem and using the expression 1.2.1, we have that:

$$||\mathbf{G}||_2^2 = ||g_c||_2^2 = \text{tr}\left[C \int_0^\infty e^{At} B B^T e^{A^T t} dt \; C^T\right] \tag{3.14}$$

where X^T represents the transpose matrix of X. Defining the matrix

$$W_c := \int_0^\infty e^{At} B B^T e^{A^T t} dt \tag{3.15}$$

we can re-write equation 3.14 as:

$$||\mathbf{G}||_2^2 = ||g_c||_2^2 = \text{tr}\left[C W_c C^T\right]$$

Similarly, by defining the following matrix:

$$W_o := \int_0^\infty e^{A^T t} C^T C e^{At} dt \tag{3.16}$$

it can be shown that an equivalent expression for $||\mathbf{G}||_2^2$ is:

$$||\mathbf{G}||_2^2 = ||g_c||_2^2 = \text{tr}\left[B^T W_o B\right]$$

The matrices W_c and W_o are called the *controllability* and *observability* Gramians of the LTI system $\mathbf{G}$, with state-space representation 1.4. They also represent the unique solutions to the following linear matrix equations, also known as *matrix Lyapunov equations*:

$$A W_c + W_c A^T + B B^T = 0 \qquad W_o A + A^T W_o + C^T C = 0 \tag{3.17}$$

For a complete proof on the identities 3.17, refer to [14].

Signal Interpretation of the H_2 norm

As described in [35], for an explicit input/output interpretation of the H_2 norm, recall its transfer function representation (see equation 1.12):

$$\mathbf{Y}(s) = \mathbf{G}(s) * \mathbf{U}(s)$$

Assuming a SISO system, suppose that the Laplace transform $\mathbf{U}(s)$ of the input $u(t)$ contains equal ammounts of all frequencies, i.e.:

$$\mathbf{U}(s) = 1$$

Then,

$$||\mathbf{Y}(s)||_2 = \mathbf{G}(s) = \Big(\int_{-\infty}^{\infty} ||\mathbf{G}(j\omega)||_F^2 d\omega \Big)^{1/2}$$

In this case, the interpretation of the $||\mathbf{G}||_2$ is an average system gain over all frequencies or the average increase in the output of the system, due to its input, over all frequencies.

3.2.2 H_∞ - norm

The ∞-norm of the transfer function matrix $\mathbf{G}(s)$ provides a measure of the worse-case system gain.

Definition 3.2.4. *Provided that the system* $\mathbf{G}$ *is stable, then its* H_∞ *norm is:*

$$||\mathbf{G}||_\infty := \sup_\omega ||\mathbf{G}(j\omega)||_F \tag{3.18}$$

$||\mathbf{G}(j\omega)||_F$ is the factor by which the amplitude of a sinusoidal input with angular frequency ω is magnified by the system. Therefore, the H_∞-norm represents the largest factor by which any sinusoid is magnified by the system ([35]).

An equivalent formulation for the H_∞ norm is:

Proposition 3.2.5.

$$||G||_\infty := \sup_\omega \left\{ \frac{||\mathbf{G}v||_2}{||v||_2}; v \neq 0 \right\} \tag{3.19}$$

Now, let $u(t)$ be a signal with Laplace transform $\mathbf{U}(s)$ such that its time-domain L_2 norm (equation 3.6), as well as its frequency-domain L_2 norm (equation 3.7) are bounded. Then the following implications hold:

$$\begin{aligned} ||\mathbf{GU}||_2 &= \Big(\int_{-\infty}^{\infty} ||\mathbf{G}(j\omega)\mathbf{U}(j\omega)||_F^2 d\omega \Big)^{1/2} \\ &= \Big(\int_{-\infty}^{\infty} ||\mathbf{G}(j\omega)||_F^2 ||\mathbf{U}(j\omega)||_F^2 d\omega \Big)^{1/2} \\ &\leq \sup_\omega ||\mathbf{G}(j\omega)||_F \left(\int_{-\infty}^{\infty} ||\mathbf{U}(j\omega)||_F^2 d\omega \right)^{1/2} \\ &= ||\mathbf{G}||_\infty ||\mathbf{U}||_2 \end{aligned} \tag{3.20}$$

And, as a consequence:

$$||\mathbf{G}||_\infty \geq \frac{||\mathbf{GU}||_2}{||\mathbf{U}||_2}, \qquad \forall \mathbf{U} \neq 0 \tag{3.21}$$

There exists signals $\mathbf{U}$ which come arbitrarily close to the upper bound in equation 3.20 (see [35] for a more detailed discussion on the matter). Therefore, an equivalent definition of the H_∞ norm is the following:

$$||\mathbf{G}|| = \sup \left\{ \frac{||\mathbf{GU}||_2}{||\mathbf{U}||_2}, \quad \mathbf{U} \neq 0 \right\} \tag{3.22}$$

Note: a generalization of the notion of norm for systems exists in the linear time - varying case, as well as in the nonlinear case. This result is known as *small gain theorem* and the size through which the norm of the system will be measured is called *incremental gain*. For more details on the incremental gain norm, see [14].

Chapter 4

Monotone Examples

A necessary step in evaluating the approximation methods on monotone systems was applying them on concrete examples. These systems were chosen such that they span over a wide range of features, i.e. they have different number of species or their nonlinearities have different forms. All examples were taken from the existent literature on I/O monotone systems and they are representations of different biological processes. The parameter vectors of the systems (kinetic rates) were kept constant. However, the systems were simulated under (mainly constant) inputs ranging over several orders of magnitude.
In the monotone systems analysis (especially in the Sontag framework, see [13],[32], [6], [33]), one often starts with uncontrolled dynamical systems of the form 1.1. Such systems are transformed to the controlled form 1.2 by the replacement of all the appearances of (at least) one of the system's states in the rate of change of (at least one) other state by input channels. Automatically, the replaced variables becomes the output channels of the new I/O system. This procedure, which we will exemplify on the monotone systems presented in this chapter, is informally refered to as *breaking the loop*. For formal definitions on the matter, see [12].

4.1 Example1

Example1 was initially introduced by Mischaikow et al. in [9] and then also studied, in a slightly simplified form, by Enciso in [12]. The system models the messenger RNA transcription and translation processes in an unicellular organism (yeast). More specifically, the organism produces a protein that crosses the nuclear membrane and promotes the further production of its own messenger RNA ([12]). The version which we will present here is almost identical to Enciso's, with one additional simplification: the influence of the additional protein λ on the messenger RNA transcription was neglected.

The ODE form of the system is the following:

$$\begin{cases} \dfrac{\mathrm{d}p}{\mathrm{d}t} = -K_e p(t) + K_i q(t) - a_2 p(t) \\ \dfrac{\mathrm{d}q}{\mathrm{d}t} = K_e p(t) - K_i q(t) - a_3 q(t) + T(r(t)) \\ \dfrac{\mathrm{d}r}{\mathrm{d}r} = H(p(t)) - a_1 r(t) \end{cases} \tag{4.1}$$

where r is the mRNA concentration, p is the concentration of the intracellular protein and q is the concentration of the extracellular protein. The functions H and T represent the transcription and translation rates and should be chosen such that:

$$\frac{\partial H(p(t))}{\partial p(t)} > 0 \text{ and } \frac{\partial T(r(t))}{\partial r(t)} > 0$$

The constants a_1, a_2 and a_3 represent the degradation coefficients, while K_i and K_e are the rates of import and export of protein through the nuclear membrane. All these constants need to be positive.

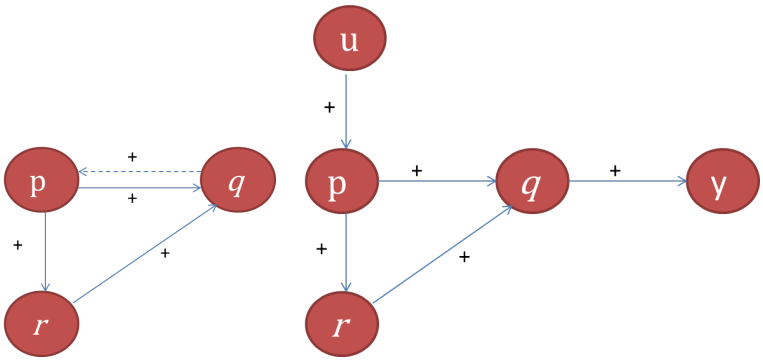
In order to transform the system 4.1 to an I/O form, the influence of the variable q on the variable p is replaced with an external influence, the input $u : \mathbb{R}_+ \to \mathbb{R}$, which can be experimentally manipulated. The loop is closed again by setting the output $y : \mathbb{R}_+ \to \mathbb{R}$ equal to the variable whose influence was ignored, q (see figure 4.1). The controlled system has the following ODE form:

$$\begin{cases} \dfrac{\mathrm{d}p}{\mathrm{d}t} = K_i u(t) - K_e p(t) - a_2 p(t) \\ \dfrac{\mathrm{d}q}{\mathrm{d}t} = K_e p(t) - K_i q(t) - a_3 q(t) + T(r(t)) \\ \dfrac{\mathrm{d}r}{\mathrm{d}r} = H(p(t)) - a_1 r(t) \\ y(t) = q(t) \end{cases} \tag{4.2}$$

As the system is defined on positive orhatnt cones, its monotonicity can be established by using the graph-theoretical criterium (proposition 2.2.14). Indeed, as it can be seen in figure 4.1, the species graph of the system does not contain any negative cycles.

For any fixed control (input) $\bar{u}$, the system reaches an unique globally asymptotically stable equilibrium ([12]). The mapping $\bar{u} \to (p(\bar{u}), q(\bar{u}), r(\bar{u}))$ represents the input-to-state characteristic of the system (definition 2.3.2; for a detailed discussion on the monostability of this steady state, see [12]):

$$p(\bar{u}) = \frac{k_i \bar{u}}{k_e + a_2} \tag{4.3}$$



(a) Uncontrolled System (b) I/O System

Figure 4.1: The species graphs for the uncontrolled system 4.1 and the controlled one 4.2.
The dashed line denoting the influence of the variable q on the variable p in subfigure 4.1a is replaced by an external influence from input u in subfigure 4.1b, transforming the system from an autonomous form to an I/O one.

$$q(\bar{u}) = \frac{1}{k_i + a_3} T\left(\frac{1}{a_1} H\left(\frac{k_i}{k_e + a_2}\bar{u}\right)\right) + \frac{k_i}{k_e + a_2}\frac{k_e}{k_i + a_3}\bar{u} \tag{4.4}$$

$$r(\bar{u}) = H\left(\frac{k_i \bar{u}}{(k_e + a_2)\, a_1}\right) \tag{4.5}$$

The mapping $\bar{u} \to y(\bar{u}) = q(\bar{u})$ in equation 4.4 represents the input-to-output characteristic of the system (definition 2.3.3).

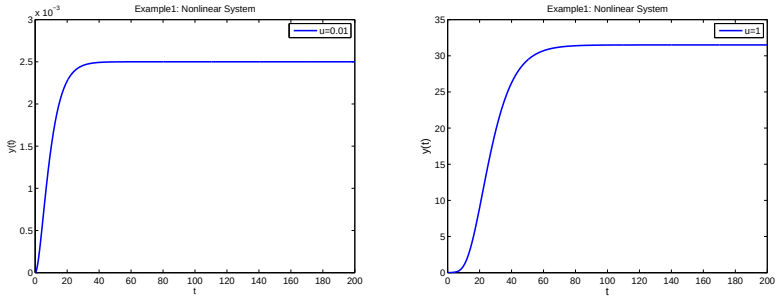
In order to simplify the analysis of the system, specific function forms for the transcription and translation rates were chosen:

$$H(x) = x^2 \text{ and } T(x) = x^2 \tag{4.6}$$

As both H and T were chosen to be quadratic functions, equation 4.4 will show a fourth-order dependency output/input. The qualitative behavior of the output $q(\bar{u})$ remains the same, regardless of the value of the constant input $\bar{u}$. However, the value of the output scales with the one of the input, by a factor which can be derived from the actual dependency 4.4 (see figure 4.2).

4.2 Example2

The second example also belongs to Enciso [12] and represents a coupled biological circuit. In its uncontrolled form, the version presented here is identical to the one in [12], while the controlled version is more simplified, as the resulting controlled system is SISO, whereas in [12] it is MIMO (2 input/output channels). This simplification however brings important changes to the dynamics of



(a) Input: $u = 0.01$ (b) Input: $u = 1$

Figure 4.2: The simulated output of system 4.1, for the following parameter values: $a_1 = a_2 = a_3 = K_i = K_e = 0.1$ and initial condition $(0, 0, 0)$. Total simulation time: 200 seconds. The qualitative behavior of the system is the same, regardless of the input value.

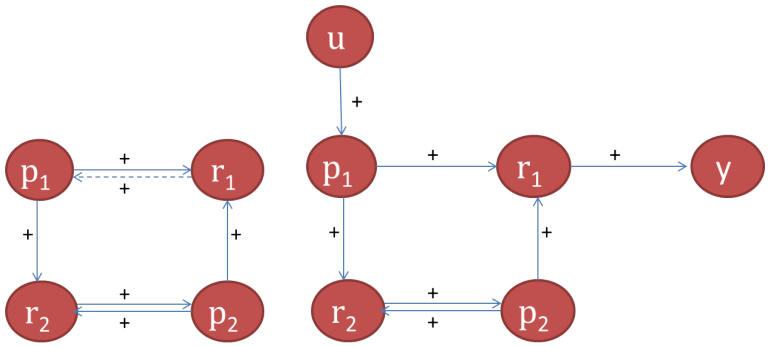
the system, as its I/O form is not endowed with a input-to-state characteristic anymore.
The biological circuit describes the messenger RNA transcription, a process generally regulated by (in principle more than one) *transcription factors*. The transcription factors promote (or inhibit) the binding of the enzyme RNA polymerase to the DNA sequence. An *autoregulatory transcription factor* regulates the production of its own messenger RNA.
We will assume here two autoregulatory transcription factors p_1 and p_2 and their corresponding messenger RNAs, r_1 and r_2. The two transcription processes will be coupled through the assumption that the transctiption of each messenger RNA is regulated by both transctiption factors. Hence, the temporal dynamics of the circuit is represented through the following uncontrolled system consisting of 4 ODEs, for $i \in 1, 2$:

$$\begin{cases} \dfrac{\mathrm{d}p_i(t)}{\mathrm{d}t} = a_i r_i(t) - b_i p_i(t) \\ \dfrac{\mathrm{d}r_i(t)}{\mathrm{d}t} = g_i(p_1(t), p_2(t)) - c_i r_i(t) \end{cases} \tag{4.7}$$

The functions $g_i(p_1, p_2)$ represent the transcription rates and should be chosen such that, $\forall i \in \{1, 2\}$:

$$\frac{\partial g_i(t)}{\partial p_i(t)} \geq 0,\ g_i(p_1(t), p_2(t)) \geq 0 \text{ and } \exists M_i \geq 0 \text{ s.t. } g_i(p_1, p_2) \leq M_i$$

The boundness of the functions $g_i(p_1, p_2)$ bounds the solutions r_i as $t \to \infty$ which, in turn, bound the solutions p_i. Hence, the boundness requirement of



(a) Uncontrolled System (b) I/O System

Figure 4.3: The species graphs for the uncontrolled system 4.7 and the controlled one 4.8.
The dashed line denoting the influence of the variable r_1 on the variable p_1 in subfigure 4.3a is replaced by an external influence from input u in subfigure 4.3b, transforming the system from an autonomous form to an I/O one.

the functions $g_i(p_1, p_2)$ (which is always satisfied in biochemical networks, as rate laws are always bounded) ensures the boundness of the solutions to the system 4.7.

In order to transform the uncontrolled system into a SISO I/O one, the influence of the variable r_1 on the variable p_1 is replaced by an input $u : \mathbb{R}_+ \to \mathbb{R}$, while the replaced variable r_1 becomes the output of the system $y : \mathbb{R}_+ \to \mathbb{R}$ (see figure 4.3). The ODE system becomes:

$$\begin{cases} \dfrac{\mathrm{d}p_1(t)}{\mathrm{d}t} &= a_1 u(t) - b_1 p_1(t) \\ \dfrac{\mathrm{d}p_2(t)}{\mathrm{d}t} &= a_2 r_2(t) - b_2 p_2(t) \\ \dfrac{\mathrm{d}r_1(t)}{\mathrm{d}t} &= g_1(p_1(t), p_2(t)) - c_1 r_1(t) \\ \dfrac{\mathrm{d}r_2(t)}{\mathrm{d}t} &= g_2(p_1(t), p_2(t)) - c_2 r_2(t) \\ y(t) &= r_1(t) \end{cases} \tag{4.8}$$

The controlled system is monotone, as can be seen from its species graph (figure 4.3), which doesn't contain negative cycles.

Assuming Hill type kinetics ([18]), the form of the functions g_i is:

$$g_i(p_1, p_2) = \hat{\sigma}_i \frac{p_1^{m_i} p_2^{n_i}}{\hat{K}_i + p_1^{m_i} p_2^{n_i}}$$

The coefficients m_i and n_i describe the cooperativity with which the proteins bind to the DNA sequence. For instance, if, before acting on the DNA sequence of p_1, two p_1 proteins need to bind, then $m_i = 2$. It is therefore a reasonable assumption that the cooperativity of a given protein is the same, regardless of which DNA sequence it binds to: $m_1 = m_2 = m$ and $n_1 = n_2 = n$. Here, in order to ease the calculations, we choose $m = 2$ and $n = 1$, hence:

$$g_i(p_1, p_2) = \hat{\sigma}_i \frac{p_1^2 p_2}{\hat{K}_i + p_1^2 p_2}$$

The remaining coefficients $\hat{K}_i$ and $\hat{\sigma}_i$ are particularly determined by the way in which the proteins bind to the each DNA sequence and how they aid the polymerase enzyme ([12]). For a fixed input $u(t) \equiv \bar{u}$, the system 4.8 admits two steady-states:

$$p_1(\bar{u}) = \frac{a_1}{b_1}\bar{u} \tag{4.9}$$

$$p_2(\bar{u}) = \frac{\hat{\sigma}_2 a_2}{b_2} - \frac{b_1^2 \hat{K}_2}{a_1^2 \bar{u}^2} \tag{4.10}$$

$$r_1(\bar{u}) = \frac{\hat{\sigma}_1}{c_1} \frac{\hat{\sigma}_2 a_2 a_1^2 \bar{u}^2 - b_1^2 b_2 \hat{K}_2}{b_1^2 b_2 (\hat{K}_1 - \hat{K}_2) + \hat{\sigma}_2 a_2 a_1^2 \bar{u}^2} \tag{4.11}$$

$$r_2(\bar{u}) = \hat{\sigma_2} - \frac{\hat{K}_2 b_1^2 b_2}{a_1^2 a_2 \bar{u}^2} \tag{4.12}$$

and

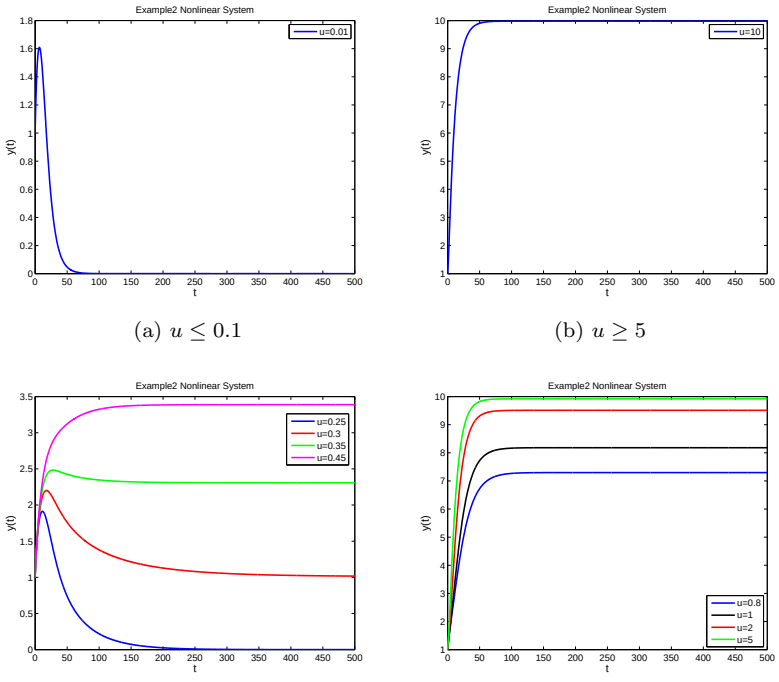
$$p_1(\bar{u}) = \frac{a_1}{b_1}\bar{u} \tag{4.13}$$

$$p_2(\bar{u}) = 0;\ r_1(\bar{u}) = 0;\ r_2(\bar{u}) = 0 \tag{4.14}$$

The system toggles between these two steady states, depending on the input value and on the initial condition. In our case however, for the fixed parameter vector used, the system showed three types of qualitatively different behavior (see figure 4.4).

4.3 Example3

Example3 reprsents a two-variable Cdc2 Wee system ([28], [8]) and it is studied by Angeli, Ferrell and Sontag in [1] as an example of monotone bistable system. As here we are not concerned with studying the bistability of the system, we



(a) $u \leq 0.1$

(b) $u \geq 5$

(c) $u \in (0,1;0.5)$

(d) $u \in [0.5;5)$

Figure 4.4: The simulated output for system 4.8, for the following parameter values: $a_1 = a_2 = b_1 = b_2 = c_1 = c_2 = 0.1$, $\hat{\sigma}_1 = \hat{\sigma}_2 = \hat{K}_2 = 1, \hat{K}_1 = 2$ and initial condition $(1,1,1,1)$. Total simulation time: 500 seconds. For inputs $u \leq 0.1$, the output is invariant to the input and its qualitative behavior is the one from subfigure 4.4a. For inputs $u \geq 5$, the output reaches saturation and it becomes invariant to the input as well, as it can be seen in subfigure 4.4b. Subfigure 4.4c shows the qualitatively different outputs for inputs in the range $(0,1;0.5)$ and subfigure 4.4d shows the qualitatively similar outputs when the input is in the range of $[0.5;5)$, scaling with the input value until the output reaches saturation.

neglect the extra parameter v which Angeli uses in [1] for representing the gain of the system and for rendering the system either monostable or bistable. More exactly, here, this parameter is set to 1. In order to model the interaction between the two proteins as a two-variable system, it is assumed that both Wee1 and Cdc2 exist in only two forms (rather than multiple forms, as is actually the case): an active and an inactive one. The active forms will be denoted by x_1 and z_1 and the inactive ones by x_2 and z_2, respectively. Second, it is assumed as well that the phosphorylations of Cdc2 and Wee1 are reversed by some constitutively active phosphatases, which ignores the contribution of Cdc25 regulation to the bistability of the Cdc2 system. Finally, the inhibition of each kinase by the other is approximated by a Hill equation. ([1]).

The equations for this model system are:

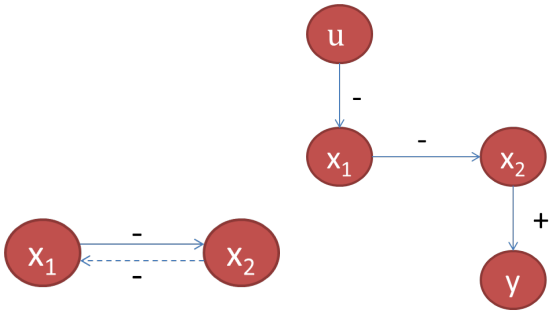
$$\begin{cases} \dfrac{\mathrm{d}x_1(t)}{\mathrm{d}t} &= a_1 z_1(t) - \frac{\beta_1 x_1 x_2^{\gamma_1}}{K_1 + x_2^{\gamma_1}} \\ \dfrac{\mathrm{d}z_1(t)}{\mathrm{d}t} &= -a_1 z_1(t) + \frac{\beta_1 x_1 x_2^{\gamma_1}}{K_1 + x_2^{\gamma_1}} \\ \dfrac{\mathrm{d}x_2(t)}{\mathrm{d}t} &= a_2 z_2(t) - \frac{\beta_2 x_2 x_1^{\gamma_2}}{K_2 + x_1^{\gamma_2}} \\ \dfrac{\mathrm{d}z_2(t)}{\mathrm{d}t} &= -a_2 z_2(t) + \frac{\beta_2 x_2 x_1^{\gamma_2}}{K_2 + x_1^{\gamma_2}} \end{cases} \tag{4.15}$$

where α_1 and α_2, β_1 and β_2 are rate constants, K_1 and K_2 are the Michaelis Menten constants and γ_1 and γ_2 are Hill coefficients. If we assume that the total concentration of the proteins remains constant and we express it in fractional terms, i.e. $z_1 = 1 - x_1$ and $z_2 = 1 - x_2$, the dynamics of the system 4.15 is equivalently described by the following 2 ODEs:

$$\begin{cases} \dfrac{\mathrm{d}x_1(t)}{\mathrm{d}t} &= a_1(1 - x_1(t)) - \frac{\beta_1 x_1 x_2^{\gamma_1}}{K_1 + x_2^{\gamma_1}} \\ \dfrac{\mathrm{d}x_2(t)}{\mathrm{d}t} &= a_2(1 - x_2(t)) - \frac{\beta_2 x_2 x_1^{\gamma_2}}{K_2 + x_1^{\gamma_2}} \end{cases} \tag{4.16}$$

In order to transform the closed-loop system 4.17 into an open-loop one, the influence of the variable x_2 on the variable x_1 is replaced by the external influence of the input u. The variable x_2 hence becomes the output of the system (see figure 4.5). The new SISO form is the following:

$$\begin{cases} \dfrac{\mathrm{d}x_1(t)}{\mathrm{d}t} & = a_1(1 - x_1(t)) - \frac{\beta_1 x_1 u^{\gamma_1}}{K_1 + u^{\gamma_1}} \\ \dfrac{\mathrm{d}x_2(t)}{\mathrm{d}t} & = a_2(1 - x_2(t)) - \frac{\beta_2 x_2 x_1^{\gamma_2}}{K_2 + x_1^{\gamma_2}} \\ y(t) & = x_2(t) \end{cases} \tag{4.17}$$



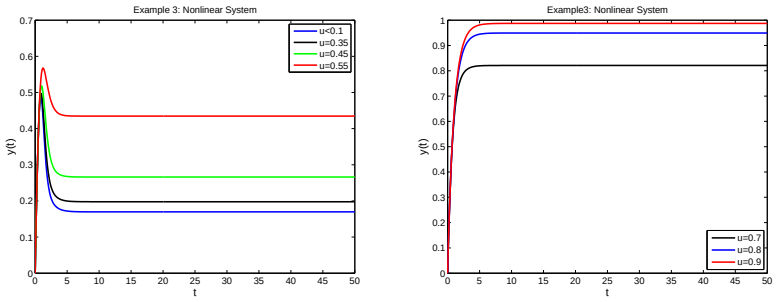
(a) Uncontrolled System (b) I/O System

Figure 4.5: The species graphs for the uncontrolled system 4.16 and the controlled one 4.17.
The dashed line denoting the influence of the variable x_2 on the variable x_1 in subfigure 4.5a is replaced by an external influence from input u in subfigure 4.5b, transforming the system from an autonomous form to an I/O one.

For a fixed input $\bar{u}$, the system reaches an unique equilibirum point $(x_1(\bar{u}), x_2(\bar{u}))$ and is endowed with a static input-to-state characteristic (for a formal proof, see [1]). The corresponding static input-to-output characteristic (definition 2.3.3) is the following:

$$x_2(\bar{u}) = \frac{a_2(K_2 + (a_1(K_1 + \bar{u}^{\gamma_1}))^{\gamma_2})}{a_2 K_2(a_1 K_1 + a_1 \bar{u}^{\gamma_1} + \beta_1 \bar{u}^{\gamma_1}) + (a_2 + \beta_2)(a_1(K_1 + \bar{u}^{\gamma_1}))^{\gamma_2}} \tag{4.18}$$

As in this example the concentrations are represented in fractional terms, the system was simulated for constant inputs $\bar{u} \leq 1$. Essentially, the system shows two qualitatively different types of dynamical behavior (see figure 4.6). The static input-to-output characteristic is an increasing function, as higher steady-state values are reached for higher inputs. The system shows fast convergence to steady state.



(a) Input: $u \leq 0.55$ (b) Input: $0.55 < u \leq 1$

Figure 4.6: The simulated output for system 4.17, for the following parameter values: $a_1 = a_2 = 1$, $\beta_1 = 200$, $\beta_2 = 10$, $\gamma_1 = \gamma_2 = 4$, $K_1 = 30, K_2 = 1$ and initial condition $(0, 0)$. Total simulation time: 20 seconds.

Chapter 5

SI Approach

If seen in the System Identification framework, the present system approximation problem becomes: *"Given multiple (as many as needed) input/output signal pairs, identify the system that produced them."*. In principle, the need for system identification arises when one possesses little or no extra information on the underlying system, other than input/output signal pairs. The available data is fitted to different model structures, gradually increasing in complexity. The performance of the models on new datasets is being evaluated with respect to pre-defined metrics and, generally, the simplest model explaining the test data is selected. Increasing the complexity of the model will unavoidably increase the fit of the test data, therefore definiting a suitable threshold complexity/accuracy is essential in the identification process.
However, the present settings are different than the usual system identification ones. Here, the underlying system is fully known. This automatically generates two main advantages:

1. Knowing the model class of the original system means that certain model families can be automatically discarded, provided that their main characteristics are irrelevant/contradictory to the original mode type.

2. There exists no restriction regarding the type nor quantity of test, as well as validation, data used. Knowing the underlying system allows the generation of as many as desired dataset pairs.

When aiming at approximating monotone systems, this type of approach is interesting also because it can offer insight into their dynamical behavior *per se*. More specifically, the "core characteristics" of monotone systems, which can be identified regardless of the parameter values of the system (deduced from the training input/output pairs, but successfuly extended to new validation datasets), could show preference towards specific model types used for identification. This could open the way to a more in-depth theoretical investigation on the specific relation between these two system structures.

5.1 Experiment Design

Following the same terminology as in [20], we will define *an experiment* as a pair of input/output signals. The proper experiment design is a prior and necessary step to the identification routine. In practice, this step essentially translates into choosing suitable inputs, whose corresponding outputs provide sufficient information to the identification routine, such that the resulting system has predictive capacities. In the present case, different experiments were merged in a single dataset, which was used for identifying the monotone systems presented in Chapter 4.

One first consistent (i.e. valid for all the monotone modules described in this thesis) observation regarding the estimation and validation datasets was that, if the estimation dataset is composed of a reduced number of experiments with constant input, it becomes *not exciting enough* for the identification routine, regardless of the type of validation input (constant or not-constant) used. More specifically, by *not exciting enough* we here mean that the identified models were not scaling well on different validation datasets, while their dynamical behavior remained qualitatively and quantitatively close to the dynamical behavior obtained for the (few) estimation experiments.

A second consistent observation was that, if the estimation dataset consists of experiments with different types of input signals (sum of sine waves, Gaussian noise, binary signals etc) and the validation experiment is produced under a constant input, then the predictive capacity of the model is heavily decreased. For instance, if experiments with inputs sine waves and corresponding sine waves as outputs are used for estimation, the model will, in some cases, predict sine wave response for constant inputs in the validation datasets, even if this should not have been the case. The main reason for this lack of predictive capacity was the overfitting tendency of the approximation routines.

A third observation concerns the validation experiment. Even if, as a rule, the input of the validation experiment was never among the inputs used for estimation, its order of magnitude should be covered by the estimation dataset. In other words, the validation input should not dramatically differ from any of the estimation ones. Note however that, most of the times, this is a neccessary, but far from sufficient, requirement.

Therefore, to conclude, providing suitable estimation and validation datasets, exciting enough, but not too data-rich, is by far a non-trivial task. We will illustrate all these observations with plots, while discussing the performance of the identification algorithms in detail on the specific monotone modules.

As the next chapter is entirely based on approximations of monotone systems using linear systems, we will here focus on the Hammerstein Wiener identification. An extra reason for this choice was that, under the norms defined in Chapter 3 and under carefully chosen pairs of estimation/validation datasets, this model structure generally had a very good behavior in approximating the monotone systems.

5.2 Hammerstein Wiener Models

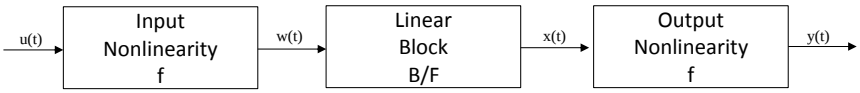


Figure 5.1: Hammerstein Wiener model structure

One approach in identifying nonlinear system is proposing a structure which consists of an interconnection of simpler systems. This is also the case of the *Hammerstein Wiener* model structure ([20]) consisting of three blocks: nonlinear - linear - nonlinear (see figure 5.1). If modeled as a transfer function, the static nonlinearities become input and output signals to the dynamic linear block. By static nonlinearity, we here mean a memoryless system, i.e. a system for which the value of the output at a given time t depends only on the input at that particular time t. We will call these functions *input nonlinearity* and *output nonlinearity*. As also described in the figure 5.1, the input signal $u(t)$ is initially transformed through the input nonlinearity, rendering the output signal $w(t)$, which enters as an input to the linear block. $x(t)$ is the resulting output, which now enters as an input to the output nonlinearity. Its output represents the output of the entire system. If a model only contains the input nonlinearity, it is called a *Hammerstein model*. On the other hand, in case it only contains the output nonlinearity, it is called a *Wiener model*.

5.3 Application

As we saw, the task of identifying a Hammerstein Wiener model structure consists of identifying a linear block, preceded and followed by a nonlinearity. For each of the studied monotone examples, the following default settings were applied:

- In the case of the linear block, the number of poles and zeroes of the transfer function (see [16] for a detailed explanation of these concepts) of each system were set to the corresponding numbers of poles and zero from the transfer function of the linearized version around the steady state of the nonlinear system.
- Both the input and the output nonlinearities were chosen to be piecewise linear.

Essentially, the analysis of the dynamical behavior of the identified system was conducted by varying these three important factors:

- Input;

- Number of units in the input nonlinearity.
- Number of units in the output nonlinearity.

In the case of Example1 (system 4.2), we will graphically reinforce what was also stated in the introductory part of this chapter, namely the importance of the estimation and validation datasets in the identification routine. For the other systems, the following (empirically - derived) rules for estimation will be applied:

- The estimation dataset should contain (a high-enough number of) only constant inputs;
- Inputs with (at least) similar order-of-magnitude to the validation input should be present in the estimation dataset.

5.3.1 Example1

Example1 (system 4.2) was presented in Chapter 4 and its main characteristic was seen to be its qualitatively unchanged output response, regardless of the input value (see figure 4.2).

Figure 5.2 shows (mainly successful) nonlinear Hammerstein Wiener model approximations with varying number of linear units in the input and output nonlinearities, simulated under two different constant input values: 20 (subfigure 5.2a) and 1 (subfigure 5.2b), together with the ranking of these systems under the norms described in Chapter 3 (subfigures 5.2c and 5.2d, respectively). When changing the validation input from 20 to 1, most of the previously identified models dramatically decrease their performance and only some still perform well on the validation input, keeping their scalability (as it can be seen in the two tables). For a better visibility, subfigure 5.2b only shows the models which kept their scalability. However, in the table 5.2d, the performance of all the previously identified Hammerstein Wiener models, for a validation input of 1, can be analyzed.
The models which performed well on both inputs are the ones which have high number of input and output nonlinearity units (at least 5). Moreover, the models which had just 1 input nonlinearity unit constantly perform worse than the others, under all the norms.

Figure 5.3 shows two cases in which the previously well-performing Hammerstein Wiener identified models now perform very unsatisfactory, under all norms, due to changes in either the estimation or the validation dataset. In subfigure 5.3a, the validation dataset is a very complex one, consisting of signals of different types. The Hammerstein Wiener nonlinear model with 10 input and 10 output nonlinearities, estimated on this complex dataset, performs very poorly under the validation input $u = 1$ (compare with figure 5.2, where the same model, estimated on a different dataset, rendered very good results). As we only wanted to make here a visual claim, the performance of this approximated system under the defined norms was not computed. On the other hand,

subfigure 5.3b shows how all the approximations failed if validated under the input 0.01, even if inputs of this range existed in the estimation dataset. The models which previously worked well on the inputs 20 and 1 prove no scalability for such a small input. However, this is also an evidence of the internal equivalence of these models: even if they fail to accurately predict the right output for an input of 0.01, they still perform the same. Following the same reasoning as in subfigure 5.3a, the actual performance of the poorly performing models under the considered norms is not computed, hence not in the table.

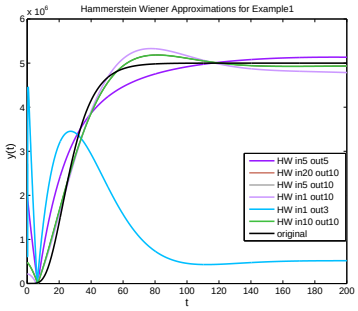
Therefore, we can conclude that, in the case of system 4.2, Hammerstein Wiener is a suitable model structure, which works very well (under all the norms) for some parts of the input space, where for other parts it fails dramatically. It is also worth noticing that, in principle, the used norm measures were consistent between themselves (models performing well/poorly, generally did so under all norms).

5.3.2 Example2

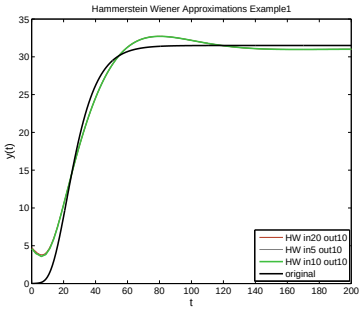
Example2 (system 4.8) was presented in Chapter 4 and it was seen to show 3 types of qualitatively different dynamical behavior, depending on the input value (see figure 4.4). The Hammerstein Wiener model identification routine rendered, in principle, very good results when being applied to this system. Even if, for estimation, inputs from all the 4 ranges described in figure 4.4 were given, only 2 of these qualitatively different behaviors were kept by the Hammerstein Wiener models(see figures 5.4 and 5.5). Hence, as expected, the identified model is less specific than the original one.

As one can see in more detail in figure 5.4, for an input of 0.1, the dynamical behavior of the system is generally well approximated. For an input of 0.4 however, even if the output of the Hammerstein Wiener approximation is, in most cases, quantitatively close to the original output, its qualitative behavior is different and more similar to the output corresponding to an input between 0.2 and 0.3 (see again 4.4). Same phenomenon can be seen for an input of 0.8 5.5, in which the output of some of the identified Hammerstein Wiener models tend to reproduce the qualitative behavior of higher valued inputs such as 2 or 4, while others are are qualitatively similar to outputs produced under inputs ≤ 0.4. However, it has to be mentioned that these differences are mostly qualitative. Quantitatively, the approximated trajectories stay generally very close to the original ones.

Hence, we can conclude that the identified models are not responsive to very small changes in the input. This happens mostly for inputs close to the border between different qualitative behaviors. This is the reason why, for this system, all the identified Hammerstein Wiener approximations performed better on estimation inputs corresponding to wider ranges of qualitatively similar dynamical behavior, such as $u \leq 0.1$ and $u \geq 5$.
Also, by analyzing the ranking tables, no consistent conclusion about which specific Hammerstein Wiener structure is more suitable for this system can be drawn, as the quality of any specific model changes dramatically with changing



(a) input $u = 20$



(b) input $u = 1$

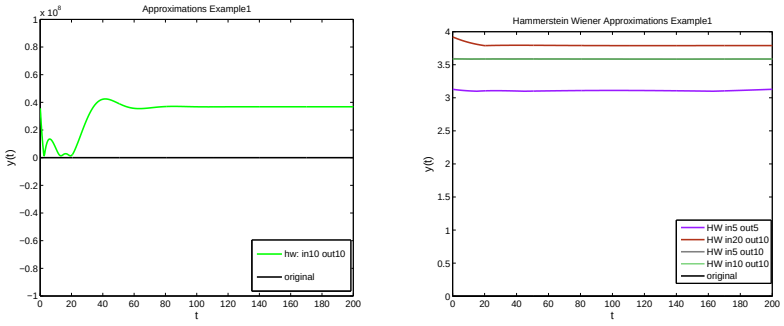
	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25
NHW in5 out5	5.1079e+007	1.7234e+006	1.9853e+006	2.5539e+005	5.9489e+005	1.5632e+005
NHW in20 out10	2.4799e+007	7.0273e+005	4.8075e+005	1.2399e+005	1.6056e+005	3.2019e+005
NHW in5 out10	2.4799e+007	7.0273e+005	4.8075e+005	1.2399e+005	1.6056e+005	3.2019e+005
NHW in1 out10	3.6419e+007	9.1400e+005	3.5570e+005	1.8210e+005	9.6678e+004	1.1926e+005
NHW in1 out3	7.1614e+008	1.7174e+007	4.5694e+006	3.5807e+006	1.4271e+006	3.1424e+006
NHW in10 out10	2.4799e+007	7.0273e+005	4.8075e+005	1.2399e+005	1.6056e+005	3.2019e+005

(c) Performance evaluation for input $u = 20$

	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25
NHW in5 out5	323.5387	10.9171	12.5701	1.6177	3.1425	9.4276
NHW in20 out10	189.8595	5.8995	4.7463	0.9493	1.1966	3.5597
NHW in5 out10	185.5285	5.7312	4.6029	0.9276	1.1866	3.6587
NHW in1 out10	4.0812e+005	1.0596e+004	4.8309e+003	2.0406e+003	1.2077e+003	3.6232e+003
NHW in1 out3	2.4696e+006	5.9986e+004	1.6193e+004	1.2348e+004	4.0482e+003	1.2145e+004
NHW in10 out10	186.5107	5.7408	4.5952	0.9326	1.1488	3.4464

(d) Performance evaluation for input $u = 1$

Figure 5.2: Analysis of successfully identified Hammerstein Wiener model structures for system 4.2, simulated under the following parameter values:$a_1 = a_2 = a_3 = K_i = K_e = 0.1$ and initial condition $(0, 0, 0)$. Total simulation time: 200 seconds. For subfigures 5.2a and 5.2b, the simulated outputs of different Hammerstein Wiener(HW) identified models are shown: *in* represents the number of units in the input nonlinearity, while *out* represents the number of units in the output nonlinearity. In the tabels 5.2c and 5.2d, these models are evaluated, by being compared to the output of the system 4.2. *Stab norm* is the stability from definition 3.1.9, with different weight values.



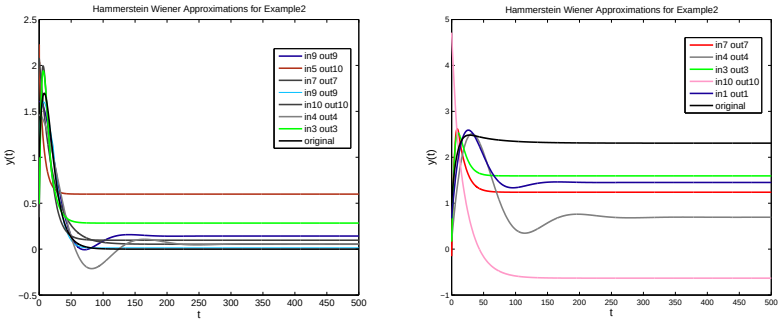
(a) input $u = 20$; complex estimation data (b) input $u = 0.01$

Figure 5.3: Comparison of simulated output for Hammerstein Wiener identified models for system 4.2, simulated under the following parameter values:$a_1 = a_2 = a_3 = K_i = K_e = 0.1$ and initial condition $(0, 0, 0)$. Total simulation time: 200 seconds. *in* represents the number of units in the input nonlinearity, while *out* represents the number of units in the output nonlinearity. In subfigure 5.3a, the estimation was pefromed on a complex dataset, consisting of signals of different types, while in subfigure 5.3b, all the approximations previously performing well in figure 5.2, on different inputs, failed under the input 0.01

input.

5.3.3 Example3

Example2 (system 4.8) was presented in Chapter 4 and it was seen to show 2 types of qualitatively different dynamical behavior, depending on the input value (see figure 4.6). The Hammerstein Wiener model identification routine rendered, in principle, very good results when being applied to this system. As it can be seen in figure 5.6, the two behaviors are kept, together with the corresponding input ranges. Empirically, on these two cases, we can state that the models with 10/10 an 5/5 input/output nonlinearity units perform better than the others.



(a) input $u = 0.1$ (b) input $u = 0.4$

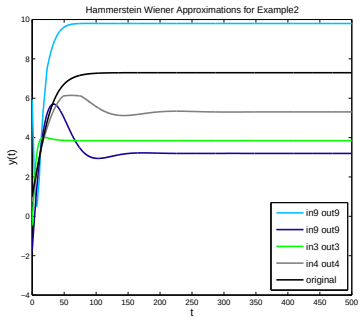
	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25
NHW in9 out9	8.7053	0.1695	0.1909	0.0435	0.0730	0.1516
NHW in4 out4	40.5203	0.6956	0.4598	0.2026	0.2843	0.4013
NHW in10 out10	36.4355	0.7145	1.0826	0.1822	0.3674	0.8442
NHW in7 out7	50.8227	0.7826	0.6533	0.2541	0.2362	0.5143
NHW in1 out1	64.0903	0.9584	0.6027	0.3205	0.2687	0.4914
NHW in3 out3	133.7252	1.9236	0.5106	0.6686	0.3402	0.4538

(c) Performance evaluation for input $u = 0.1$

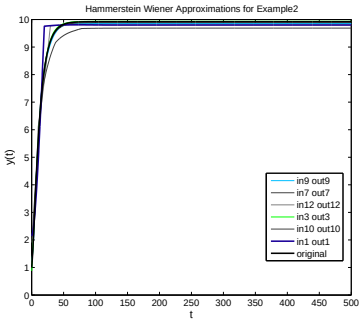
	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25
NHW in9 out9	8.9704e+003	129.4500	19.2200	44.8521	19.1109	18.8927
NHW in4 out4	737.5774	10.9163	1.9890	3.6879	1.7167	1.1723
NHW in10 out10	1.4040e+003	20.1033	3.7149	7.0202	3.1402	3.5233
NHW in7 out7	523.4204	7.4781	1.1551	2.6171	1.1477	1.1527
NHW in1 out1	396.5621	5.8278	1.0155	1.9828	0.8591	0.5465
NHW in3 out3	352.0147	5.0286	0.8383	1.7601	0.8108	0.8291

(d) Performance evaluation for input $u = 0.4$

Figure 5.4: Analysis of identified Hammerstein Wiener models for system 4.8, with parameters: $a_1 = a_2 = b_1 = b_2 = c_1 = c_2 = 0.1$, $\hat{\sigma}_1 = \hat{\sigma}_2 = \hat{K}_2 = 1, \hat{K}_1 = 2$ and initial condition $(1, 1, 1, 1)$, simulated for 500 seconds. Subfigures 5.4a and 5.4b show the outputs of the HW models;*in* and *out* are the numbers of input and output nonlinearity units. The tabels 5.4c and 5.4d show the similarity scores between these outputs and the original one.



(a) input $u = 0.8$



(b) input $u = 5$

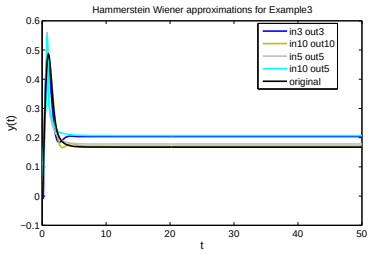
	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25
NHW in9 out9	1.2463e+003	17.7673	4.9333	6.2315	3.4643	4.4436
NHW in4 out4	873.8296	12.9760	2.1751	4.3691	1.7572	0.9213
NHW in10 out10	1.1646e+003	16.6123	3.1424	5.8228	2.6597	2.9815
NHW in7 out7	1.2426e+003	17.7214	3.2883	6.2129	2.9329	3.1698
NHW in1 out1	1.8555e+003	27.3020	4.3317	9.2775	3.9423	3.1635
NHW in3 out3	1.6116e+003	23.2983	3.4516	8.0581	8.0581	2.6559

(c) Performance evaluation for input $u = 0.8$

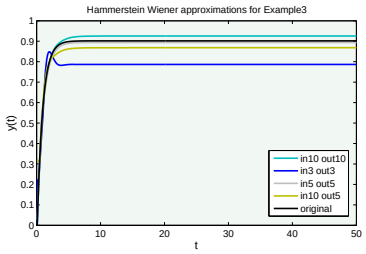
	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25
NHW in9 out9	32.8219	0.4754	0.2389	0.1641	0.1094	0.1957
NHW in4 out4	62.5518	0.9657	0.5555	0.3128	0.2313	0.4474
NHW in10 out10	129.0767	1.9131	0.9238	0.6454	0.5574	0.8016
NHW in7 out7	18.5315	0.2910	0.2516	0.0927	0.0934	0.1989
NHW in1 out1	77.4775	1.6428	1.5863	0.3874	0.4923	1.2216
NHW in3 out3	8.3579	0.1307	0.1219	0.0418	0.0599	0.1012

(d) Performance evaluation for input $u = 5$

Figure 5.5: Analysis of identified Hammerstein Wiener models for system 4.8, with parameters: $a_1 = a_2 = b_1 = b_2 = c_1 = c_2 = 0.1$, $\hat{\sigma}_1 = \hat{\sigma}_2 = \hat{K}_2 = 1, \hat{K}_1 = 2$ and initial condition $(1, 1, 1, 1)$, simulated for 500 seconds. Subfigures 5.5a and 5.5b show the outputs of the HW models;*in* and *out* are the numbers of input and output nonlinearity units. The tabels 5.5c and 5.5d show the similarity scores between these outputs and the original one.



(a) input $u = 0.15$



(b) input $u = 0.75$

	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25
NHW in10 out10	0.2016	0.0116	0.0237	0.0040	0.0083	0.0186
NHW in3 out3	1.7808	0.0806	0.1030	0.0356	0.0529	0.0863
NHW in5 out5	0.6910	0.0512	0.1777	0.0138	0.0525	0.1360
NHW in7 out7	2.0306	0.0959	0.1737	0.0406	0.0731	0.1402

(c) Performance evaluation for input $u = 0.15$

	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25
NHW in10 out10	1.1008	0.0504	0.0237	0.0220	0.0237	0.0237
NHW in3 out3	5.5405	0.2509	0.1263	0.1108	0.1181	0.1236
NHW in5 out5	0.4288	0.0207	0.0472	0.0086	0.0177	0.0373
NHW in7 out7	1.7060	0.0845	0.3147	0.0341	0.1038	0.2444

(d) Performance evaluation for input $u = 0.75$

Figure 5.6: Analysis of identified Hammerstein Wiener models for system 4.8, with parameters: $a_1 = a_2 = b_1 = b_2 = c_1 = c_2 = 0.1$, $\hat{\sigma}_1 = \hat{\sigma}_2 = \hat{K}_2 = 1, \hat{K}_1 = 2$ and initial condition $(1, 1, 1, 1)$, simulated for 500 seconds. Subfigures 5.5a and 5.5b show the outputs of the HW models;*in* and *out* are the numbers of input and output nonlinearity units. The tabels 5.5c and 5.5d show the similarity scores between these outputs and the original one.

Chapter 6

Robust Control Approach

Even if commonly used interchangeably, *model reduction* and *approximation* have slightly different meanings. Approximation doesn't necessarily imply a theoretical connection between the two systems, whereas in model reduction, the reduced system is built on the theoretical characteristics of the original one. The main idea behind model reduction techniques is that, from a system's output perspective, a high - dimensional state vector can be accurately represented through a lower - dimensional one. The connection between these two representations is done through a projection space, an invertible matrix T. In the transformed representation, the states of the system are descendingly ordered according to the contribution they bring to the output. Different reduction methods imply different ways of finding the transformation T. This equivalently translates into different properties that the new system's representation will have. The next step, which amounts to actually *reducing* the system, is the truncation of some states. Depending on the procedure and on the application at hand, different truncation criteria are used.
In our framework, the goal is to approximate monotone systems. Keeping this in mind, we formulate the following requirmements, which a "good" model reduction technique should satisfy:

1. The reduced system should be "less complex" than the original one (in a sense to be made formal later on). Essentially, a less complex system will be easier to analyze and simulate.

2. Theoretical error bounds between the original system and the reduced one should be guaranteed.

3. The reduced system should share the same stability properties as the original one. For instance, if the original LTI system is stable (theorem 1.3.2) for a fixed parameter vector, then the reduced system should keep the same dynamical behavior.

4. If the original system is monotone, then the reduced one should (ideally) as well be monotone.

The method which we will focus on here, *balanced trunaction*, is among the most commonly employed model reduction schemes. First introduced by Mullis and Roberts in 1976 ([26]) and later applied to control systems by Moore in 1982 ([25]), balanced truncation posesses the big advantage of fully verifying the first three requirments enumerated above. Moreover, even if the fourth requirement is not satisfied by default, conditions under which the initial reduced system can be transformed into a monotone one can be derived.
In the balanced truncation framework, the transformation T is chosen such that, in the new coordinates, the system's modes are ranked according to a predefined measure based on specific properties of the system, such as *observability* and *controllability*. The method is called *balanced* because, under the chosen measure, controllability and observability will always be equally quantified. An unique indicator, the *Hankel Singular Value*, is assigned to each mode, measuring its contribution to the output. The Hankel Singular values are properties of the system and not dependent on its state-space representation, hence invariant to any transformation T. By simply removing the states which contribute in a not significant ammount to the output, the behavior of the trimmed model will be similar to the behavior of the untrimmed one. All these theoretical concepts will be discussed in detail in this chapter.

6.1 Observability and Controllability

Informally, if an I/O system is controllable, then it can reach any arbitrary final state, when starting at any arbitrary initial state, by means of an external input. On the other hand, if the system is observable, then any arbitrary initial state, as well as all its intermediate states, can be determined when only knowing the external outputs.
We will restrict the formal analysis of these two concepts for the case of stable LTI systems G, as discussed in Chapter 1. For a better readability, we will show here once more the state-space representation of an LTI stable system:

$$G : \begin{cases} \dfrac{\mathrm{d}x(t)}{\mathrm{d}t} &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{cases} \tag{6.1}$$

where, as the system is stable, the matrix A is assumed to be Hurwitz (theorem 1.3.2). The specific terminology used, as well as the overall workflow in describing these concepts, can be found in [23].

6.1.1 Observability

Recall the *homogeneous response* and the *homogeneous output response* of the system G discussed in Chapter1 (equation 1.7), i.e. the expression of the states and output respectively, when the input is an identically vanishing function. Recall as well as well as the L_2 norm of a signal, measuring its energy (equation 3.6), as discussed in Chapter 3.

If the state-space representation of the system G is fixed, then the homogeneous output response is a *linear function* in the initial condition x_0. We will denote this function by E_o, representing a measure of the "observable output energy" accumulated in the initial state x_0:

$$E_o(x_0) := \int_0^\infty y_h^2(t)dt \tag{6.2}$$

By replacing the expression of the homogenous output response y_h into the expression of the L_2 norm, one obtains that:

$$E_o(x_0) = x_0' W_o x_0 \tag{6.3}$$

where

$$W_o = \int_0^\infty e^{A^T\tau} C^T C e^{A\tau} d\tau \tag{6.4}$$

W_o is the *observability Gramian* matrix of the system (first introduced in the case of the H_2-norm, see equation 3.16).

By definition, $E_o(x_0) \geq 0, \forall x_0 \in \mathbb{R}^n$, which translates into $x_0' W_o x_0 \geq 0, \forall x_0 \in \mathbb{R}^n$, meaning that the observability Gramian W_o is always positive semidefinite. By employing linear algebra tools, it can be derived that:

$$E_o(x_0) = 0 \ \forall x_0 \in \mathbb{R}^n \Leftrightarrow \mathcal{O}x_0 = 0 \ \forall x_0 \in \mathbb{R}^n \tag{6.5}$$

where $\mathcal{O} := [C \ CA \ CA^2 \ \dots \ CA^{n-1}]$ is the *observability matrix* assigned to a LTI system with the state space representation 6.1. Equation 6.5 is equivalent to the condition that matrix $\mathcal{O}$ is not full-ranked.
To conclude, the matrix $\mathcal{O}$ is full ranked if and only if the observability Gramian is positive definite. In case this happens, the system G (or the pair (C, A)) is said to be *observable*. When the pair (C, A) is observable, the *primal observability measure* is given by:

$$E_o(x_0) = \frac{1}{pW_0^{-1}p'} \text{ for } p \neq 0 \tag{6.6}$$

If we want to look at the observability of the system from a different perspective, let p be a row vector with n elements. By multiplying the vector p with the state-vector of the system, x, one obtains a linear combination of the system states x. Unlike $E_o(x_0)$, which will be refered to as the *primal* observability measure, the *dual* observability measure is dependent on the vector p, employed to obtain the linear combination of elements of the system's initial condition:

$$E^o(p) := \inf_{x_0 : px_0 = 1} E_0(x_0) \text{ for } p \neq 0 \tag{6.7}$$

It represents the minimal output energy which can be observed for $t \geq 0$ when $px(0) = 1$.

6.1.2 Controllability

Recall now the state response of the system 6.1 (equation 1.5). Just for theoretical purposes, assume that the signals were defined on $\mathbb{R}$, instead of $\mathbb{R}_+$, which means also looking at negative timepoints. In this case, as the matrix A was assumed to be Hurwitz, the state response satisfies $x(t) \to 0$ as $t \to \infty$ and, more precisely, for $t < 0$:

$$x(t) = \int_0^\infty e^{A\tau} B u(t-\tau) d\tau \tag{6.8}$$

In this case, one can say that the input u *drives* the state of the system from $x(-\infty) = 0$ up to the initial condition x_0. For $t = 0$, equation 6.8 can be seen as an unique mapping between the input signal u and x_0. For inputs defined (solely) for $t \leq 0$, we will denote this functional dependency by X:

$$X(u) = \int_0^\infty e^{A\tau} B u(-\tau) d\tau \tag{6.9}$$

and hence:

$$x_0(u) = X(u) \tag{6.10}$$

In the controllability case, we will first define the *dual* controllability measure. This measure represents the maximum energy which can be used to reach the initial condition x_0, by using an input u of unit energy. Using the same notation, where px represents a linear combination of the states of the system:

$$E^c(p) := \sup_{||u||_2 \leq 1} pX(u) \tag{6.11}$$

The *primal* controllability measure is defined on the basis of the dual one and it is a function of the column vector x_0:

$$E_c(x_0) := \inf_{p:\ px_0 = 1} E^c(p) \tag{6.12}$$

Following a similar reasoning as in the observability case, by replacing the expresion for the state response and taking into account that the matrix A is Hurwitz, one can derive a quadratic dependency of $E^c(p)$ with respect to p. More precisely:

$$E^c(p) = pW_c p' \tag{6.13}$$

where

$$W_o = \int_0^\infty e^{A\tau} B B^T e^{A^T \tau} d\tau \tag{6.14}$$

is the *controllability Gramian* matrix of the system (first defined in the case of the H_2 norm as well, see equation 3.15).
By definition, $E^c(p) \geq 0, \forall p^T \in \mathbb{R}^n$, which translates into $pW_c p' \geq 0, \forall p^T \in \mathbb{R}^n$, meaning that the controllability Gramian W_c is always positive semidefinite. By employing linear algebra tools, it can be derived that:

$$E^c(p) = 0 \ \forall p^T \in \mathbb{R}^n \Leftrightarrow \mathcal{C}p = 0 \ \forall p^T \in \mathbb{R}^n \tag{6.15}$$

where $\mathcal{C} := [B\ AB\ A^2B\ \ldots\ A^{n-1}B]$ is the *controllability matrix* assigned to a LTI system with the state space representation 6.1. Equation 6.15 is equivalent to the condition that matrix $\mathcal{C}$ is not full-ranked.
Hence, the matrix $\mathcal{C}$ is full ranked if and only if the controllability Gramian is a positive definite matrix. In case this happens, the system G (or the pair (A, B)) is said to be *observable*.
Whenever the pair (A, B) is controllabile, the primal controllability measure E_c is given by:

$$E_c(x_0) = \frac{1}{x_0' W_c^{-1} x_0} \text{ for } x_0 \neq 0 \tag{6.16}$$

6.2 Transformation of State Variables

As we saw, the state-space realization of an LTI system is given by the set of 4 constant matrices (A, B, C, D). However, the set of equivalent state-space realizations for the same transfer function $\mathbf{G}$ is infnite. We will now emphasize the procedure through which these equivalent state - space representations are obtained, which amounts to transforming the state vector x through linear operations, i.e. linearly independent combinations of the original state variables (see [29] for further details).
More specifically, given the state vector $x(t) = \big(x_1(t), \ldots, x_n(t)\big)$, a new state vector $\hat{x}(t) = \big(\hat{x}_1(t), \ldots, \hat{x}_n(t)\big)$ may be generated as such:

$$\hat{x}_i(t) = t_{i1}x_1(t) + t_{i2}x_2(t) + \ldots + t_{in}x_n(t) \tag{6.17}$$

where t_{ij} are real constants, $\forall 1 \leq i \leq n$. This linear transformation can be written in matrix form:

$$\hat{x} = Tx \tag{6.18}$$

provided that $T \in \mathbb{R}^{n \times n}$ is a non-singular matrix. This being the case, we will call T a *matrix transformation* or simply a *transformation*. Equation 6.18 is equivalent to $x = T^{-1}\hat{x}$ (note than using the notation in 6.18, t_{ij} are the elements of T^{-1}). By replacing $x = T^{-1}\hat{x}$ in the state-space representation of the system, we obtain:

$$G : \begin{cases} \dfrac{\mathrm{d}(T^{-1}\hat{x})(t)}{\mathrm{d}t} & = AT^{-1}\hat{x}(t) + Bu(t) \\ y(t) & = CT^{-1}\hat{x}(t) + Du(t) \end{cases} \tag{6.19}$$

In order to fully transform the system into the $\hat{x}$ coordiantes, we multiply the left side of the first equation by T:

$$G : \begin{cases} \dfrac{\mathrm{d}\hat{x}(t)}{\mathrm{d}t} & = TAT^{-1}\hat{x}(t) + TBu(t) \\ y(t) & = CT^{-1}\hat{x}(t) + Du(t) \end{cases} \tag{6.20}$$

If we define the new matrices $\hat{A} := TAT^{-1}$, $\hat{B} := TB$, $\hat{C} := CT^{-1}$ and $\hat{D} := D$, then the system 6.20 becomes:

$$G : \begin{cases} \dfrac{\mathrm{d}\hat{x}(t)}{\mathrm{d}t} & = \hat{A}\hat{x}(t) + \hat{B}u(t) \\ y(t) & = \hat{C}\hat{x}(t) + \hat{D}u(t) \end{cases} \tag{6.21}$$

The system 6.20 is an equivalent state-space representation of the system 6.1, hence its input/output dynamical behavior is unchanged. In consequence, as the eigenvalues of the A matrix in a state-space representation of a LTI system carry information about the stability properties of the system, it follows that the eigenvalues should as well be invariant under state transformation. We will formally prove this statement.

Proposition 6.2.1. *The eigenvalues of the matrix A of a LTI system are invariant under matrix transformations, i.e. $eig(A) = eig(\hat{A})$*

Proof. The eigenvalues of a matrix are the solution to its characteristic equation:

$$\begin{aligned} \det\left[\lambda I - \hat{A}\right] &= \det\left[\lambda I - TAT^{-1}\right] \\ &= \det\left[\lambda TIT^{-1} - TAT^{-1}\right] \\ &= \det\left[(T(\lambda I - A)T^{-1}\right] \\ &= \det[T]\det[\lambda I - A]\det\left[T^{-1}\right] \text{ (distributivity property)} \\ &= \det[T]\det[\lambda I - A]\frac{1}{\det[T]} \text{ (invertibility of T)} \\ &= \det[\lambda I - A] \end{aligned} \tag{6.22}$$

□

We will call these values the *Hankel singular values* of the system with transfer function matrix $\mathbf{G}$ and we will denote them by σ_i. From a controllability perspective, the Hankel singular values σ_i quantify how far one can reach in the direction of each of its corresponding eigenvectors, by using an input u with enegery $||u||_2 \leq 1$. From an observability perspective, the Hankel singular values σ_i quantify how much output energy $||y||_2$ exists if the initial condition has the value of its corresponding eigenvectors.

6.3 Balanced Truncation

6.3.1 The Transformation T

Our goal is to transform the state-space realization 6.1 into a new state-space realization, for which the *minimum* of the joint controllability and observability measure over all non-zero elements is *maximal*. A basis in this new subspace

will yeild the columns on the transformation used for comuting between the two representations ([23]).

The following theorem is stated in [23] and can be proven by straightforward computations:

Theorem 6.3.1. *Let G be a LTI system both controllable and observable and $W_c = L_c L_c^T$ and $W_o = L_o L_o^T$ be the Cholesky decompositions of the controllability and observability Gramians W_c and W_o, i.e. their decomposition into the product of a lower triangular matrix and its conjugate transpose. Let*

$$\rho_1 \geq \dots \rho_r \geq \rho_{r+1} \geq \rho_n \geq 0 \tag{6.23}$$

be the eigenvalues (possibly repeated) of the matrix $L_c' W_o L_c$ and $\sigma_i = \psi_i^{1/2}$. Let $\psi_1, \dots, \psi_n$ be the corresponding normalized eigenvectors of $L_c' W_o L_c$, i.e.

$$L_c' W_o L_c \psi_i = \rho_i \psi_i \qquad ||\psi_i||_2 = 1 \qquad \psi_i \psi_k = 0 \text{ for } i \neq k. \tag{6.24}$$

Then it holds that:

(a) $\rho_1 \geq \dots \rho_r \geq \rho_{r+1} \geq \rho_n \geq 0$ are also the eigenvalues of the matrix $L_o W_c L_o'$. The corresponding normalized row eigenvectors, ϕ_i, for $i \in \{1, \dots, n\}$, are:

$$\phi_i = \sigma_i^{-1} \psi_i' L_c' L_o' \tag{6.25}$$

(b) The set of all linear combinations of vectors $L_c \psi_i$ is the only n-dimensional linear subspace V in $\mathbb{R}^n$ such that $E_{oc}(v) \geq \rho_n$, $\forall v \in V$.

(c) The set of all linear combinations of row vectors $\phi_i L_o$ is the only n-dimensional linear subspace U of row n-vectors such that $E^{oc}(u) \geq \rho_n$, $\forall u \in U$.

(d) As defined previously:

$$V = L_c \Big[\psi_1 \sigma_1^{-1/2} \dots \psi_n \sigma_n^{-1/2} \Big]$$

and

$$U = \Big[\sigma_1^{-1/2} \phi_1 \dots \sigma_n^{-1/2} \phi_n \Big]^T L_o$$

Moreover, $UV = VU = I_n \Rightarrow U = V^{-1}$

Now, denote by Ψ the matrix whose columns are the ordered (by eigenvalues magnitude) eigenvectors of $L_c^T W_o L_c$ and Σ is the diagonal matrix with σ_i on the diagonal. Ψ is a orthogonal matrix, since its colums are normalized vectors. The following theorem is an immediate consequence of theorem 6.3.1:

Theorem 6.3.2. *The transformation $T = L_c \Psi \Sigma^{-1/2}$ yields an equivalent state-space model $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$, for which the new observability and controllability Gramians, computed in terms of the transformation T as: $\bar{W}_c = T^{-1} W_c T^{-T}$ and $\bar{W}_o = T^T \bar{W}_o T$ satisfy:*

$$\bar{W}_c = \bar{W}_o = \Sigma$$

For a complete proof, as well as for all the technical verifications, such as the invertibility of T, see [19].
Even if here it is not the case, it is worth mentioning that, as it can be seen from the theorem 6.3.1, in practice, one only needs to compute the matrices Ψ and Σ, while the computation of the Gramians (which becomes very demanding for large systems) can be avoided. For more information on the matter, as well as for means of computing approximations to the Gramians, see [7] and [19].

6.3.2 State Reduction

As we have now formally defined all the concepts, we can also justify the name of *balanced realization*, which comes from the fact that, in the new coordinates $\hat{x} = Tx$, the observability and controllability Gramians become identical. Moreover, if $\hat{x} = (\hat{x}_1, \ldots, \hat{x})^T$, then $\hat{x}_1$ is the most controllable and observable state, while $\hat{x}_n$ is the least observable and controllable one. The joint observability-controllability measure E_{oc} (and respectively its dual, E^{oc}) measures how much energy is transferred from the input into the output. Hence, by simple trimming, we will remove the states which are not significantly involved in this energy transfer, i.e. both hard to excite with finite amounts of input energy, as well as not resulting in much output energy. These states are the ones for which the corresponding σ_i is small.
More specifically, in the new coordinates $\hat{x}$, a reduced system for which the last $n - r$ states were discarded becomes:

$$\hat{G}_r : \begin{cases} \dfrac{\mathrm{d}\hat{x}_r(t)}{\mathrm{d}t} & = \hat{A}_r\hat{x}_r(t) + \hat{B}_r u(t) \\ y(t) & = \hat{C}_r\hat{x}_r(t) + \hat{D}_r u(t) \end{cases} \tag{6.26}$$

where $\hat{x}_r : \mathbb{R}_+ \to \mathbb{R}^{n_r}$, $u : \mathbb{R}_+ \to \mathbb{R}^m$, $y : \mathbb{R}_+ \to \mathbb{R}^{p_r}$, $A_r : \mathbb{R}_+ \to \mathbb{R}^{n_r \times n_r}$ and $A_r := \hat{A}(1 : r, 1 : r)$, $B : \mathbb{R}_+ \to \mathbb{R}^{n_r \times m}$ and $B_r := \hat{A}(1 : r, 1 : m)$, $C : \mathbb{R}_+ \to \mathbb{R}^{p_r \times n_r}$ and $C_r := \hat{A}(1 : p, 1 : r)$ and $D : \mathbb{R}_+ \to \mathbb{R}^{p \times m}$ with $\hat{D}_r = \hat{D}$.

Provided that the initial matrix A is Hurwitz, then, by proposition 6.2.1, the matrix $\hat{A}$ is Hurwitz as well. The fact that A_r, which is the a trimmed variant of $\hat{A}$, is obviously also Hurwitz. As it can be seen in theorem 6.3.1, the balanced truncation method makes use of the Cholesky decompositions of the observability and controllability Gramians. However, only symmetric and positive definite matrices can be decomposed into Cholesky factors. This is therefore the justification for the controllability and observability requirments on the system, which translate into positive definite Gramian matrices (the symmetry of the Gramians can be seen from the way they are defined, see equations 6.4 and 6.14). The same fact justifies the following definition:

Definition 6.3.3. *The* complexity *of an LTI system* $\mathbf{G}$ *(also known as its* order *or its* McMillan degree*) is the dimension of the state vector of a minimal realization of* $\mathbf{G}$*, i.e. its state-space form which is both controllable and observable.*

Therefore, in the new terminology, the reduced system G_r should have complexity $r' \leq r < n$. The reduced system will have a complexity $r' \neq r$ if it is not minimal. For our theoretical purpose here, however, this situation is not interesting as the following relation will always hold: $r' < r$, as a simple consequence of definition 6.3.3. Hence, we will assume that the reduced system has complexity r, which is also equal to the dimension of its state vector.

6.3.3 Error Bounds

Lower Bounds

One of the advantages of balanced truncation is that it offers a *lower bound error* for arbitrary reduced models, under the H_∞ norm (see definition 3.2.4 and equation 3.22):

$$\sigma_r \leq ||\mathbf{G} - \mathbf{G_k}||_\infty \tag{6.27}$$

for any system $\mathbf{G_k}$ with complexity $k < r$. A consequence of this statement (resulting from the triangle inequality of the norm) is that the H_∞ norm of any reduced model of complexity less than r will deviate with at least σ_r from the H_∞ norm of the original model. This statement is strong because it offers information about *any* reduced model, regardless of its type or origin. For a proof of 6.27, see [23].

Upper Bounds

The following theorem provides upper bounds from the H_∞ model reduction error obtained through balanced truncation:

Theorem 6.3.4.

$$||\mathbf{G} - \mathbf{G_k}||_\infty \leq 2 \sum_{i=r+1}^{n} \sigma_i \tag{6.28}$$

As the singlular values of exponentially stable LTI systems decay exponentially, the error bounds 6.27 and 6.28 are expected to be not highly different. For a proof of this statement, as well as for the derivation of these error bounds, see [23].

6.4 Application

We applied the balanced truncation model reduction scheme to the I/O monotone examples presented in Chapter 4. As the systems are not linear and the balanced truncation algorithm can only be applied on stable LTI systems, we chose to first linearize them and then, provided that the linearization is stable, apply the balanced truncation routine. Therefore, two approximation algorithms have been succesively applied on the inital (nonlinear) monotone systems:

1. Linearization around steady-state;

2. Balanced Truncation

As mentioned in the beginning of this chapter, balanced truncation posesses strong advantages from a model reduction perspective, mainly due to its upper and lower error bounds (equations 6.27 and 6.28), as well as its stability preserving properties. In addition to this, the balanced truncation technique can be seen as particularly suitable for being applied in the monotone systems framework, mainly due to the following observations, which will be formally proven in this section:

- When starting with a nonlinear monotone system, the monotonicity property is transmitted to its linearization around any operating point. Hence, the balanced truncation method will still be applied on a monotone (linear) system.
- The reduced system can be transformed to an equivalent monotone one, if certain conditions on the state space realization of the reduced one are being fulfilled.

Proposition 6.4.1. *The linearization of a monotone system is a monotone system.*

Proof. The species graph used for for identifying monotonicity (proposition 2.2.14) is constructed on the basis of the Jacobian matrix of system (in the I/O case, including the partial derivatives of the rates with respect to inputs and outputs). As a linear system has same Jacobian matrix (or, to be more precise, same linearization matrices - see Chapter 1, section 1.4) as its linearization, the conclusion follows. □

The similarity between the output trajectories of the original system and the approximated ones (simple linearization and linearization followed by balanced truncation) was evaluated under the signal and system norms described in detail in Chapter 3, for different values of the system's input. The balanced truncation routine was performed multiple times, with different reduction thresholds and, equivalently, rendering reduced models of different complexities. Moreover, unlike the sistem identification approach in Chapter 5, which approximated the monotone modules by nonlinear Hammerstein Wiener models, in this chapter also the system norms H_2 and H_∞, described in sections 3.2.1 and 3.2.2 were used, when comparing linear systems.

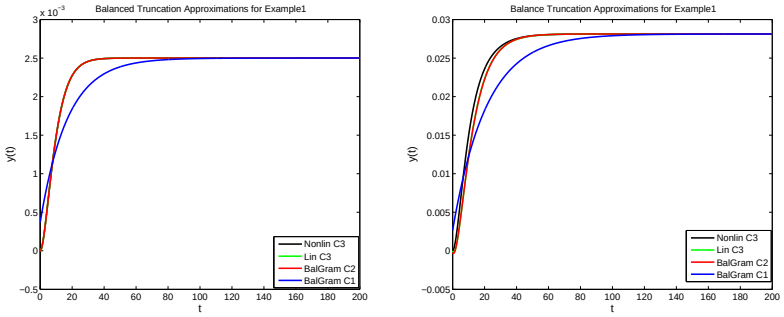
6.4.1 Example1

The dynamical behavior of Example1 (system 4.2) was already discussed in Chapter 4 and its main characteristic was seen to be a qualitatively unchanged output, regardless of the input value (see figure 4.2). It is worth mentioning that the state space realization of the linearization around the operating point given by equations 4.3, 4.4 and 4.5 is both controllable and observable, hence minimal.

We will show how the approximations performed when validated on 4 different constant inputs: 0.01, 1, 10 and 20. Generally, balanced truncation rendered very good results in the case of sysem 4.2, for inputs spanning over several orders of magnitude. However, preference was shown for small values of inputs.
More specifically, for input values less than 0.1, the nonlinear system is perfectly approximated by its linearization around the steady state (see figure 6.1, subfigures 6.1c and 6.1c for the corresponding norm measures). In addition to this, for an input of 0.01, reducing one state of the system renders a new linear 2-dimensional system whose output is perfectly similar to the nonlinear output. A slightly different output (but still very similar) is obtained if reducing two states of the balanced model, hence approximating the nonlinear output by a 1-variable linear differential equation. These observations also hold for the one order of magnitude higher input of 0.1, in an almost identical manner.
We therefore conclude that, in the case of system 4.2 and for input values ≤ 0.1, balanced truncation is an ideal method, as the system is very well approximated by a very simple dynamical system: a linear one-variable differential equation.
When incresing the input value, even if the difference between the nonlinear output and its linearization increases, the approximations still perform well under all the norms (see figure 6.2, subfigures 6.2c and 6.2d for the corresponding tables).
One consistent observation is that the unchanged qualitative behavior of the nonlinear system 4.2 is kept by its linearizations and, furthermore, by the reduced balanced models. Moreover, for all the inputs, the second-order balanced truncation is still highly similar (almost identical) to the linear system, meaning that the third state of the balanced three-orders system, with the lowest Hankel Singular Value, has little influence of the input-output behavior of the linear system.
However, the nonlinear Hammerstein Wiener identified models from Chapter 5 performed better under all the discussed norms than the present approximations, for input values of 20 and 1 (see tables from figure 5.2, subfigures 5.2c and 5.2d).

6.4.2 Example2

The dynamical behavior of Example2 (system 4.8) was already discussed in Chapter 4 and it was seen to exhibit three types of qualitatively different dynamical behaviors (see figure 4.4). The state space realization of the linearization around the operating point given by equations 4.9, 4.10, 4.11 and 4.12 is non-minimal, as the linear system is neither controllable nor observable. The (equivalent) minimal realization, on which the balanced truncation routine will be applied, has order 2. As it can be seen in figures 6.3 and 6.4, the linearization, as well as its balanced truncation (which approximates the system by one a single one-variable differential equation), work better for larger values of the inputs (0.8 and 5). As the input increases in magnitude, so does the quality of the approximations. As in the case of system 4.2, there are input ranges for which the unreduced linear system almost perfectly approximates the nonlinear



(a) input $u = 0.01$ (b) input $u = 0.1$

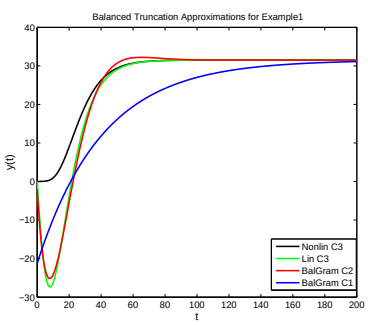
	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25	H2 norm	Hinf norm
Lin C3	5.4839e-006	3.1844e-007	2.7498e-007	2.7419e-008	8.2670e-008	2.1088e-007		
BalGram C2	5.4147e-006	3.3091e-007	2.9065e-007	2.7073e-008	7.8973e-008	2.2009e-007	0.0225	0.0518
BalGram C1	0.0160	6.9005e-004	4.5324e-004	8.0146e-005	2.6141e-004	3.8930e-004	0.0225	0.0518

(c) Performance evaluation for input $u = 0.001$

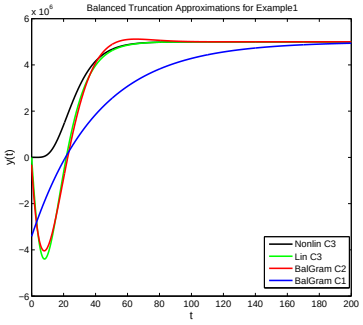
	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25	H2 norm	Hinf norm
Lin C3	0.0515	0.0031	0.0028	2.5738e-004	7.8256e-004	0.0021		
BalGram C2	0.0517	0.0032	0.0028	2.5844e-004	7.5701e-004	0.0021	2.1982e-004	8.1260e-004
BalGram C1	0.2318	0.0090	0.0054	0.0012	0.0038	0.0049	0.0275	0.0668

(d) Performance evaluation for input $u = 0.01$

Figure 6.1: Analysis of balanced truncation model reduction routine applied on system 4.2, simulated under the following parameter values:$a_1 = a_2 = a_3 = K_i = K_e = 0.1$ and initial condition $(0, 0, 0)$ and linearized around its steady-state. Total simulation time: 200 seconds. Subfigures 6.1a and 6.1b show the simulated outputs of: the nonlinear system (*Nonlin*), its linearization (*Lin*) and its balanced reduced models (*BalGram*). On the legend, the number next to the output type represents its complexity, e.g. $C2$ means a system of complexity 2. Subfigures 6.1c and 6.1d show the ranking of these models under the norms discussed in Chapter 3, as compared to the output of system 4.2. *Stab norm* is the stability from definition 3.1.9, with different weight values and the system norms H_2 and H_∞ are computed between the linearization of the nonlinear system and the reduced linear models.



(a) input $u = 1$



(b) input $u = 20$

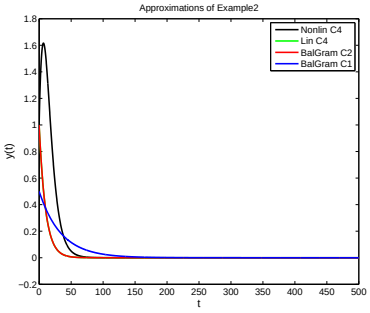
	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25	H2 norm	Hinf norm
Lin C3	514.7526	31.3108	27.8018	2.5738	7.8255	21.1431		
BalGram C2	565.9504	31.0420	25.5835	2.8298	7.5032	19.5567	1.5730	3.6024
BalGram C1	1.2345e+003	36.0594	21.2856	6.1726	16.1628	19.5780	11.0877	37.6449

(c) Performance evaluation for input $u = 1$

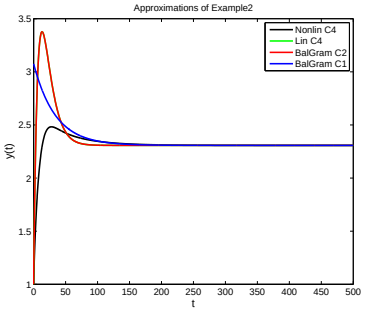
	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25	H2 norm	Hinf norm
Lin C3	8.2360e+007	5.0097e+006	4.4483e+006	4.1180e+005	1.2521e+006	3.3829e+006		
BalGram C2	9.0619e+007	4.9663e+006	4.0908e+006	4.5309e+005	1.2008e+006	3.1275e+006	1.2653e+004	2.8962e+004
BalGram C1	1.9750e+008	5.7667e+006	3.4178e+006	3.4178e+006	2.5863e+006	3.1407e+006	8.8675e+004	3.0122e+005

(d) Performance evaluation for input $u = 20$

Figure 6.2: Analysis of balanced truncation model reduction routine applied on system 4.2, simulated under the following parameter values:$a_1 = a_2 = a_3 = K_i = K_e = 0.1$ and initial condition $(0, 0, 0)$ and linearized around its steady-state. Total simulation time: 200 seconds. Subfigures 6.1a and 6.1b show the simulated outputs of: the nonlinear system (*Nonlin*), its linearization (*Lin*) and its balanced reduced models (*BalGram*). On the legend, the number next to the output type represents its complexity, e.g. $C2$ means a system of complexity 2. Subfigures 6.2c and 6.2d show the ranking of these models under the norms discussed in Chapter 3, as compared to the output of system 4.2. *Stab norm* is the stability from definition 3.1.9, with different weight values and the system norms H_2 and H_∞ are computed between the linearization of the nonlinear system and the reduced linear models.



(a) input $u = 0.1$



(b) input $u = 0.4$

	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25	H2 norm	Hinf norm
Lin C2	28.5944	1.4923	1.1556	0.0572	0.2900	0.8670		
BalGram C1	27.8473	1.4722	1.1997	0.0557	0.3196	0.9063	1.7443e-015	5.6858e-015

(c) Performance evaluation for input $u = 0.1$

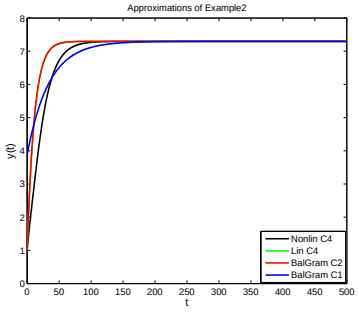
	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25	H2 norm	Hinf norm
Lin C2	157.2638	3.7818	1.5618	0.3145	0.8464	1.3234		
BalGram C1	148.5736	3.5319	1.4916	0.2971	0.7978	1.2603	0.0021	0.0069

(d) Performance evaluation for input $u = 0.4$

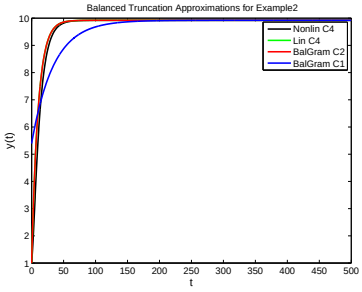
Figure 6.3: Balanced Truncation applied on system 4.8, together with the performance evaluation under the defined norms.*Stab* represents the *Stability norm*, with different weights. The system was simulated for the following parameter values: $a_1 = a_2 = b_1 = b_2 = c_1 = c_2 = 0.1$, $\hat{\sigma}_1 = \hat{\sigma}_2 = \hat{K}_2 = 1, \hat{K}_1 = 2$ and initial condition $(1, 1, 1, 1)$. Total simulation time: 500 seconds.

one and, in the same time, for the same input ranges, the entire system can be decently approximated by only one differential equation (see tables 6.4d and 6.4c).

As compared with the nonlinear Hammerstein Wiener identification, for an input 0f 0.1, better results than with the balanced truncation routine were obtained for the HW model structure with 9 units in the input, as well as in the output nonlinearity. For the input values of 0.4 and 0.8, however, balanced truncation ranked much better under all the norms, while for an input value of 5, both balanced truncation and some of the Hammerstein Wiener structures obtained very good scores.



(a) input $u = 0.8$



(b) input $u = 5$

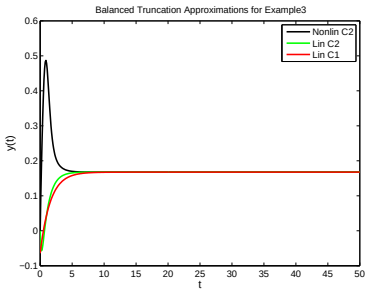
	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25	H2 norm	Hinf norm
Lin C2	101.9827	3.8531	2.1679	0.2040	0.6293	1.6550		
BalGram C1	67.4180	3.1162	2.6682	0.1348	0.7079	2.0148	0.3413	1.1124

(c) Performance evaluation for input $u = 0.8$

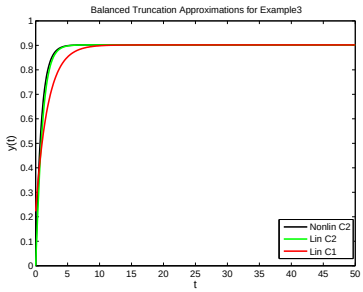
	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25	H2 norm	Hinf norm
Lin C2	23.8715	1.4010	1.2804	0.0477	0.3204	0.9604		
BalGram C1	100.4391	3.8334	4.3899	0.2009	1.2784	3.3527	0.0020	0.0066

(d) Performance evaluation for input $u = 5$

Figure 6.4: Balanced Truncation applied on system 4.8, together with the performance evaluation under the defined norms. *Stab* represents the *Stability norm*, with different weights. The system was simulated for the following parameter values: $a_1 = a_2 = b_1 = b_2 = c_1 = c_2 = 0.1$, $\hat{\sigma}_1 = \hat{\sigma}_2 = \hat{K}_2 = 1, \hat{K}_1 = 2$ and initial condition $(1, 1, 1, 1)$. Total simulation time: 500 seconds.



(a) input $u = 0.15$



(b) input $u = 0.75$

	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25	H2 norm	Hinf norm
Lin C2	0.7231	0.1507	0.4649	0.0145	0.1162	0.3487		
BalGram C1	0.7852	0.1506	0.4547	0.0157	0.1140	0.3411	0.0051	0.0028

(c) Performance evaluation for input $u = 0.15$

	L1 norm	L2 norm	Linf norm	Avg norm	Stab norm w1 = 0.25, w2 = 0.75	Stab norm w1 = 0.75, w2 = 0.25	H2 norm	Hinf norm
Lin C2	0.1341	0.0266	0.0794	0.0027	0.0199	0.0596		
BalGram C1	0.6073	0.0797	0.2220	0.0121	0.0582	0.1674	0.2923	0.2083

(d) Performance evaluation for input $u = 0.75$

Figure 6.5: Balanced Truncation applied on system 4.17, together with the performance evaluation under the defined norms. *Stab* represents the *Stability norm*, with different weights. The system was simulated for the following parameter values: $a_1 = a_2 = 1$, $\beta_1 = 200$, $\beta_2 = 10$, $\gamma_1 = \gamma_2 = 4$, $K_1 = 30, K_2 = 1$ and initial condition $(0, 0)$. Total simulation time: 20 seconds

6.4.3 Example3

The dynamical behavior of Example3 (system 4.17) was already discussed in Chapter 4 and it was seen to exhibit 2 types of qualitatively different dynamical behaviors (see figure 4.6). The linearization around the steady state of the nonlinear system is a minimal stable system. The balanced truncation approach performs decently on small inputs, such as 0.15 and well for larger input values, such as 0.75 (see figure 6.5), case in which the linearization almost perfectly approximates the nonlinear output.
If compared with the Hammerstein Wiener models, the model with 10/10 input/output nonlinearity pieces performs better than the balanced truncation routine, for an input of 0.1. However, in the case of higher input, the balanced truncation routine performs better.

6.5 Monotone Transformation

A highly desired property of this model reduction routine is the monotonicity of the reduced system. Even if this property is not satisfied by default, theoretical connections between the transformed system, under the balanced transformation T, and a monotone system can be derived. In this sense, the following proposition, discussed in [19] and proven in [25], is of particular interest:

Proposition 6.5.1. *If the Hankel singular values σ_i are distinct, then its corresponding eigenvectors (the columns of the transformation T) are uniquely determined to within sign. In other words, the transformation T is esentially unique, up to postmultiplication by any matrix of the form diag($\pm 1, \ldots, \pm 1$). In case two or more σ_i are equal, then its corresponding eigenvectors can be arbitrarely rotated in their corresponding eigenspace.*

If the Hankel singular values of the system are different, the above proposition basically states that, provided that the transformation T is balanced, any transformation $T_1 := T\mathcal{D}$ is balanced as well, where $\mathcal{D}$=diag($\pm 1, \ldots, \pm 1$). Moreover, there are no other balanced transformations. Let's now derive the actual form of the equivalent balanced state space realization $(\hat{A}_1, \hat{B}_1, \hat{C}_1, \hat{D}_1)$. Recall that $\hat{A} = TAT^{-1}$, $\hat{B} = TB$, $\hat{C} = CT^{-1}$ and $\hat{D} = D$. Under the transformation T_1, we have therefore that:

$$\hat{A}_1 = T_1 A T_1^{-1} \qquad \hat{B}_1 := T_1 B \qquad \hat{C}_1 := C T_1^{-1} \qquad \hat{D}_1 := D$$

which, by replacing $T_1 = T\mathcal{D}$, becomes:

$$\hat{A}_1 = T\mathcal{D}A\mathcal{D}^{-1}T^{-1} \qquad \hat{B}_1 = T\mathcal{D}B \qquad \hat{C}_1 = C\mathcal{D}^{-1}T^{-1} \qquad \hat{D}_1 = D \quad (6.29)$$

Therefore, the equivalent balanced state-space realization is:

$$(\hat{A}_1, \hat{B}_1, \hat{C}_1, \hat{D}_1) := (\mathcal{D}A\mathcal{D}^{-1}, \mathcal{D}B, C\mathcal{D}^{-1}, D) \quad (6.30)$$

The first step in understanding what the effect of the pre and post multiplications with the matrix $\mathcal{D}$ and its inverse is on the matrices A, B, C and D is investigating the actual form of the matrix $\mathcal{D}^{-1}$, when knowing that $\mathcal{D} = diag(\pm 1, \ldots, \pm 1)$.

Proposition 6.5.2. *If $\mathcal{D}$ =diag($\pm 1, \ldots, \pm 1$), then $\mathcal{D}^{-1} = \mathcal{D}$*

Proof. It is easy to see that, if the matrix M =diag($m_1, \ldots, m_n$), then the matrix M^{-1} exists and M^{-1} =diag($\frac{1}{m_1}, \ldots, \frac{1}{m_n}$). If setting $m_i = \pm 1$, one reconvers the desired conclusion. □

Therefore, taking into account proposition 6.5.2:

$$(\hat{A}_1, \hat{B}_1, \hat{C}_1, \hat{D}_1) := (\mathcal{D}A\mathcal{D}, \mathcal{D}B, C\mathcal{D}, D) \quad (6.31)$$

In one of the extreme cases, when all the diagonal elements of $\mathcal{D}$ are 1, the 4 matrices describing the state space realization of the LTI system remain unchanged. Hence, it makes sense studying the effect that a single -1 element on the diagonal of the $\mathcal{D}$ matrix has on the set $(\hat{A}_1, \hat{B}_1, \hat{C}_1, \hat{D}_1)$.

Proposition 6.5.3. *Assume that $\mathcal{D} = diag(d_1, \ldots, d_n)$ where $\exists!$ i s.t. $1 \leq i \leq n$, $d_i = -1$ and $d_j = 1, \forall j \neq i$. Then, if M is a square nxn matrix with elements m_{ij}:*

$$\mathcal{D}M = \begin{bmatrix} m_{11} & \ldots & m_{1n} \\ \vdots & & \vdots \\ -m_{i1} & \ldots & -m_{in} \\ \vdots & & \vdots \\ m_{n1} & \ldots & m_{nn} \end{bmatrix} \tag{6.32}$$

$$M\mathcal{D} = \begin{bmatrix} m_{11} & \ldots & -m_{1i} & \ldots & m_{1n} \\ \vdots & & \vdots & & \vdots \\ m_{n1} & \ldots & -mni & \ldots & m_{nn} \end{bmatrix} \tag{6.33}$$

$$\mathcal{D}M\mathcal{D} = \begin{bmatrix} m_{11} & \ldots & -m_{1i} & \ldots & m_{1n} \\ \vdots & & \vdots & & \vdots \\ -m_{i1} & \ldots & m_{ii} & \ldots & -m_{in} \\ \vdots & & \vdots & & \vdots \\ m_{n1} & \ldots & -mni & \ldots & m_{nn} \end{bmatrix} \tag{6.34}$$

The proposition 6.5.3 baiscally states that, if the diagonal matrix $\mathcal{D}$ has only one -1 element on the i^{th} row and column and 1 elements otherwise, then:

- A premultiplication with $\mathcal{D}$ is equivalent to a a matrix with elements with reversed signs on the i^{th} row, the other elements remain unchanged;
- A premultiplication with $\mathcal{D}$ is equivalent to a a matrix with elements with reversed signs on the i^{th} column, the other elements remain unchanged;
- A pre- and postmultiplication with $\mathcal{D}$ is equivalent to a a matrix with elements with reversed signs on the i^{th} row and column, the other elements remain unchanged (including the element on the i_{th} position in the diagonal, at the intersection between the i^{th} row and column).

Proposition 6.5.4. *Assume that $\mathcal{D} = diag(d_1, \ldots, d_n)$ where $\exists S \subseteq \{1, n\}$ such that $d_i = -1, \forall i \in S$ and $d_j = 1, \forall j \in \{1, \ldots, n\} - S$. Then, if M is a square nxn matrix with elements m_{ij} and $M' = \mathcal{D}M\mathcal{D}$, $M'' = \mathcal{D}M$ and $M''' = M\mathcal{D}$:*

$$\forall i \in S \text{ and } \forall j \in \{1, \ldots, n\}, j \neq i, \text{ it holds that: } m'_{ij} = -m_{ij}$$

$$\forall i \in \{1, \ldots, n\} \text{ and } \forall j \in S, j \neq i, \text{ it holds that: } m'_{ij} = -m_{ij}$$

$$\forall i \in S \text{ and } \forall j \in \{1, \ldots, n\} \text{ it holds that: } m''_{ij} = -m_{ij}$$

$$\forall i \in \{1, \ldots, n\} \text{ and } \forall j \in S \text{ it holds that: } m'''_{ij} = -m_{ij}$$

Proof. The proof can be done either by straighforward calculation or by induction. □

The reason why we employed this study here was to provide a way to change the species graph (built on the basis of the matrices A, B and C) of a given balanced system into an equivalent one, balanced as well, but also monotone. Proposition 6.5.4 helps in this aspects, as, by carefully choosing the matrix $\mathcal{D}$, one can break all the negative cycles from a network, by "spotting" the negative edges which need to be removed. Note however that the solution of pre- and postmultiplication with the matrix $\mathcal{D}$ is quite restrictive, as it flips the sign of full rows and columns. Therefore, this type of solution is mostly suited for very low-dimensional systems.

Chapter 7

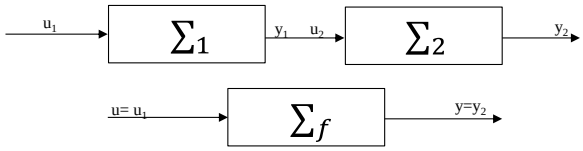
Interconnections

The last goal of this thesis was a brief excursion into the effect of the studied approximation techniques on the interconnections of two monotone modules. More specifically, the differences between these two scenarios were investigated:

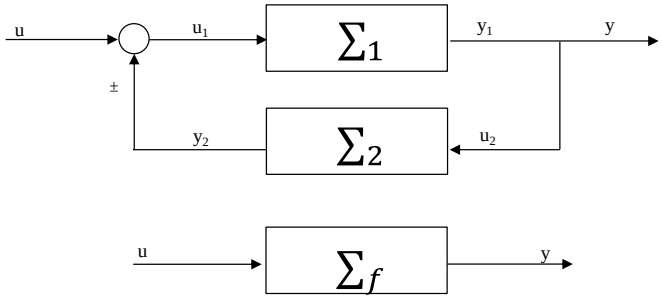
1. Interconnecting two approximated systems;
2. Directly approximating the system consisting of two interconnected monotone modules.

In systems theory, the term *interconnection* can be interpreted in different ways, out of which the two most commonly employed directions are: *cascade interconnections* and *feedback interconnections* (see figure 7.1). Without formally defining any of these interconnection methods for now, we will justify the fact that, in the present framework, systems were only coupled in cascade, by the claims that a cascade interconnection of monotone systems is still monotone, while a feedback interconnection, not necessarily. For a formal proof on the first claim, see [2], while for a discussion on the second, see [32].

In Chapters 5 and 6, we discussed in detail two very different approximation methods: nonlinear Hammerstein Wiener identification and linear balanced truncation. In the present case however, some of the inputs are time-dependent, as they were initially outputs of other systems. This makes the balanced truncation unapplicable when interconnecting systems. Therefore, we will here only use the nonlinear Wiener Hammerstein identification in studying interconnections.
Hence, each of the corresponding subsystems is identified on the basis of a Hammerstein Wiener structure, following the same procedure described in Chapter 5. As a conclusion to the performance evaluation analysis in the same chapter, we conducted our analysis here on 4 types of Hammerstein Wiener model structures, whose nonlinearities have different numbers of input and output units (5/10, 10/10, 1/10 and 5/5).
In concrete terms, two of the discussed I/O monotone examples: Example1 (system 4.2) and Example2 (system 4.8) will be cascade connected.



(a) Cascade interconnection of two systems.



(b) Feedback interconnection of two systems

Figure 7.1: Interconnections of two systems

7.1 Cascade Interconnections

Let Σ_1 and Σ_2 be two nonlinear systems:

$$\Sigma_1 : \begin{cases} \dfrac{\mathrm{d}x_1(t)}{\mathrm{d}t} = f_1(x_1(t), u_1(t)) \\ y_1(t) = g_1(x_1(t), u_2(t)) \end{cases} \tag{7.1}$$

$$\Sigma_2 : \begin{cases} \dfrac{\mathrm{d}x_2(t)}{\mathrm{d}t} = f_2(x_2(t), u_2(t)) \\ y_2(t) = g_2(x_2(t), u_2(t)) \end{cases} \tag{7.2}$$

with $x_1 : \mathbb{R}_+ \to \mathbb{R}^{n_1}$, $u_1 : \mathbb{R}_+ \to \mathbb{R}^m$, $y_1 : \mathbb{R}_+ \to \mathbb{R}^{p_1}$, $x_2 : \mathbb{R}_+ \to \mathbb{R}^{n_2}$, $u_2 : \mathbb{R}_+ \to \mathbb{R}^{p_1}$, $y_2 : \mathbb{R}_+ \to \mathbb{R}^p$.

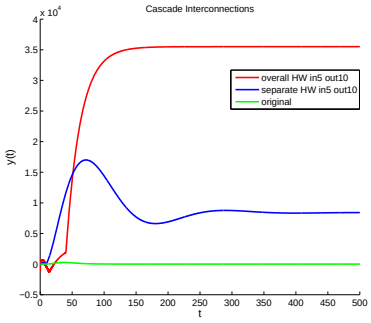
Definition 7.1.1. *Given the systems Σ_1 and Σ_2, with the form 7.1 and 7.2, respectively, the system Σ_c, with input $u_1 : \mathbb{R}_+ \to \mathbb{R}^m$ and output $y_2 : \mathbb{R}_+ \to \mathbb{R}^p$, is called the* cascade interconnection *of Σ_1 and Σ_2:*

$$\Sigma_c : \begin{cases} \dfrac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t), u_1(t)) \\ y_2(t) = g(x(t), u(t)) \end{cases} \tag{7.3}$$

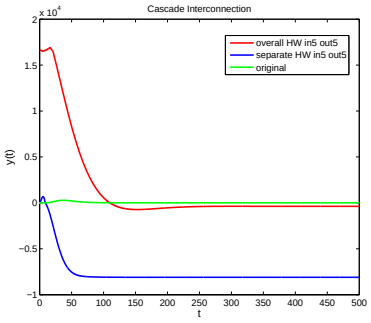
In our specific framework, the combined system Σ_c is the interconnection of, in order, systems 4.8 and 4.2. For each of the 4 Hammerstein Wiener structures, the output of the nonlinear cascade - which will be referred to as the *original* output, the approximated output of the nonlinear cascade - which will be referred to as the *overall* approximation and the cascade of the approximated systems - which will be referred to as the *separate* approximations) were validated under 4 constant inputs: 0.1, 0.4, 0.8 and 2. With very few exceptions, for the small inputs, neither of the overall or separate approximations showed similar behavior with the original output (see figure 7.2, together with the corresponding performance evaluation tables as dicussed in 3). However, in some of the cases, the similarity between the separate and overall approximations was higher than the similarity between either of them and the original output.

As a general observation, it was noticed, throughtout the estimation routine, that, under higher inputs, the both the overall and the separate approximations performed better. Figure 7.3 shows the most successfull approximations, together with the corresponding norm scores, obtained under the highest input $u = 5$.

As a last empirical observations, we state that, in principle, for all the 4 validation inputs used, the Hammerstein Wiener model structures with highest number of input and output nonlinearity units performed better in approximating the qualitative behavior of the original nonlinear cascade system (more specifically, the model structures with 5/10 and 10/10 input/output nonlinearity units performed better than the ones with 1/10 and 5/5). In this sense, see figure 7.4.



(a) plot: in5 out10; input $u = 0.4$



(b) plot: in5 out5; input $u = 0.4$

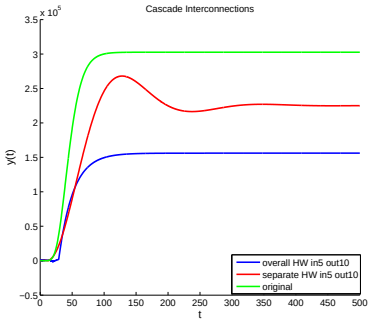
	L1 norm	L2 norm	Linf norm	Avg norm
Overall in5 out10	1.1513e+007	1.7303e+005	2.8825e+004	2.3026e+004
Separate in5 out10	1.5592e+007	2.3204e+005	3.5505e+004	3.1184e+004

(c) norms: in5 out10; input $u = 0.4$;

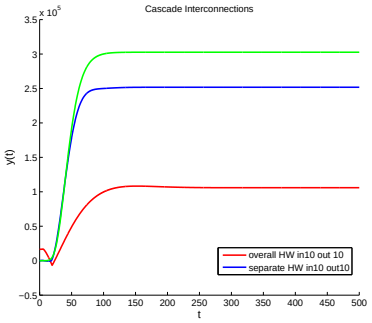
	L1 norm	L2 norm	Linf norm	Avg norm
Overall in5 out5	4.7116e+006	6.9545e+004	1.8869e+004	9.4233e+003
Separate in5 out5	1.5602e+007	2.3209e+005	3.5504e+004	3.1205e+004

(d) norms: in5 out5; input $u = 0.4$;

Figure 7.2: Cascade interconnection of systems 4.8 and 4.2, together with their performance ranking under the defined norms.



(a) plot: in5 out10; input $u = 5$



(b) plot: in10 out10; input $u = 5$

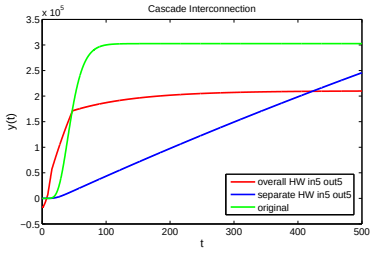
	L1 norm	L2 norm	Linf norm	Avg norm
Overall in5 out10	3.4703e+007	5.1885e+005	1.1552e+005	6.9406e+004
Separate in5 out10	6.7073e+007	9.8399e+005	1.5086e+005	1.3415e+005

(c) norms: in5 out10; input $u = 5$;

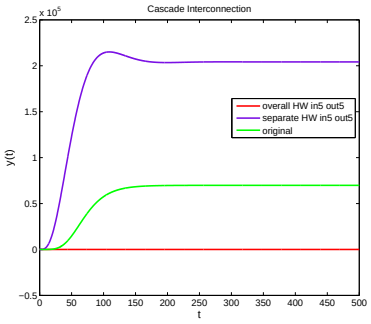
	L1 norm	L2 norm	Linf norm	Avg norm
Overall in10 out10	2.2322e+007	3.3299e+005	5.1177e+004	4.4645e+004
Separate in10 out10	9.0203e+007	1.3203e+006	2.0283e+005	1.8041e+005

(d) norms: in10 out10; input $u = 5$;

Figure 7.3: Cascade interconnection of systems 4.8 and 4.2, together with their performance ranking under the defined norms.



(a) plot: in1 out10; input $u = 5$



(b) plot: in5 out5; input $u = 0.8$

	L1 norm	L2 norm	Linf norm	Avg norm
Overall in1 out10	7.6323e+007	1.1895e+006	2.5890e+005	6.9406e+004
Separate in1 out10	4.5706e+007	6.6399e+005	1.1288e+005	9.1411e+004

(c) norms: in1 out10; input $u = 5$;

	L1 norm	L2 norm	Linf norm	Avg norm
Overall in1 out10	6.3655e+007	9.2860e+005	1.5741e+005	1.2731e+005
Separate in1 out10	2.9710e+007	4.4649e+005	6.9815e+004	5.9419e+004

(d) norms: in5 out5; input $u = 0.8$;

Figure 7.4: Cascade interconnection of systems 4.8 and 4.2, together with their performance ranking under the defined norms.

Conclusions

In this thesis, we aimed at defining the suitable computational and theoretical framework for approximating monotone modules. Next to the widely known L_p norms, we introduced two new (semi)norms for signal comparison: the *finite horizon average norm* and the *stability seminorm*. The latter was also justified to be more suitable for monotone systems approximation. All the used signal norms were discretized and adapted to the computational environment. Suitable monotone example systems, spanning over a wide range of features, were analyzed and implemented. On the basis of this defined theoretical and computational framework, we developed two conceptually different types of approaches in approximating the monotone systems:

1. The empirical approach (System Identification);
2. The theoretical approach (Balanced Truncation).

In the empirical approach, the example systems were identified using a Hammerstein Wiener nonlinear model structure, consisting of two static nonlinearities with a linear block in between. In the identification routine, suitable estimation and validation dataset had to be defined and, on their basis, models with different types of input and output nonlinearities were estimated. We used the knowledge of the actual system in tuning the parameters of the linear block. In the theoretical approach, the systems were first linearized around one of their equilibrium points. Then, they were transformed to an equivalent state space representation, in which the states were ranked according to the contribution they bring in the input-output behavior of the system. The last step was simply trimming the states for which this contribution was not significant and obtaining an approximation of the initial, nonlinear system.
The most interesting conclusions can be drawn from comparing these two different types of approaches. In principle, we can state that they both gave very good results on the studied monotone modules. However, some strong particularities of each method can be deduced:

- The system identification approach works on much narrower input spaces than the theoretical one: there are some areas in the input space where the approximations perform very well and some others on which they perform very poorly. In this sense, balanced truncation, being a theoretical method, has a more consistent behavior and also works for much broader zones of the input space;
- The system identification approach doesn't produce scalable models: in principle, when for two close input values, the qualitative behavior of the system should have been different, the prediction consists of similar outputs.
- The theoretical method offers bounds on the approximation error between the two systems, as well as it guarantees that the stability properties are kept.

- Balanced truncation offers simpler models as surrogates for the monotone modules, whereas the Hammerstein Wiener ones are more complex, hence they require more effort in simulation and analysis.
- Even if not explicitely treated in this thesis, the balanced truncation method is very sensitive to noise, as its basis is the linearization of the system around the steady state, which is valid as long as the deviation variables are small enough.

Having these directions clear, one can imagine as future improvements developments of more advanced theoretical methods for the study of the approximation of monotone systems. As a first idea, the Hammerstein Wiener model structure could serve as a basis in this investigation.

Bibliography

[1] D. Angeli, J. E. Ferrell, and E. D. Sontag. Detection of multistability, bifurcations, and hysteresis in a large class of biological positive-feedback systems. *Proc Natl Acad Sci U S A*, 101(7):1822–1827, February 2004.

[2] D. Angeli and E.D. Sontag. Monotone control systems. *IEEE Trans. Automat. Control*, 48(10):1684–1698, 2003.

[3] D. Angeli and E.D. Sontag. A note on multistability and monotone i/o systems. In *Proc. IEEE Conf. Decision and Control, Maui, Dec. 2003, IEEE Publications, 2003*, pages 67–72, 2003.

[4] D. Angeli and E.D. Sontag. Oscillations in i/o monotone systems. *IEEE Transactions on Circuits and Systems, Special Issue on Systems Biology*, 55:166–176, 2008. Preprint version in arXiv q-bio.QM/0701018, 14 Jan 2007.

[5] D. Angeli and E.D. Sontag. Translation-invariant monotone systems, and a global convergence result for enzymatic futile cycles. *Nonlinear Analysis Series B: Real World Applications*, 9:128–140, 2008.

[6] David Angeli and Eduardo D. Sontag. Monotone control systems, 2003.

[7] A. C. Antoulas, D. C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics*, 280:193–219, 2001.

[8] Christoph P. Bagowski, Jaya Besser, Christian R. Frey, and James E. Ferrell. The JNK cascade as a biochemical switch in mammalian cells: Ultrasensitive and all-or-none responses. *Current Biology*, 13(4):315–320, February 2003.

[9] Erik M. Boczko, Terrance G. Cooper, Tomas Gedeon, Konstantin Mischaikow, Deborah G. Murdock, Siddharth Pratap, and K. Sam Wells. Structure theorems and the dynamics of nitrogen catabolite repression in yeast. *Proceedings of the National Academy of Sciences of the United States of America*, 102(16):5647–5652, April 2005.

[10] E. N. Dancer. Some remarks on a boundedness assumption for monotone dynamical systems. *Proceedings of the American Mathematical Society*, 126(3):pp. 801–807, 1998.

[11] John Comstock Doyle, Bruce A. Francis, and Allen R. Tannenbaum. *Feedback Control Theory*. Prentice Hall Professional Technical Reference, 1991.

[12] G. Enciso. *Monotone Input/Output Systems, and applications to biological systems*. PhD thesis, Rutgers, The State University of New Jersey, 2005.

[13] German Enciso and Eduardo D. Sontag. Monotone systems under positive feedback: Multistability and a reduction theorem, 2004.

[14] Michael Green and David J. N. Limebeer. *Linear robust control*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[15] F. Harary. On the notion of balance of a signed graph. *Michigan Math J*, 2:143–146, 1953.

[16] T. Kailath. *Linear systems*. Prentice-Hall information and system sciences series. Prentice-Hall, 1980.

[17] H.-M. Kaltenbach, S. Constantinescu, J. Feigelman, and J. Stelling. Graph-based decomposition of biochemical reaction networks into monotone subsystems. In *Proceedings of WABI 2011, volume 6833 of Lect. Notes Bioinformatics, Springer*, pages 139–150, 2011.

[18] J.P. Keener and J. Sneyd. *Mathematical physiology*. Interdisciplinary applied mathematics: Mathematical biology. Springer, 1998.

[19] A. Laub, M. Heath, C. Paige, and R. Ward. Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. *Automatic Control, IEEE Transactions on*, 32(2):115 – 122, feb 1987.

[20] Lennart Ljung. *System identification: theory for the user*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1986.

[21] J. Lygeros and F. Ramponi. Lecture notes on linear system theory. ETH Zurich, 2010.

[22] Hirsch M and Smith HL. *Monotone dynamical systems. In: Handbook of differential equations, ordinary differential equations (second volume).* Elsevier, Amsterdam, 2005.

[23] Alex Megretski. Model order reduction. University Lecture, 2004.

[24] J.L. Melsa and D.G. Schultz. *Linear control systems*. McGraw-Hill series in electronic systems. McGraw-Hill, 1969.

[25] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26:17–32, 1981.

[26] C. Mullis and R. Roberts. Synthesis of minimum roundoff noise fixed point digital filters. *Circuits and Systems, IEEE Transactions on*, 23(9):551 – 562, sep 1976.

[27] Andrew Packard. Dynamic systems and feedback. University Lecture, 2002.

[28] Joseph R. Pomerening, Eduardo D. Sontag, and James E. Ferrell. Building a cell cycle oscillator: hysteresis and bistability in the activation of Cdc2. *Nature Cell Biology*, 5(4):346–351, March 2003.

[29] Derek Rowell. Analysis and design of feedback control system. University Lecture, 2002.

[30] William McC. Siebert. *Circuits,signals,and systems.* MIT Press, Cambridge, MA, USA, 1986.

[31] Hal L. Smith. *Monotone dynamical systems: An introduction to the theory of competitive and cooperative systems.* Providence, R.I: American Mathematical Society, 1995.

[32] Eduardo Sontag. Monotone and near-monotone biochemical networks. *Systems and Synthetic Biology*, 1(2):59–87, April 2007.

[33] Eduardo D. Sontag. *Mathematical control theory : deterministic finite dimensional systems.* Texts in applied mathematics, 6. Springer, 2nd edition, July 1998.

[34] Steven H. Strogatz. *Nonlinear Dynamics And Chaos: With Applications To Physics, Biology, Chemistry, And Engineering (Studies in Nonlinearity).* Studies in nonlinearity. Westview Press, 1 edition, January 2001.

[35] Hannu Toivonen. Robust control by state-space methods. University Lecture, 2002.

[36] Thomas Zaslavsky. Balanced decompositions of a signed graph. *J. Comb. Theory, Ser. B*, 43(1):1–13, 1987.