Universidade Federal do Rio de Janeiro
Escola Politécnica

CONTROLE TOLERANTE A FALHAS DESCENTRALIZADO E BALANCEADO PARA PLANTAS DE PRIMEIRA ORDEM

FERNANDA CRISTINA SERRÃO MATIAS

Projeto de Graduação apresentado ao Curso de Engenharia de Controle e Automação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientadores:
Prof. Amit Bhaya, Ph.D
Prof. Frederico C. Jandre, D.Sc

Rio de Janeiro, RJ - Brazil

2013

CONTROLE TOLERANTE A FALHAS DESCENTRALIZADO E BALANCEADO PARA PLANTAS DE PRIMEIRA ORDEM

FERNANDA CRISTINA SERRAO MATIAS

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO DE CONTROLE E AUTOMAÇÃO

Aprovado por:

Prof. Frederico C. Jandre, D.Sc.

Prof. Afonso Celso Del Nero Gomes, D.Sc.

Prof. Marcos Vicente de Brito Moreira, D.Sc.

Rio de Janeiro, RJ - Brasil

2013

Serrão Matias, Fernanda Cristina.

Controle Tolerante a Falhas Descentralizado e Balanceado para Plantas de Primeira Ordem

57 páginas

Projeto de Graduação UFRJ/ Escola Politécnica/Curso de Engenharia de Controle e Automação, 2013.

Orientadores:
Prof. Amit Bhaya, Ph.D
Prof. Frederico C. Jandre, D.Sc

1. Tolerância a falhas
2. Descentralizado
3. Controle Balanceado
4. Redundância
5. Consenso

I. Serrão Matias, Fernanda Cristina. IV. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia de Controle e Automação. V. Título.

Acknowledgement

First of all I would like to thank my parents, Marlene and Joaquim, who inspired, encouraged and fully supported me in every trial that came my way. Words can hardly express my gratitude to them.

I would like to offer my special thanks to Marcello Victorino for his support, encouragement and patience during those past months.

I would like to express my deep gratitude to Professors Amit Bhaya and Frederico Jandre, my tutors, for their patient guidance, enthusiastic encouragement and useful critiques of this work.

My experience as a graduate student would not have been the same without my class *Controle-T12*. Thank you for all my classmates which made this long experience much more pleasant.

Last but not least, I would like to thank the professors that made part of my graduation, teaching and inspiring me during all those years.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Controle e Automação.

CONTROLE TOLERANTE A FALHAS DESCENTRALIZADO E BALANCEADO PARA PLANTAS DE PRIMEIRA ORDEM

Fernanda Cristina Serrão Matias

2013

Orientadores:
Prof. Amit Bhaya, Ph.D
Prof. Frederico C. Jandre, D.Sc
Curso: Engenharia de Controle e Automação

Este trabalho propõe uma lei de controle baseada em um grupo de controladores trabalhando em paralelo com o objetivo de resolver o problema de regulação de uma planta de primeira ordem de modo que essa regulação seja tolerante a falhas nos controladores e que, adicionalmente, as ações de controle estejam divididas igualmente pelos controladores que estão em funcionamento. A validação do método proposto é feita através de simulações baseadas nas equações ordinárias diferenciais do sistema. Por fim, um estudo considerando plantas de primeira ordem com atraso é realizado com o intuito de analisar os resultados e uma solução é proposta para o novo sistema.
Palavras Chaves: Tolerância a falhas, Descentralizado, Controle Balanceado, Redundância, Consenso.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer.

FAULT TOLERANT DECENTRALIZED RELIABLE BALANCED CONTROL FOR FIRST ORDER PLANTS

Fernanda Cristina Serrão Matias

2013

Advisors:
Prof. Amit Bhaya, Ph.D
Prof. Frederico C. Jandre, D.Sc
Course: Engenharia de Controle e Automação

This work proposes a control law for a bank of controllers working in parallel in order to solve the regulation problem for a first order plant in such a way that the regulation is tolerant to controller failures and in addition, the control efforts are divided equally amongst functioning controllers. The validity of the proposed method is given by simulations based on the system ordinary differential equations. Finally, a study considering first order plants with time delay is presented and a solution is proposed for the new system.
Keywords: Fault-tolerance, Decentralized, Balanced control, Redundancy, Consensus.

Contents

List of Figures

List of Tables

List of abbreviations

MTTF - Mean Time To Failures

MTBF - Mean Time Between Failures

MPC - Model Predictive Control

DRBCP - Decentralized Reliable Balanced Control Problem

LTI - Linear Time Invariant

ODE - Ordinary differential equation

RHP - Right half plane positive real axis

List of Symbols

$\mathbb{R}$: Real numbers

$\mathbb{C}$: Complex numbers

$\mathbb{C}^-$:Left-half complex plane

$\mathcal{O}$: Observability matrix

$\mathcal{C}$: Controllability matrix

Chapter 1

Introduction

Recently a lot of attention has been given to fault-tolerant systems. The aim of fault tolerant control is to design a controller which is able to guarantee stability and satisfactory performance even if a sensor, actuator or other component begins to behave differently from what was assumed during the design process.

In 1997 Patton presented a systematic approach to design fault-tolerant control [17]. The planning and design is made up of three different stages:

1. Fault-tolerant System Requirement Analysis
2. Redundancy Design
3. Fault Accommodation Design

In the first stage an analysis is required to identify the possibility of failure for each component as well as their effects on the system. In the redundancy design stage, a location for each redundancy should be determined. Finally, the configuration of the controllers, sensors and actuators should be determined. Also in this last step, all the requirements to develop a controller reconfiguration in case a failure occurs should be determined.

In this work we assume that failures only occur in the actuator and/or controller components and that a fault-tolerant control is required to guarantee output regulation. It is assumed that n identical controllers are connected in parallel, with each one connected to its own actuator and the individual actuator signals are then added to produce the overall actuator signal to the plant. The objective of this work is to develop a control law which fulfils the requirements of regulation to the reference and equality between controllers in the sense of equal actuation by each one.

1.1 Motivation

The motivation for requiring balanced operation of the controllers comes from the consideration of mean time to failures (MTTF) and mean time between failures (MTBF). It is known that parallel redundancy, which is being used here, increases MTTF by a factor given by a partial sum of the harmonic series. As far as MTBF is concerned, if a controller operates at a certain percentage of its rated load, then its MTBF increases nonlinearly, in accordance with a exponential distribution [11], also motivating the use of a balanced control solution. Also, in theory, we have the well-known result that the MTBF of a system with k parallel redundant paths (without repairs) is proportional to the partial sum of the harmonic series where λ is the parameter of the continuous exponential density function:

$$MTBF(k) = \frac{1}{\lambda}\left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k}\right).$$

Since the harmonic series diverges, it follows that a system consisting of an infinite number of parallel paths, each of which has a mean time to failure of unit of time, would never fail [12].

Several studies can be found in the literature presenting fault-tolerant control approaches. Nevertheless, equality of controller efforts is not always sought. Furthermore, the usual design of multiple controller networks relies on communication between them and a central intelligence, which is not always possible or desirable.

1.2 Objective

This work aims to develop a **fault-tolerant decentralized control law**, which is capable of **signal tracking and disturbance rejection** for a certain class of plants. The goal is to implement several controllers in parallel, which do not have any communication between them to avoid the use of more components that could induce a higher probability of failure. It is also important to **divide the control action equally** between the controllers to ensure that none of them will be overloaded, leading to extension of device lifetimes. In this work it is considered that if a component fails, it could be repaired and added back to the system at some instant posterior to the fault. When this happens, a new distribution of the controller efforts is required in order to maintain equal individual actuator efforts, and it should be recalled that none of the controllers has knowledge of how many controllers are active or, indeed, of their individual actuation effort.

1.3 Description of the project

This introduction above presented, in a qualitative manner, the fault-tolerant control problem and the main concepts related to it. The objective of this work was also described as well as the motivation for this study.

Chapter 2 begins by establishing some theoretical definitions necessary for the comprehension of this project. Subsequently a literature review briefly discusses some papers related to fault-tolerant systems and balanced control laws. A recently proposed method called Control Allocation is presented. This method, designed to achieve balanced control in multiple actuator problems for aircrafts and wind turbines is the one which comes closest to providing a solution to the problem studied in this project.

Chapter 3 starts with an analysis of a linear first order plant in unity feedback configuration with a bank of identical linear controllers. A state space analysis of this system yields the important insight that the decentralized reliable balanced control (DRBC) problem cannot be solved in this linear context, and this is verified with some simulations. As a consequence, each linear controller is now endowed with a piecewise linear switched logic, in addition to an ordinary PI controller. This natural approach is shown to solve the DRBCP, although with some limitations.

Chapter 4 formulates a new approach to the DRBC problem. It describes the design and synthesis of a more sophisticated nonlinear switched control logic that is capable of equalizing controller efforts as well as regulating the output of the plant to its reference input. This chapter provides proofs of some of the basic stability results, although a complete proof is still missing. Simulations verify the theoretical results, the conjecture of stability, as well as the improvement in performance with respect to the piecewise linear switched logic studied in the previous chapter. Simulation studies of the sensitivity to parameter variations are also presented.

Chapter 5 studies the performance of the proposed DRBC for first order plants with delay. A study of first order plants with time delay is presented in simulations 5.4 to **??**. The aim of this chapter is to define the limitations of the proposed method as well possible applications.

The last chapter summarises the content of this project by presenting some conclusions and suggesting topics for future work.

Finally, a brief explanation about the computer program that generated the results is presented. This program was developed in Matlab and the code can be found in the Appendices.

Chapter 2

Review of the Literature

A literature search reveals a significant number of papers related to fault-tolerant and reliable control problems, as well as to consensus and balanced control problems. However the DRBCP as formulated in the previous chapter does not seem to have been studied before. In this section a brief summary of currently available literature is presented.

The DRBC problem seeks a decentralized control solution. In [16] Panagi presented a study of decentralized fault tolerant control of a class of interconnected nonlinear (but linearizable) subsystems. However, faults were considered to occur only in the subsystem instead of occurring in controller. The motivation of the paper is to propose a decentralized control law for which there is no need to redesign the control law if a fault occurs in the specified subsystems. The method proposed by the author is a decentralized adaptive approximation of a fault-tolerant control and is based on the principle that the controllers are able to identify a fault occurrence as well as to estimate the approximate effect of this fault in the dynamics system. The problem formulation permits occurrence of multiple faults at unknown times. This approach does not satisfy the condition imposed above, namely, that no controller should use any information about the other controllers.

Much research has been carried out on consensus methods, however, most papers on the subject assume that there is a fixed number of actuators/subsystems, or a given topology with information on neighbours. To solve the consensus problem, Model Predictive Control (MPC), is proposed in [6] where an artificial consensus trajectory is generated by a consensus algorithm and then each agent in the system follows this trajectory using MPC and respecting the given constraint. In [10], Kim et al. propose a control law to force output consensus for heterogeneous systems. The aim is to propose a method which can control the system output of an uncertain multi-agent system. To reach output consensus of heterogeneous multi-agent systems, a consensus of certain homogenous parts is achieved and internal models are used for control synthesis. As mentioned above, in most of the literature on the control consensus problem, information on the number of controllers/systems in the loop is required and this does not satisfy the "no-information" requirement of DRBCP.

Some other papers involving balanced control solutions, fault-tolerant systems [5] and reliable control [9] laws have been published, but all of these presuppose the use of more information than allowed in DRBCP. The literature most relevant for DRBC present problem is that on Control Allocation presented by Bodson and co-workers and it is presented in a little more detail in the following section.

2.1 Control Allocation

This new method is used for over-actuated systems, that is, when there are more control inputs than the outputs to be controlled. Control allocation is the problem of distributing control effort among multiple and redundant actuators. In brief, this can be thought of as the minimization of control effort by the minimization of the maximum actuator action. A survey about this new control technique can be found in [1].

In [4], Bodson converts the problem to a linear programming problem and the solution is based on the simplex algorithm. This solution was developed in order to solve the control allocation problem of an aircraft, for which the controlled variable is represented by a vector of deflection angles of control surface of the aircraft.

The ganging concept, which is the use of multiple redundant actuators, can be represented mathematically by:

$$u = G\nu$$

ν : pseudo effectors

u : actuator command vector

G : ganging matrix

The control allocation solution is introduced in the context of model reference control. The mathematical representation of the state-space model is given by (2.1) and (2.2).

$$\dot{x}_A = A_A x_A + Bu + d \tag{2.1}$$

$$y_A = C x_A \tag{2.2}$$

The state vector $x_A \in \mathbb{R}^n$, in the aircraft system, may represent the angle of attack, the pitch rate, the roll rate and the yaw rate. The output vector $y_A \in \mathbb{R}^q$ may contain the pitch rate, the roll rate and the yaw rate. The control input vector $u \in \mathbb{R}^p$ consist in the commanded actuator positions and the vector $d \in \mathbb{R}^n$ the forces and moments that disturb the aircraft system.

The desired dynamics for the reference model is given in (2.3). The vector r_M is defined as the reference input vector.

$$\dot{y}_M = A_M y_M + B_M r_M \tag{2.3}$$

The derivative of y_A is given by:

$$\dot{y}_A = C A_A x_A + CBu + Cd. \tag{2.4}$$

Equating $\dot{y}_M$ with $\dot{y}_A$, the following expression can be achieved:

$$CBu = -CAx_A - Cd + A_M y_A + B_M r_M. \tag{2.5}$$

To calculate the control output vector, it is required to solve a system of linear equations which has more unknowns than equations where u is constrained by (2.6).

$$u_{min,i} \leq u_i \leq u_{max,i}, \text{ for } i = 1, 2, ..., p \tag{2.6}$$

This constraints for u_i come from the actuator limitations.

Now that the problem has been specified, the control allocation problem can be defined as a combined error minimization and control optimization problem. The problem formulation for each case is presented below.

1. **Error minimization problem**

   Given matrix CB, it is desired to find u such that

   $$J = \|CBu - a_d\|$$

   is minimized and the vector u satisfies the constraint given by (2.6).

2. **Control minimization problem**

   If the solution of the error minimization problem, given by u_1 is not unique, control minimization is performed. For this problem the matrix CB and the solution u_1 are known. Given the vector u_p which represents some preferred actuator output, it is desired to find u (that satisfy the constraint condition in (2.6)) such that

   $$J = \|u - u_p\| \tag{2.7}$$

   is minimized and $(CB)u = (CB)u_1$.

The mixed optimization is the combination of the error minimization problem and the control minimization problem defined as follows:

Given matrix CB and vector u_p, find u such that

$$J = \|CBu - a_d\| + \xi\|u - u_p\|$$

is minimized, such that this solution is subject to $u_{min} \leq u \leq u_{max}$.

The term ξ is a weight or penalty that is added to a optimization problem to prioritise one of its terms. In this case, ξ is usually chosen to be small with the aim of prioritising the first term of J (error minimization).

Note that this solution requires a_d and u_p values, that is, the desired control action for each controller and the preferred values for the actuators. In the present problem these values are unknown since there is no information about how many controllers are functioning inside the unity feedback configuration. The allocation problem and the ganging concept is close to the DRBCP formulation except for the fact that the controllers must have global information about the status of other controllers.

Chapter 3

Problem formulation

3.1 Introduction: The Decentralized Reliable Balanced Control Problem (DRBCP)

This project studies the problem of reliable, balanced control of a plant in unity feedback configuration, using a bank of k identical controllers, denoted $S(P, C^k)$ (see Fig.3.1), to regulate the plant output to a constant reference input. A decentralized solution is required, in other words there is no communication between the controllers and thus no information about their functioning. The terms reliable and balanced are now made more precise.

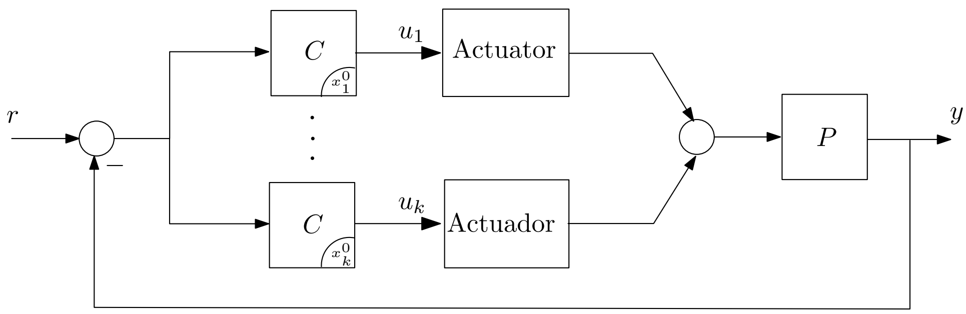


Figure 3.1: Plant with bank of k identical controllers and actuators in unity feedback configuration, denoted $S(P, C^k)$.

- The term *reliable* refers to the fact that the system $S(P, C^m)$ should remain stable and regulate the output to the constant reference input, for all m between 1 and $k-1$. In other words, even if $k - m$ controllers fail, the remaining m controllers guarantee the system stabilization.

- The term *balanced* refers to the fact that all the m active controllers are expected to participate equally in the regulation of the output to the constant input. In other words, it is required that all the m active controllers have the same output, for all values of m between 1 and k.

We will consider the simple case of $k = 2$ (Fig. 3.2), which is going to be the main focus of this project. Given the fact that the actual problem does not depends on the actuators configuration, from now on they will be considered as a unitary gain and so their representation will not be made explicit in the diagrams.

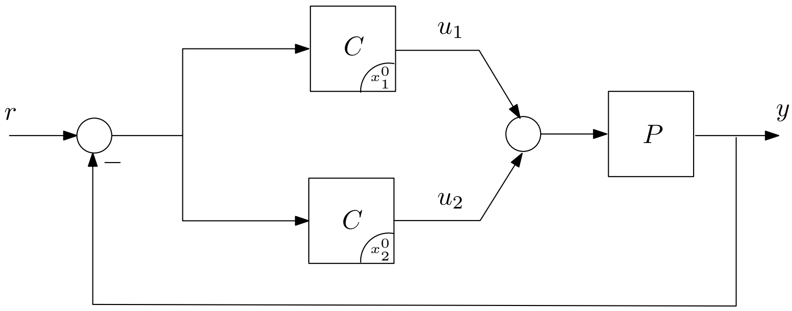


Figure 3.2: Plant with (bank of) 2 identical controllers in unity feedback configuration, denoted $S(P, C^2)$.

Temporarily assuming that the controllers are distinct and are denoted C_1 and C_2, it is easy to compute the transfer function between r and y, denoted h_{yr}:

$$h_{yr} = P(I + C_1P + C_2P)^{-1}(C_1 + C_2) \tag{3.1}$$

Assuming that the plant and controllers are single-input, single-output (SISO) and that $C_1 = C_2 = C$,

$$h_{yr} = \frac{2PC}{1 + 2PC}$$

so that, as expected, with identical controllers having the same initial conditions, the effect of 2 controllers in parallel is additive and equivalent to using the controller $2C$ with the plant P in unity feedback configuration. Note that this also means that if C stabilizes P with a certain gain margin, then the parallel combination of two identical controllers, in the configuration $S(P, C^2)$, stabilizes P with half this gain margin. It is also clear that the problem of balancing controllers only arises if there is an initial unbalance, either in the initial conditions, or because a failure occurs. In view of these observations, we arrive at the following statement of the general *decentralized reliable balanced control problem* (DRBCP):

> Suppose that the controller C stabilizes the LTI plant P and solves the regulation problem. Is it possible to connect k identical controllers C in parallel to plant P in unity feedback configuration (as in Figure 3.1), so that, if some nonempty subset of controllers, with possibly different initial conditions, is active at any given time, this subset stabilizes the plant P and solves the regulation problem, with all active controllers tending to attain the same output asymptotically?

Note that this problem looks superficially like the problem of requiring the controllers to attain consensus with regard to their outputs; however, a completely decentralized solution is being asked for, in which there is no communication between the controllers and the only information that is available to all the controllers is the error between the plant output and the reference input. In a typical consensus problem, a controller would exchange information with some 'neighboring' controllers in order to achieve consensus.

3.1.1 A naive switching approach to solving the RBCP for the case of two parallel controllers

Let us consider the two parallel controller case. A switching control solution, as depicted in Figure 3.3 appears to solve the RBCP.

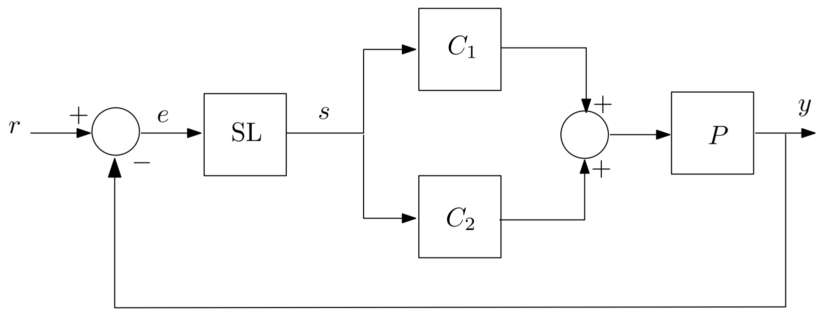


Figure 3.3: Plant P with 2 identical controllers plus switching logic SL defined in (3.2), in unity feedback configuration.

SL is the switching logic, defined as follows:

$$w = \begin{cases} e, & \text{if } e > \frac{r}{2} \\ \frac{e}{2}, & \text{otherwise} \end{cases} \tag{3.2}$$

Note however that this solution assumes that the controllers are either active or completely inactive and, furthermore, does not generalize to the case of three or more controllers.

In view of this, some questions that arise immediately are as follows:

- Is it possible to solve the DRBCP for an LTI plant using a bank of LTI controllers? If not, what kind of controller should be used?
- Is it possible to propose a solution to the DRBCP in which each active controller is unaware of the total number of active controllers and of the instants at which they become active?

The next section attempts to throw some light on the answer to the first question, using state space analysis, which allows the explicit consideration of different initial states for the two controllers.

3.2 State Space analysis of the two parallel controllers and plant in unity feedback configuration

Suppose that the identical controllers, with states called x_1 and x_2, have the state space representations (A, B, C, D), with initial states x_1^0 and x_2^0 and the input v is given by the system error. In other words, controller i, for $i = 1, 2$ has the representation:

$$\dot{x}_i = Ax_i + Bv \tag{3.3}$$

$$y_i = Cx_i + Dv \tag{3.4}$$

The outputs y_1 and y_2 are added to give the input u to the plant P, which has the state space representation (A_p, B_p, C_p), with its state denoted as x_p:

$$\dot{x}_p = A_p x_p + B_p u \tag{3.5}$$

$$y_p = C_p x_p \tag{3.6}$$

With the controllers and plant interconnected as in Fig. 3.4, the overall interconnected system,

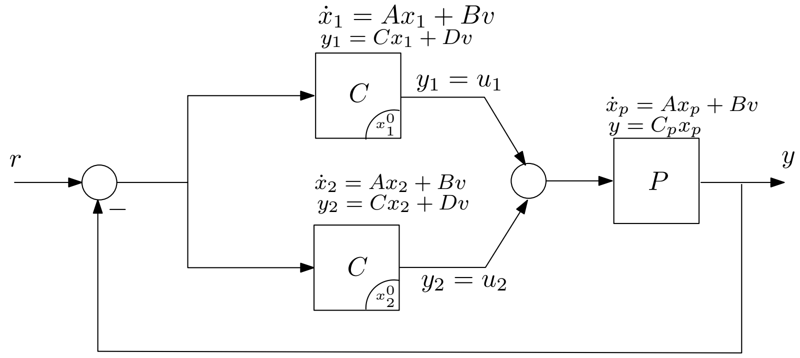


Figure 3.4: Plant P with 2 identical controllers, in unity feedback configuration, with state space representations specified.

which has state vector $x = [x_1, x_2, x_p]$, has the state space representation shown in 3.7 and 3.8.

$$\dot{x}_1 = Ax_1 + B(r - C_p x_p) = Ax_1 - BC_p x_p + Br$$
$$\dot{x}_2 = Ax_2 + B(r - C_p x_p) = Ax_2 - BC_p x_p + Br$$

$$\dot{x}_p = A_p x_p + B_p[Cx_1 + Dv + Cx_2 + Dv]$$
$$\dot{x}_p = A_p x_p + B_p C x_1 + B_p C x_2 + 2B_p D(r - C_p x_p)$$
$$\dot{x}_p = (A_p - 2B_p C_p D)x_p + B_p C x_1 + B_p C x_2 + 2B_p D r$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_p \end{bmatrix} = \begin{bmatrix} A & 0 & -BC_p \\ 0 & A & -BC_p \\ B_pC & B_pC & A_p - 2B_pC_pD \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_p \end{bmatrix} + \begin{bmatrix} B \\ B \\ 2B_pD \end{bmatrix} r \tag{3.7}$$

$$y = [0 \quad 0 \quad C_p] \begin{bmatrix} x_1 \\ x_2 \\ x_p \end{bmatrix} \tag{3.8}$$

The class of admissible controller faults $\mathcal{F}$ is defined as the one in which each controller can become active at any given time, but always with zero initial condition when it transitions from inactive to active and, of course, with the proviso that, at any given time, at least one controller is active.

In this state space setting, DRBCP (for two parallel controllers) can be formulated precisely as follows:

Given a plant P with state space representation (A_p, B_p, C_p), find a controller, if such exists, with state space representation (A, B, C, D), such that in the configuration of Figure 3.4, for all initial conditions x_1^0, x_2^0, and for all faults in the class $\mathcal{F}$, the following conditions are both satisfied:

- $y \to r$ for a positive constant reference input r, (i.e., reliable regulation).
- $|y_1(t) - y_2(t)| \to 0$ as $t \to \infty$, whenever both controllers are active, (i.e., balanced controllers).

Remark: Observe that, if an error e is defined as $y_1 - y_2$, then, from the state space equations above, it follows that

$$\dot{e}(t) = CA(x_1(t) - x_2(t)),$$

whence

$$\int_{t_1}^{t_2} \dot{e}(t)dt = e(t_2) - e(t_1) = CA \int_{t_1}^{t_2} (x_1(t) - x_2(t))dt, \tag{3.9}$$

which shows the effect of the initial conditions and evolution of the state on evolution of the error on the interval $[t_1, t_2]$.

3.2.1 Controllability analysis

From (3.7), the controllability matrix $\mathcal{C}$ for the overall system is calculated to be:

$$\mathcal{C} = \begin{bmatrix} B & AB - 2BB_pC_pD & \star \\ B & AB - 2BB_pC_pD & \star \\ 2B_pD & 2BB_pC + 2B_pD(A_p - 2B_pC_pD) & \square \end{bmatrix} \tag{3.10}$$

where

$$\star = B(A^2 - BB_pCC_p) - 2B_pD(ABC_p + BC_p(A_p - 2B_pC_pD)) - B^2B_pCC_p$$

$$\square = 2B(AB_pC + B_pC(A_p - 2B_pC_pD)) + 2B_pD((A_p - 2B_pC_pD)^2 - 2BB_pCC_p).$$

The first two block rows of $\mathcal{C}$ are identical, indicating that it is not full rank, so that the system is not controllable from the input r. This has the following interpretation. Even if it is permitted to choose the control input r arbitrarily, it may not be possible to steer from a state where $x_1(t_i) \neq x_2(t_i)$ to a final state where $x_1(t_f) = x_2(t_f)$ (and hence $y_1(t_f) = y_2(t_f)$). Of course, this interpretation holds *a fortiori* with a constant input r.

This controllability analysis gives a clue that the DRBCP may not admit a solution with a bank of LTI controllers. In other words, some nonlinear control may be necessary.

3.2.2 Observability analysis

From (3.7) and (3.8), the observability matrix $\mathcal{O}$ is calculated to be:

$$\mathcal{O} = \begin{bmatrix} 0 & 0 & C_p \\ C_pB_pC & C_pB_pC & C_p(A_p - 2B_pC_pD) \\ \star & \star & \square \end{bmatrix} \tag{3.11}$$

where

$$\star = AB_pCC_p + B_pCC_p(A_p - 2B_pC_pD)$$
$$\square = C_p(A_p - 2B_pC_pD)^2 - 2BB_pCC_p^2.$$

The first two block columns of $\mathcal{O}$ are identical, so that it is not full rank and the overall system is not observable. The interpretation of this is that, for zero input, the overall state x of the system cannot be deduced from observation of the output y alone.

3.2.3 Simulation results for linear controllers and plant

To illustrate the problem presented above and to analyse the methods (hereafter proposed), a computer program based on Euler numerical method for ODE solution was created on Matlab platform (see Appendix). The switching controller proposed above results in an ODE with discontinuous right hand side. Thus higher order methods to solve it, which depend on the existence of higher order derivatives, are inappropriate and the first order forward Euler method must be used. For this study, a system composed of a first-order process was used to explore the DRBCP where, for the simulations in this chapter, $\beta = 1$.

$$P(s) = \frac{\beta}{s + \beta}$$

Before starting the analysis of the *decentralized reliable balanced control problem*, a controller for a simple feedback configuration should be chosen and calibrated (recall that the controller should be able to stabilize and solve the regulation problem for a constant reference).

The *Internal Model Principle* [7] says that if it is desired that a system track a reference $r(t)$ or reject a disturbance $d(t)$, with dynamics given by the polynomial $T(s)$, then a controller that has a copy of this polynomial in the standard one degree-of-freedom control architecture is able to reject the effect of the disturbance or make the output track the reference. Therefore to control a first order plant and to guarantee reference regulation, which is a constant, a PI (Proportional-Integrative) controller is sufficient. To calibrate the controller gains a system composed of a PI controller and the chosen plant is considered:

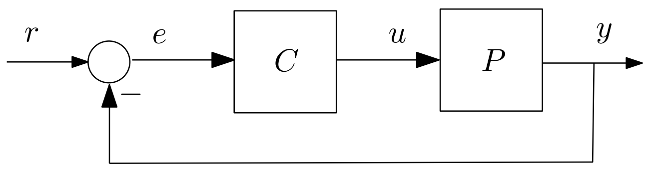


Figure 3.5: LTI system containing a PI controller and the plant in unity feedback configuration.

The PI controller expression is represented by 3.12.

$$C(s) = K_p e(t) + K_I \int_{t_i}^{t_f} e(\tau) d\tau \tag{3.12}$$

The Root Locus was chosen to obtain a quick analysis of the system stability (Fig. 3.5). From Fig. 3.6, it is easy to observe that all root locus branches are contained in the left half plane,

$$\mathbb{C}^- := \{z \in \mathbb{C} | \Re(z) < 0\}$$

i.e., the LTI system is stable for any system gain.

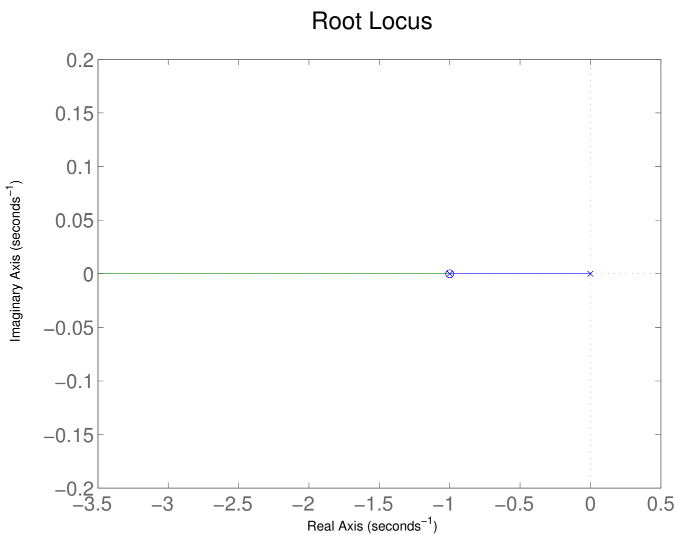
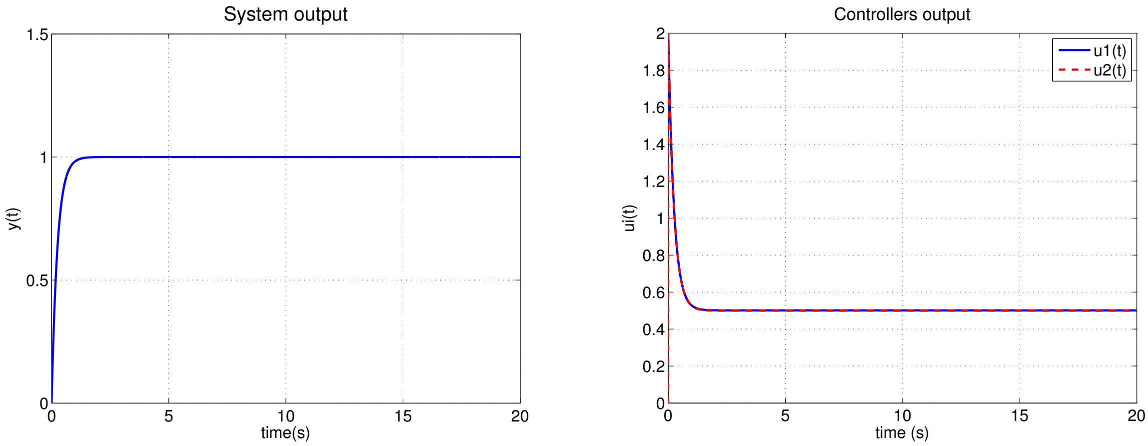


Figure 3.6: Root Locus for the system given by a first order plant and a PI controller in unity feedback configuration.



(a) System output $y_p(t)$ (b) Controllers output $u_1(t)$ and $u_2(t)$.

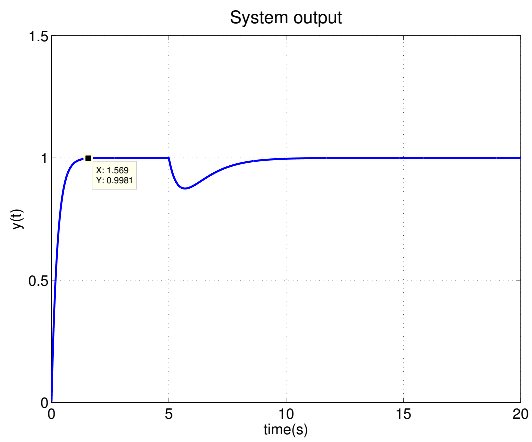
Figure 3.7: Simulation results for a feedback system with two controllers (initialized with null initial conditions) in parallel connected to a first order plant. Note that $u_1(t) = u_2(t)$ for all t.

The chosen proportional and integrative gains as well as the plant parameters are in Table 3.1. The given reference is considered as a constant.

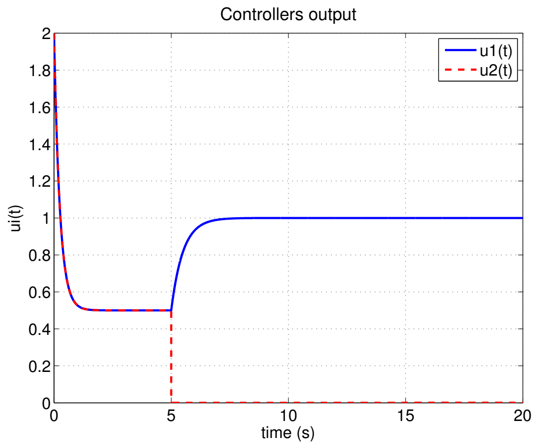
Table 3.1: System parameters

	Plant	Controller	
Parameters	β	K_P	K_I
Value	1	2	2

Once these system parameters are chosen, the simulation for two PI controllers placed in parallel and connected to a first-order plant can be analysed. In Figure 3.7a and 3.7b, both controllers are initialized at $t = 0s$ with null initial conditions. From simulation results one can observe a balance between controller outputs. However, Figure 3.8a and 3.8b depict what happens when a failure occurs in the second controller. In this new scenario, when C_2 zeros its output, C_1 takes on the control of system and guarantees tracking of the reference. At $t = 15s$ the controller C_2 is reconnected to the system with null initial conditions. However it is not able to act on the system, since the error signal that drives it and its initial condition are null (see the equations from the controller state space representation). This preliminary analysis of the LTI system shows that by using a linear controller, tracking is guaranteed, however for different initial conditions, the controllers could not reach the same levels of effort.



(a) System output $y_p(t)$



(b) Controllers output $u_1(t)$ and $u_2(t)$.

Figure 3.8: Simulation results for a feedback system with two controllers in parallel connected to a first order plant. A fault in C_2 occurs at $t = 5s$ (red dashed line). At $t = 15s$ C_2 is fixed and returned to the system with null initial conditions.

3.3 Analysis of the naive switching approach in state space

The results presented above show the necessity of a nonlinear logic for DRBCP. Considering the approach given by Figure 3.3, the new state space representation is presented in two different cases:

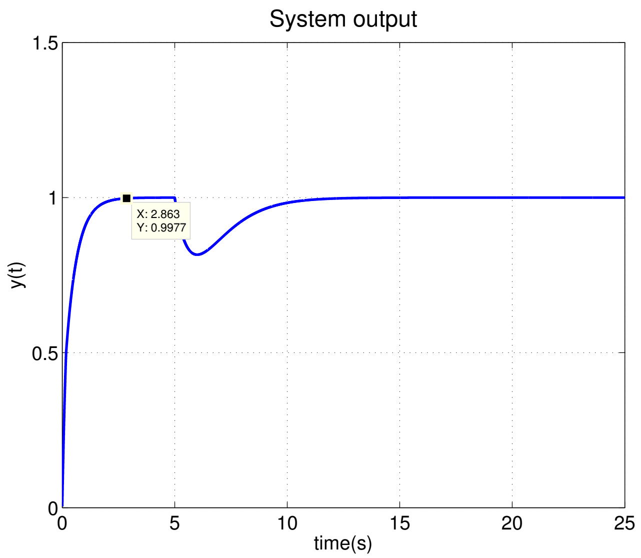
- For $e \geq 0.5r$: in this case, the system state space is given by (3.7) and (3.8).
- For $e < 0.5r$ for which the state space representation is given by the following expression:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_p \end{bmatrix} = \begin{bmatrix} A & 0 & -(BC_p)/2 \\ 0 & A & -(BC_p)/2 \\ B_pC & B_pC & A_p - B_pC_pD \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_p \end{bmatrix} + \begin{bmatrix} B/2 \\ B/2 \\ B_pD \end{bmatrix} r \tag{3.13}$$

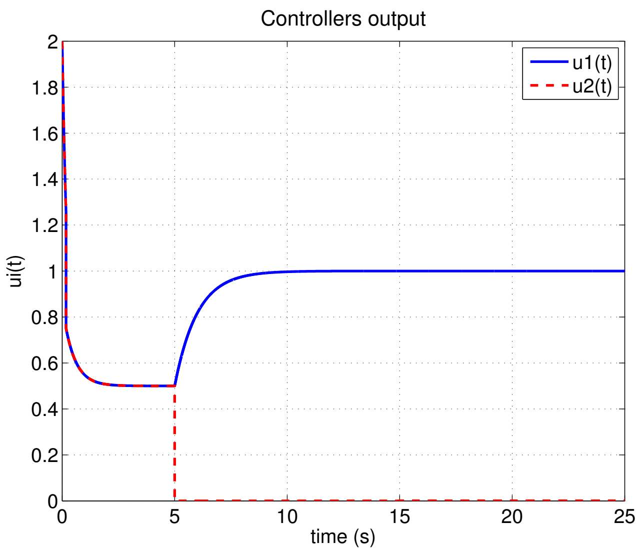
$$y = [0 \quad 0 \quad C_p] \begin{bmatrix} x_1 \\ x_2 \\ x_p \end{bmatrix} \tag{3.14}$$

From the state representation, it is easy to see that the controllability matrix is not full rank (note that it is the same case as presented for the LTI system, and the two first block rows from $\mathcal{C}$ are identical) which means that for $x_1(t) \neq x_2(t)$, the final state $x_1(t_f) = x_2(t_f)$ is never reached.

An analysis can also be made by looking at the piecewise linear switching logic equation in (3.2). The controller output y_i is calculated based on the error variable only. This means that when the controllers have different initial conditions, the control output decreases/increases by the same rate, and so, the difference between y_1 and y_2 remains the same. The results presented in Fig.3.9 illustrate the system behaviour when using the naive controller. Note that the convergence time for this system is two times slower than for the linear case. This behaviour comes from the fact that now, for errors smaller than $r/2$, the controllers gains are multiplied for a factor with value 0.5.



(a) System output $y_p(t)$



(b) Controllers output $u_1(t)$ and $u_2(t)$.

Figure 3.9: Simulation results for the naive solution with two controllers. A fault in C_2 occurs at $t = 5s$. At $t = 15s$ C_2 is fixed and added to the system with null initial conditions.

Chapter 4

Decentralized Reliable Balanced Control Law

The new proposal presented in Figure 4.1 is a nonlinear improvement on the piecewise linear logic proposed above and it is now called K_{ci}. The aim of this new logic is to solve the consensus problem between y_1 and y_2 without affecting the reference tracking.

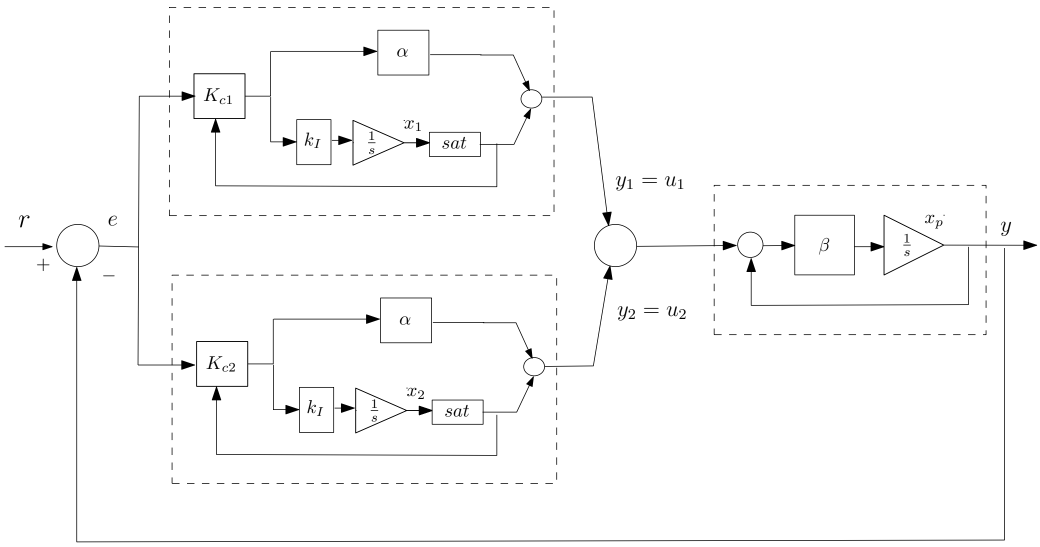


Figure 4.1: Plant P with 2 identical controllers, each containing (nonlinear) switching logic K_{ci} defined in (4.1), in unity feedback configuration.

The Switched Gain Law is represented by (4.1),

$$K_{ci} = \begin{cases} 1, \text{ if } e \geq 0 \\ 1 + \mu \frac{x_i(t)}{x_{max}}, \text{ if } e < 0 \end{cases} \tag{4.1}$$

where μ is a constant and it is constrained by $\mu > 1$ and x_{max} represents the physical system limitation (saturation of the integrator).

The purpose of this asymmetry between positive- and negative-error gains is to ensure that, if the error falls below zero (either because of a disturbance in the process or by a change in the reference), whenever a controller is more stressed than the others, both its integrator state and its overall output is the fastest to fall.

4.1 Analysis of the Decentralized Reliable Balanced Control Law

The system in Fig.4.1 is represented by the set of ordinary differential equations (ODEs) listed in (4.2). The nonlinear logic K_{ci} is described by the switching function $s(\cdot,\cdot)$ as defined in (4.3) and the function $sat(\cdot)$ represents the saturation of the integrator.

$$\begin{aligned}
\dot{x}_1 &= s(e, x_1) \\
y_1 &= \alpha s(e, x_1) + sat(x_1) \\
\dot{x}_2 &= s(e, x_2) \\
y_2 &= \alpha s(e, x_2) + sat(x_2) \\
\dot{x}_p &= \beta(y_1 + y_2 - x_p) \\
y &= x_p \\
e &= r - x_p
\end{aligned} \tag{4.2}$$

$$s(e, x_i) = \begin{cases} e, \text{ if } e \geq 0 \\ 1 + \mu \frac{x_i}{x_{max}}, \text{ if } e < 0 \end{cases} \tag{4.3}$$

Thus, for the full system, assuming that we are in the linear region of the controller saturation function ($sat(x_i)$), the state space representation is given by the two cases listed below.

1. For $e \geq 0$, which represents a linear system:

$$\begin{aligned} \dot{x}_1 &= r - x_p \\ y_1 &= \alpha(r - x_p) + x_1 \\ \dot{x}_2 &= r - x_p \\ y_2 &= \alpha(r - x_p) + x_2 \\ \dot{x}_p &= \beta(y_1 + y_2 - x_p) \\ y &= x_p \end{aligned} \tag{4.4}$$

2. For $e < 0$, a non linear system:

$$\begin{aligned} \dot{x}_1 &= (1 + \mu\tfrac{x_1}{x_{max}})(r - x_p) \\ y_1 &= \alpha(1 + \mu\tfrac{x_1}{x_{max}})(r - x_p) + x_1 \\ \dot{x}_2 &= (1 + \mu\tfrac{x_2}{x_{max}})(r - x_p) \\ y_2 &= \alpha(1 + \mu\tfrac{x_2}{x_{max}})(r - x_p) + x_2 \\ \dot{x}_p &= \beta(y_1 + y_2 - x_p) \\ y &= x_p \\ e &= r - y \end{aligned} \tag{4.5}$$

The two problems formulated below are required to be solved in order to analyse the new proposal:

Problem 1: In steady state, i.e. once the error has converged to zero, small perturbations in x_1, x_2 and x_p should be rejected, in the sense that in the presence of these perturbation:

a) e should return to zero and

b) $y_1 \to y_2$.

Problem 2: After reaching steady state,

a) if C_i drops out $(y_i \to 0)$, then e should return to zero.

b) if y_i becomes different from y_j (step disturbance), then $y_i(t) \to y_j(t)$, as $t \to \infty$.

4.1.1 Analysis in state space

For positive errors, the system state space representation (A, B, C) is given by:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_p \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ \beta & \beta & -2\alpha\beta \end{bmatrix}}_{\text{A}} \begin{bmatrix} x_1 \\ x_2 \\ x_p \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 2\alpha\beta \end{bmatrix} r \tag{4.6}$$

$$y = [0 \quad 0 \quad 1] \begin{bmatrix} x_1 \\ x_2 \\ x_p \end{bmatrix} \tag{4.7}$$

A stability analysis can be made by calculating the eigenvalues of the matrix A from the state space representation:

The characteristic polynomial of A is defined by:

$$\begin{aligned} a(s) &\triangleq \det(sI - A) \\ &= s^n + a_1 s^{n-1} + .. + a_{n-1}s + a_n \end{aligned}$$

and the roots of the polynomial are referred to as eigenvalues $\{ \lambda_i \}$ of A:

$$a(s) = (s - \lambda_1)(s - \lambda_2)...(s - \lambda_n).$$

The system is said to be stable if its eigenvalues lie in the left half plane (i.e. have negative real parts).

since the roots will be on the negative complex plane.

By calculating the characteristic polynomial of A from (4.7), we can analyze the stability of the transfer function from r to y.

$$a(s) = s(s^2 + 2\alpha\beta s + 2\beta) \tag{4.8}$$

The eigenvalues of A are given in (4.9):

$$\lambda = \left\{0, \frac{-2\alpha\beta \pm \sqrt{4\alpha\beta^2 - 8\beta)}}{2}\right\} = \{0, -\alpha\beta \pm \sqrt{\beta(\alpha^2\beta - 2)}\}. \tag{4.9}$$

Since the stability was guaranteed for the linear case, an analysis of the system stability for the zero eigenvalue correspond to the eigendirection $\begin{bmatrix}1 & -1 & 0\end{bmatrix}^T$, which means that perturbations in this eigendirections neither grow nor decay to zero, whereas perturbation in the other two eigendirections, which correspond to the two eigenvalues, do decay to zero. Thus, a kind of partial stability holds for positive values of e.

4.1.1.1 Local stability analysis around the desired equilibrium point

For the actual problem, it is desired that the system stabilizes at the equilibrium point given by:

$$\{x_1, x_2, x_3\} = \left\{\frac{r}{2}, \frac{r}{2}, r\right\}.$$

A linear approximation around this point is made by using the Jacobian matrix defined as:

$$J \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

From (4.5), we have:

$$J = \begin{bmatrix} (r - x_p)\mu & 0 & -(1 + \mu x_1) \\ 0 & (r - x_p)\mu & -(1 + \mu x_2) \\ \beta\alpha(r - x_p)\mu & \beta\alpha(r - x_p)\mu & -\beta\alpha[\mu(x_1 + x_2) + 2] - \beta \end{bmatrix}. \tag{4.10}$$

By making the substitutions, $\left(x_1, x_2, x_p\right) = \left(\frac{r}{2}, \frac{r}{2}, r\right)$, the Jacobian matrix becomes:

$$J = \begin{bmatrix} 0 & 0 & -(1 + \frac{\mu r}{2}) \\ 0 & 0 & -(1 + \frac{\mu r}{2}) \\ 0 & 0 & -\beta\alpha(\mu r + 2) - \beta \end{bmatrix} \tag{4.11}$$

and its eigenvalues are given by: $\lambda = \{0, 0, -\beta\alpha(\mu r + 2) - \beta\}$. Once again, there is neutral stability in the eigendirections $\begin{bmatrix}1 & 0 & 0\end{bmatrix}^T$, $\begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$ and stability in the third eigendirection.

It is important to observe that there are, in fact, infinitely many other equilibrium points: when $e = 0$ (i.e. $r = x_p$), then $y_i = x_i$, $i = 1, 2$ and so all x_i such that

$$x_1 + x_2 = x_p = r$$

are then points s.t. $(\lambda, r-\lambda, r)$ are equilibrium points for $\lambda \in [0, r]$. Geometrically, the equilibrium points are those given by the intersection between the planes

$$x_p = r$$
$$x_1 + x_2 = r.$$

Figure 4.2 represents these planes in (x_1, x_2, x_p) space. All these points have the same stability characteristics as $\left(\frac{r}{2}, \frac{r}{2}, r\right)$.

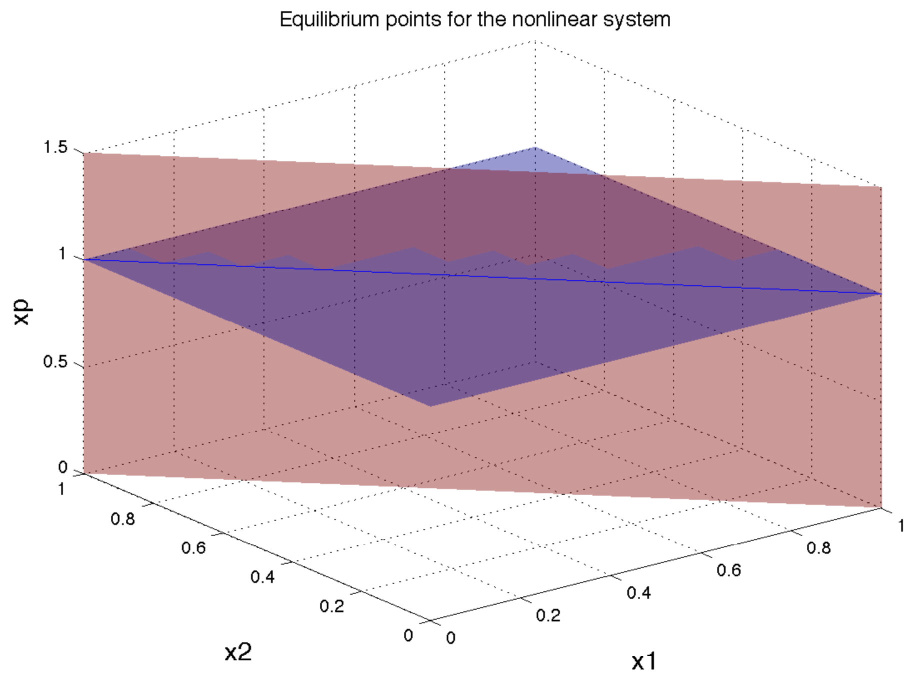


Figure 4.2: The line (in blue) consisting of the intersection between two planes represents the set of all equilibrium points for the nonlinear system.

In Figure 4.3 the (x_1, x_2) space is shown. Note that there are infinitely many points that satisfy the equation $x_1 + x_2 = r$, however, for the present problem, it is desired that the system reaches the equilibrium point given by $x_1 = x_2$.

Since all these points have the same stability characteristics, we can already conjecture the necessity of an external excitation to balance the controller states in case they arrive at points of the type $(\lambda, r-\lambda, r)$.

Now it would be important to find a region of attractivity around this set of equilibrium points, by using the Lyapunov Direct Method.

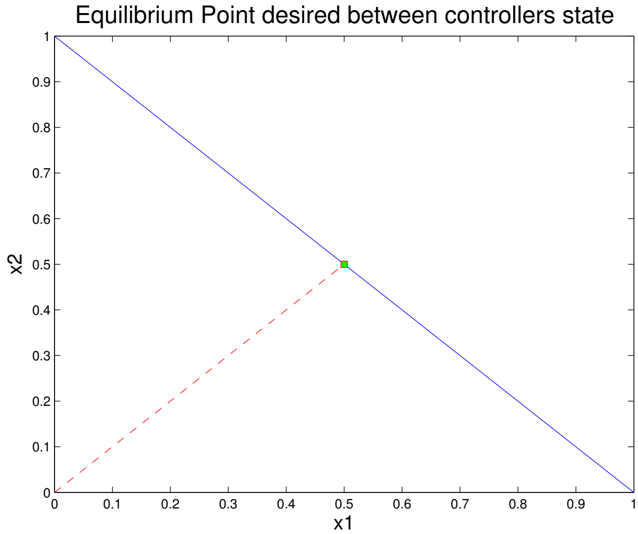


Figure 4.3: Desired equilibrium point, in green represented by the intersection between the two lines of balanced control and reference tracking.

A natural candidate for a Lyapunov function is:

$$V(e,z) = 0.5e^2 + 0.5z^2 \tag{4.12}$$

where $z = x_1 + x_2 - r$. For the equilibrium point given by $e = 0$ and $z = 0$ ($x_1 + x_2 = r$), the function $V(e,z)$ is null. From Lyapunov Direct Method, $V(e,z)$ is said to be a Lyapunov function if $V(t, e(t), z(t))$ is continuous and positive definite and if

$$\frac{d}{dt}\big(t, e(t), z(t)\big) = \dot{V}\big(t, e(t), z(t)\big) \le 0.$$

For the function chosen in (4.12), which is positive definite and continuous, we can attempt to show that $\dot{V}\big(t, e, z\big)$ is negative definite, and so that V can be defined as a Lyapunov function:

$$\dot{V}(t, e, z) = e\dot{e} + z\dot{z}$$

where,

$$e = r - x_p$$
$$\dot{e} = -\dot{x}_p$$
$$z = x_1 + x_2 - r$$
$$\dot{z} = \dot{x}_1 + \dot{x}_2$$
$$\dot{x}_i = \left(1 + \mu \frac{x_i}{x_{max}}\right)(r - x_p)$$

By making the substitutions and some simplifications we arrive at the following expression:

$$\dot{V} = -2\beta\alpha e^2 + \left(\frac{\mu}{x_{max}}\right) z^2 e - \underbrace{\beta\alpha\left(\frac{\mu}{x_{max}}\right)(z+r)e^2}_{t_3} + \underbrace{ze\left[2 + \left(\frac{\mu}{x_{max}}\right) r - \beta\right]}_{t_4} + \beta e(x_p - r). \tag{4.13}$$

The two first terms from (4.13) are negative as well the last term (by definition, $e = r - x_p$). For $z > 0$, the term t_3 is always negative and t_4 is negative if

$$\beta < 2 + \left(\frac{\mu}{x_{max}}\right) r.$$

For $z < 0$, t_3 is positive if $|z| > r$ and negative if $|z| < r$ and t_4 is negative only if

$$\beta > 2 + \left(\frac{\mu}{x_{max}}\right) r.$$

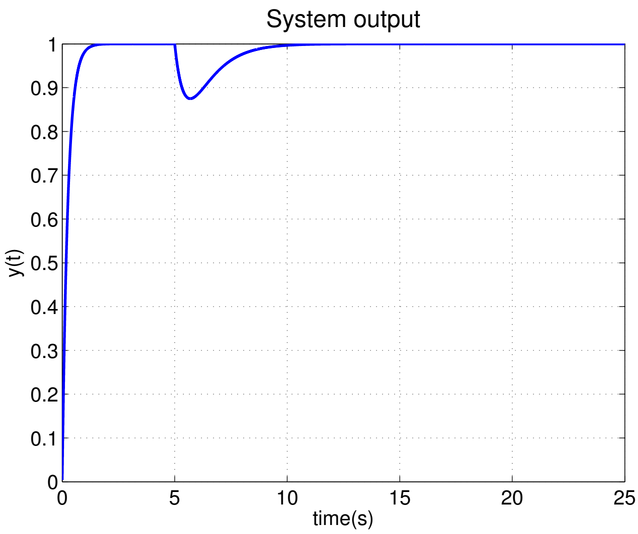
From this analysis we can see that the chosen function $V(e, z)$ is not a Lyapunov function and so, it is not possible to find a stability region with this particular function.

4.1.2 Simulation results for the Decentralized Reliable Balanced Control

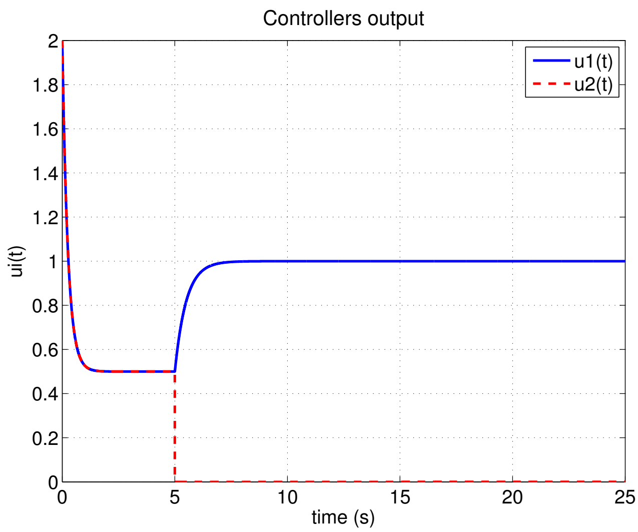
In this section some results are presented with the objective of testing the validity of this solution (for the chosen plant) and to point out some weaknesses. The chosen scenario is the same as presented in the other simulations. The system parameters are listed in Table 4.1.

Figure 4.4 illustrates (a) the system output and (b) the controller outputs. Note that the error is always positive and so the switched gain never reaches the nonlinear region given by:

$$K_{ci} = 1 + \mu \frac{x_i(t)}{x_{max}}.$$



(a) System output $y_p(t)$



(b) Controllers output $u_1(t)$ and $u_2(t)$.

Figure 4.4: Simulation results for the Decentralized Reliable Balanced Control solution with two controllers. A fault in C_2 occurs at $t = 5s$. At $t = 15s$ C_2 is fixed and returned to the system with null initial conditions.

Table 4.1: System parameters for the Decentralized Reliable Balance Control

	Plant	Controller			Switched Gain
Parameters	β	α	K_I	x_{max}	μ
Value	1	2	2	1	10

This behaviour suggests using a periodic disturbance signal added to the error signal in order to make it negative and force the system to enter the nonlinear case. By choosing a zero-mean square-wave disturbance, we seek a solution that allows the controllers to asymptotically approach a distribution of equal effort, while asymptotically tracking the mean reference. The only disadvantage of using such a persistent excitation is that the system output will never be exactly equal to the reference value, because of the presence of ripple from the zero-mean square-wave.

4.1.3 Simulation results for the Decentralized Reliable Balanced Control by adding a square wave to the error

The analysis presented in Chapter 3 revealed the necessity of nonlinearity to solve the RBCP. A solution given in this chapter was based on a switched logic with nonlinear gain. From the results above we see that a periodic disturbance added to the system error is required (see Fig. 4.5) to guarantee that K_{ci} changes each time it is used.

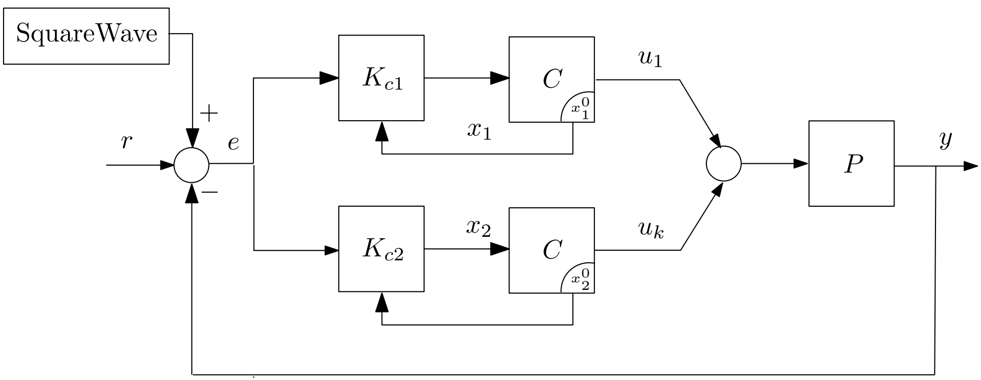


Figure 4.5: Plant P with 2 identical controllers, each containing (nonlinear) switching logic K_{ci} defined in (4.1), in unity feedback configuration. A zero-mean square wave disturbance is added to the error to force the system to enter the nonlinear case.

Figure 4.6 and 4.7 presents the results by applying the *decentralized balanced redundant control* and a zero-mean square wave added to the error, with an amplitude and frequency given by $A = 0.01$ and $F = 2$ respectively. With the aim to show with more details the system behavior, the graphs in Fig. 4.8 presents the same results but with zoom applied at $t = 0s$ to $t = 40s$.

The results show that when the controller C_2 is out of service, the controller C_1 is capable of tracking the reference by increasing its control action. At $t = 15s$, when the controller C_2 is added back to the system with null initial conditions, the RBC leads the controller outputs to the equilibrium given by $x_1 = x_2 = 0.25$. The controller state behaviour is presented in Figure 4.7a. In Fig. 4.7b the system states are represented in a 3D plot.

In the long term, the control action u_i is given by:

$$u_i = \frac{u_T}{n - m}$$

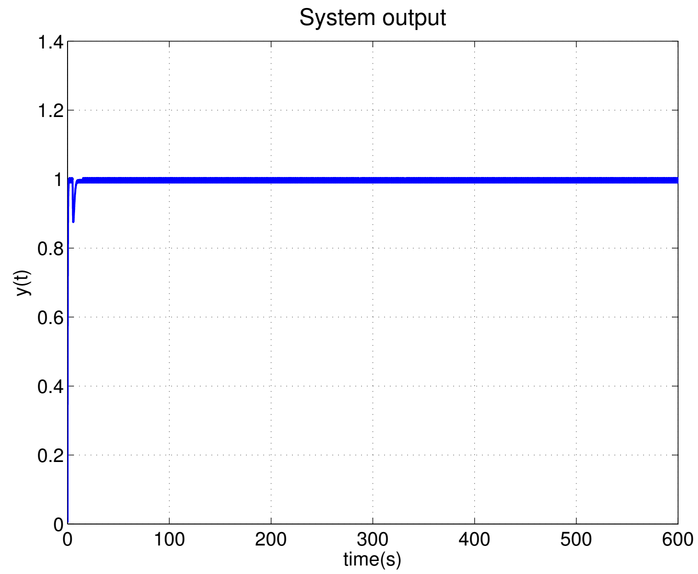
where u_T is the control action necessary to guarantee tracking of the reference, n is the total number of controllers in the system and m the number of controllers which has failed (recall that it is assumed that the controllers are either active or completely inactive).

By implementing the controller failures at different times in the simulation, a study was made in which failures of C_2 could even occur during the transient period, when the system is regulating to its steady state. In Figure 4.9a the difference between the controller outputs is shown and in Figure 4.9b the system error is calculated (In Fig. 4.10a and Fig 4.10b, a zoom is used in the begining of the simulation to illustrate with more details the results). This shows that, independently of when the failure occurs, the remaining controller is able to track the reference and when C_2 returns to activity, the controller outputs tend to reach the equilibrium $u_1(t) = u_2(t)$.

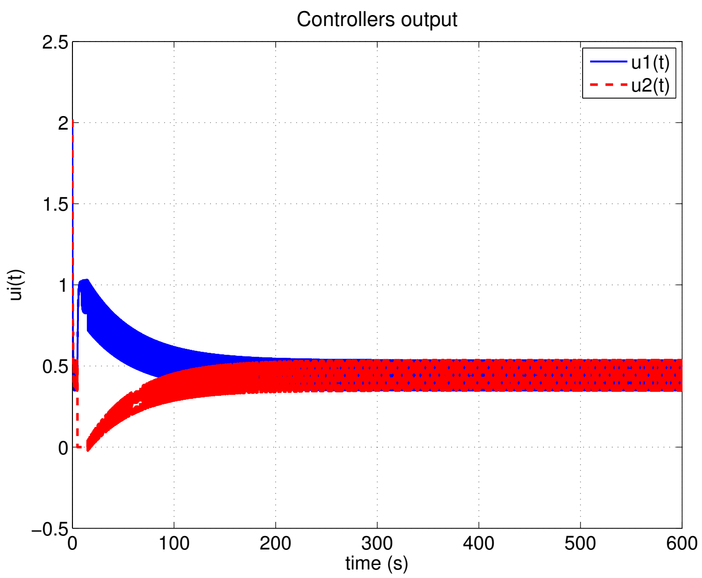
4.1.4 Study of the effect of the square wave amplitude and frequency

It is evident that the amplitude and frequency parameters from the square wave can influence the system output, and this is the objective of study in this section.

The most notable consequence of the square wave persistent excitation (p.e.) is the appearance of ripples in the system output. It is obvious that the greater the amplitude of the p.e., the greater the ripple will be. But what can we say about the system convergence time? And

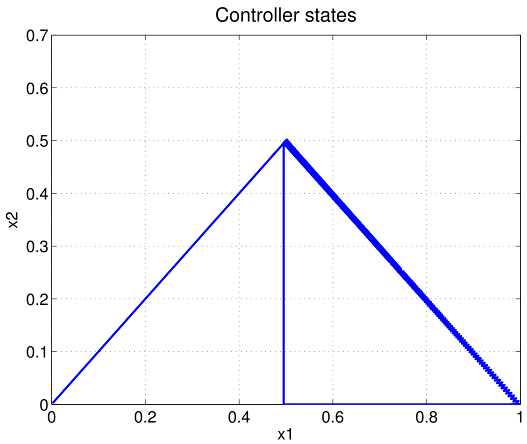


(a) System output $y_p(t)$

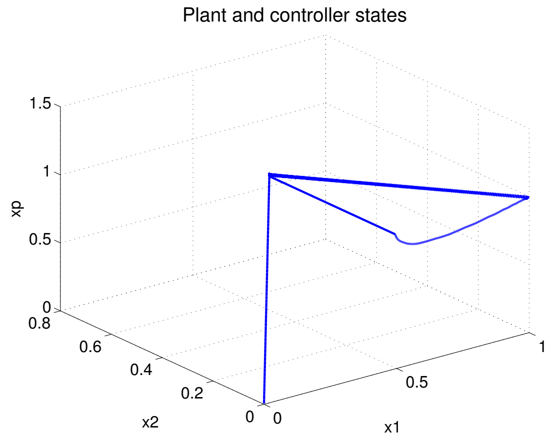


(b) Controllers output $u_1(t)$ and $u_2(t)$.

Figure 4.6: Plant and controller outputs for the Decentralized Reliable Balanced Control solution with two controllers and a zero-mean square-wave added to the system error. A fault in C_2 occurs at $t = 5s$. At $t = 15s$ C_2 is fixed and returned to the system with null initial conditions.

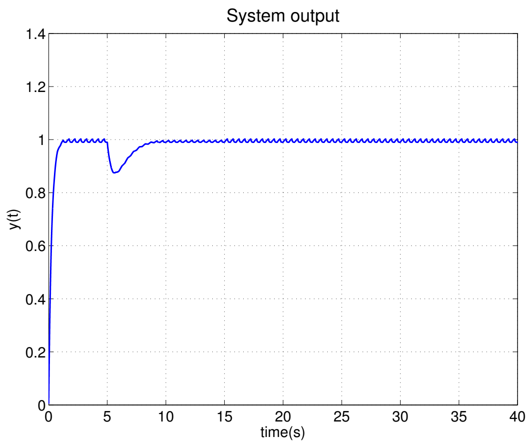


(a) Controllers states $x_1(t)$ and $x_2(t)$

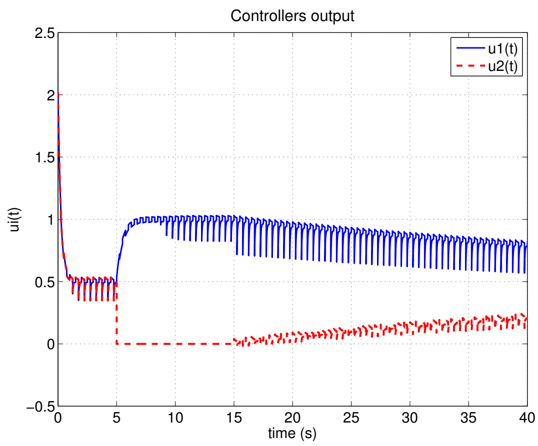


(b) System states $x_1(t)$, $x_2(t)$ and $x_p(t)$

Figure 4.7: System states for the Decentralized Reliable Balanced Control solution with two controllers and a zero-mean square-wave added to the system error. A fault in C_2 occurs at $t = 5s$. At $t = 15s$ C_2 is fixed and returned to the system with null initial conditions.

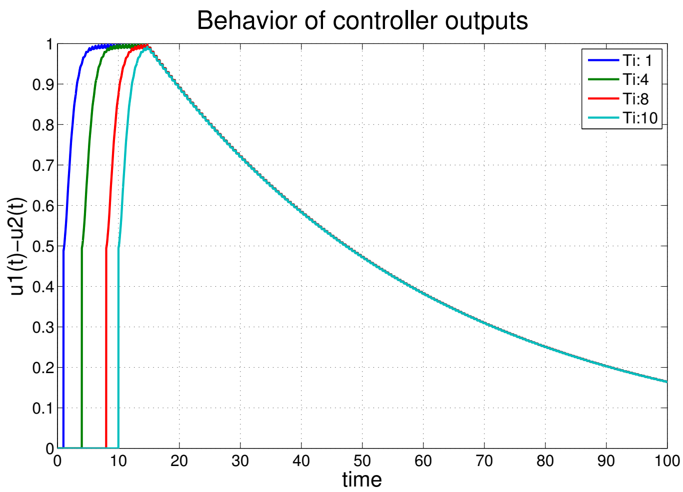


(a) System output $y_p(t)$

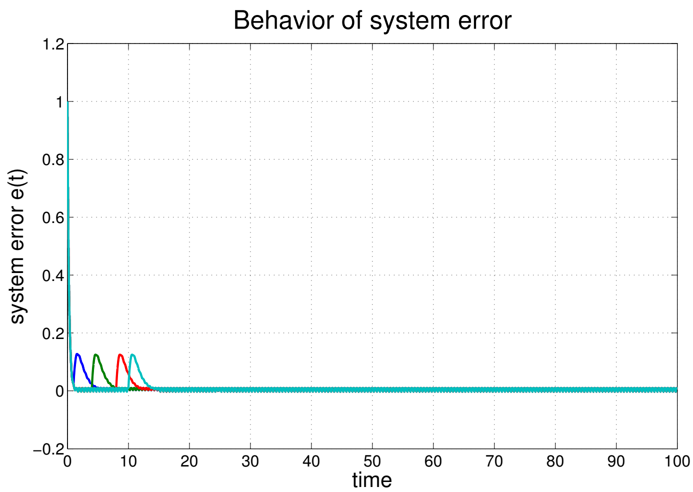


(b) Controllers output $u_1(t)$ and $u_2(t)$.

Figure 4.8: A zoom view of the simulation results for the Decentralized Reliable Balanced Control solution with two controllers and a zero-mean square-wave added to the system error. A fault in C_2 occurs at $t = 5s$. At $t = 15s$ C_2 is fixed and returned to the system with null initial conditions.

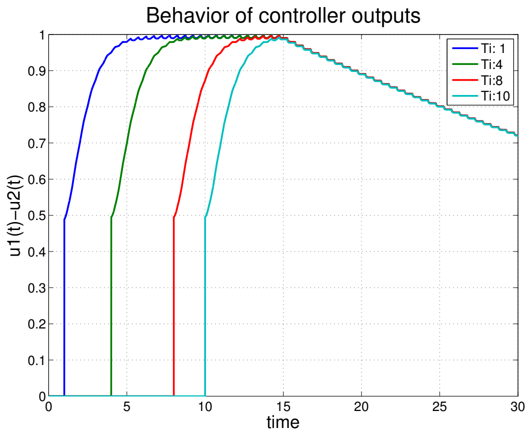


(a) Difference between controller outputs.

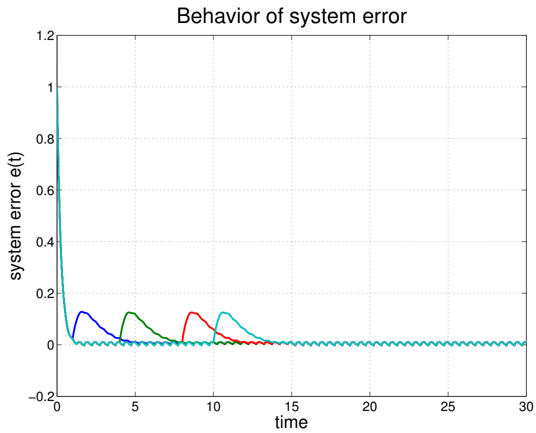


(b) System error versus time.

Figure 4.9: Simulation results with failures of C_2 occurring at different times. The results demonstrate the correct functioning of the RBC law.



(a) Difference between controller outputs.



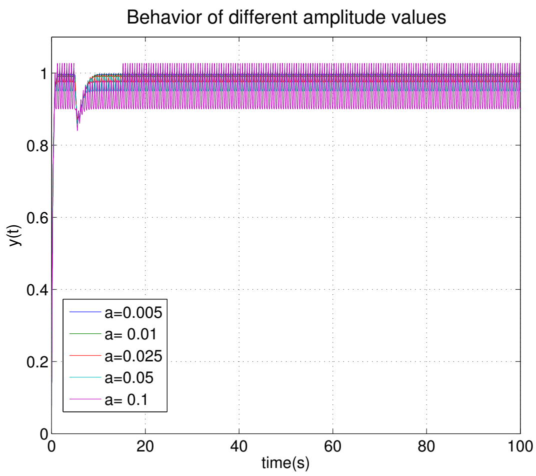
(b) System error versus time.

Figure 4.10: A zoom view of the simulation results with failures of C_2 occurring at different times. The results demonstrate the correct functioning of the RBC law.

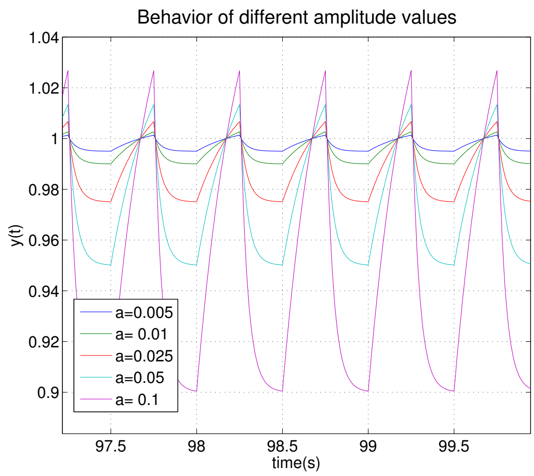
about the influence of the frequency of the square wave? Some graphs are presented in this section to illustrate the effect of variations of these parameters.

In Figure 4.11a the system output is plotted by considering different amplitude values for the square-wave. The ripple value for each amplitude values for the square wave is presented in Fig. 4.11c. In this study the ripple has been defined by the difference between the largest value of system output with the smallest one after the system has reached steady state, i. e. how much the signal will oscillate with respect to the reference.

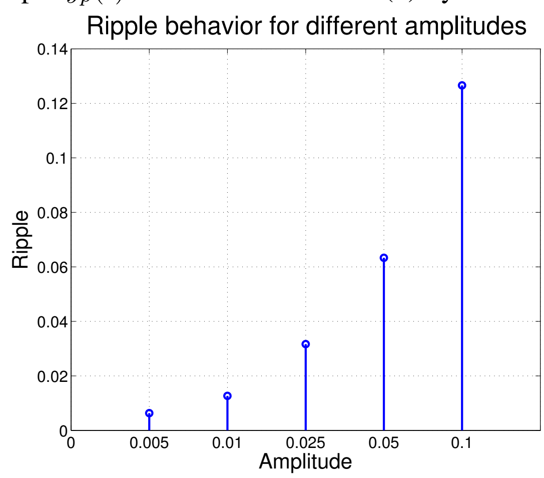
While studying the influence of frequency variation, the first observation is that the system time constant should never be slower than the period of the square wave. This means that the controller should reach steady-state at the end of every half cycle of the disturbance signal. In Figure 4.12, for frequency $f = 10$, the system does not have enough time to track the reference and so a steady state error appears in the output.



(a) System output $y_p(t)$.

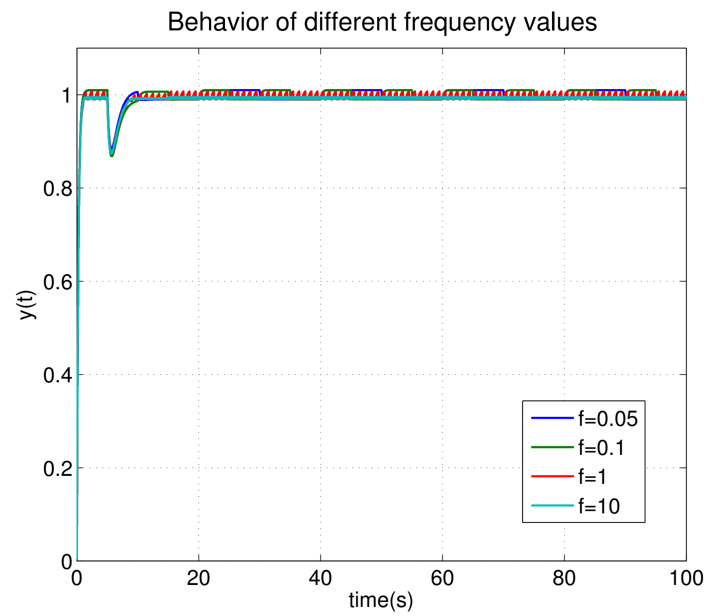


(b) System output with zoom applied.

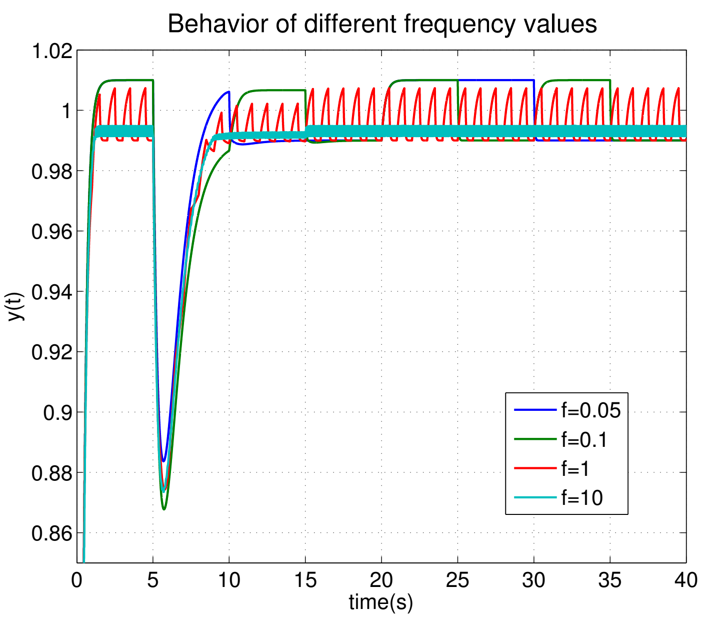


(c) Ripple presented by the system.

Figure 4.11: Behavior of the system for different values of amplitude of the persistence excitation.



(a) System output $y_p(t)$.



(b) Zoom at the first $30s$ of simulation.

Figure 4.12: Study of the influence of frequency variation on the system output.

Chapter 5

DRBC for first order plants with time delays

Delays are frequently found in feedback loops. It is expected that time delays will affect the system stability and so it is important to study this case to verify if the DRBC solution is applicable in the presence of a delay. In this chapter a study of a first order plant with time delay is presented. The idea is to add a time delay to the same first order plant from the previous chapter and verify its behaviour.

5.1 First order plant with time delay

A first order plant with time delay is expressed as follows:

$$P(s) = \frac{\beta}{s+\beta}e^{-\tau s} \tag{5.1}$$

where $\beta = 1$ and the time delay is represented by the exponential term, which is a transcendental function. To analyse this plant, the Padé-approximation is implemented, and the exponential term is now represented by a n^{th}-order polynomial as presented below.

$$e^{-\tau s} \approx \frac{1 - k_1 s + k_2 s^2 + ... \pm k_n s^n}{1 + k_1 s + k_2 s^2 + ... + k_n s^n}$$

The coefficients k_i are functions of n and the approximation precision increases with as n grows. Observe that the Padé approximation adds some right-half plane zeros to the system transfer function, and so that this system presents a non-minimum phase behaviour. To study the influence of time delay on the RBC, the first-order Padé approximation was used, i.e., $n = 1$ and by definition, $k_1 = \tau/2$:

$$P(s) = \left(\frac{\beta}{s+\beta}\right)\frac{1-(\tau/2)s}{(1+(\tau/2)s)} = \frac{\beta-(\beta\tau/2)s}{(\tau/2)s^2+(1+\beta\tau/2)s+\beta}$$

The new system is represented by (5.2).

$$\begin{aligned}
\dot{x}_1 &= s(e, x_1) \\
y_1 &= \alpha s(e, x_1) + sat(x_1) \\
\dot{x}_2 &= s(e, x_2) \\
y_2 &= \alpha s(e, x_2) + sat(x_2) \\
u &= y_1 + y_2 \\
\dot{x}_{p1} &= x_{p2} \\
\dot{x}_{p2} &= -\left(\tfrac{\beta}{\xi}\right) x_{p1} - \left(\tfrac{1+\beta\xi}{\xi}\right) x_{p2} + \left(\tfrac{\beta}{\xi}\right) u - \beta\dot{u} \\
y &= x_{p1} \\
e &= r - x_{p1}
\end{aligned} \tag{5.2}$$

The study about first order plants with time delays is performed by using Simulink and Matlab platform. The block diagram presented in Fig.5.1 represents the system composed by a first order plant with time delay with two controllers in parallel. The controllers are based on the DRBC law. Note that the first order Padé approximation was used to calculate the time delay. In Fig. 5.2 the controller C_1 is shown. The controller is resumed by a proportional gain α, a integrator with saturation and its gain k_I and the *Switch* bloc which, with the *Knonlinear* bloc, represents the switching logic K_{ci}. For those simulations, the same system parameters as in Chapter 4 (see Table 4.1) were used. As before, it is considered here that the controller C_2 fails at $t = 5s$ and at $t = 15s$ it is fixed and returned to the system with null initial conditions.

In Figure 5.3 a simulation was performed with $\tau = 0.1$. In this case, the system remains stable and, after a long time, it is able to achieve the consensus between the controller outputs. By increasing the parameter τ, as expected, the system becomes unstable. Figure 5.4 presents the results for $\tau = 0.3$.

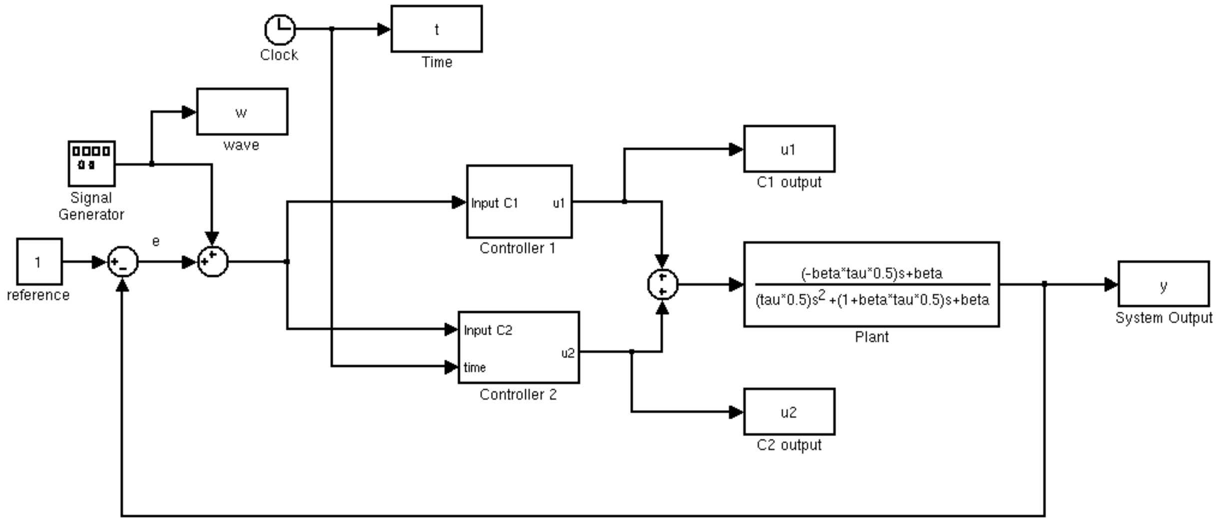


Figure 5.1: Blocks diagram (from Simulink) of the first order plant with time delay with DRBC.

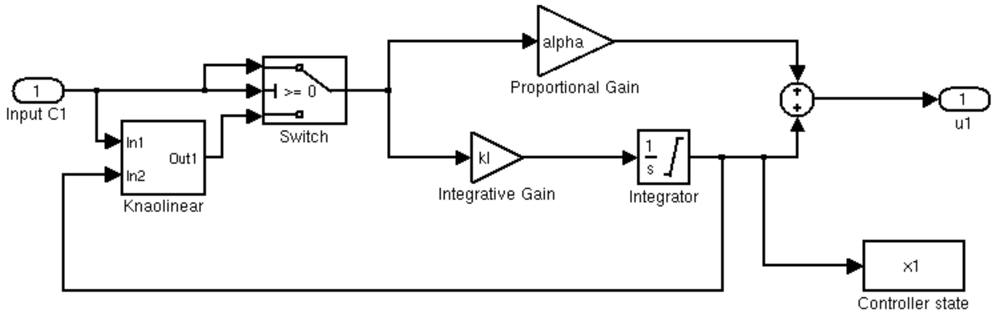
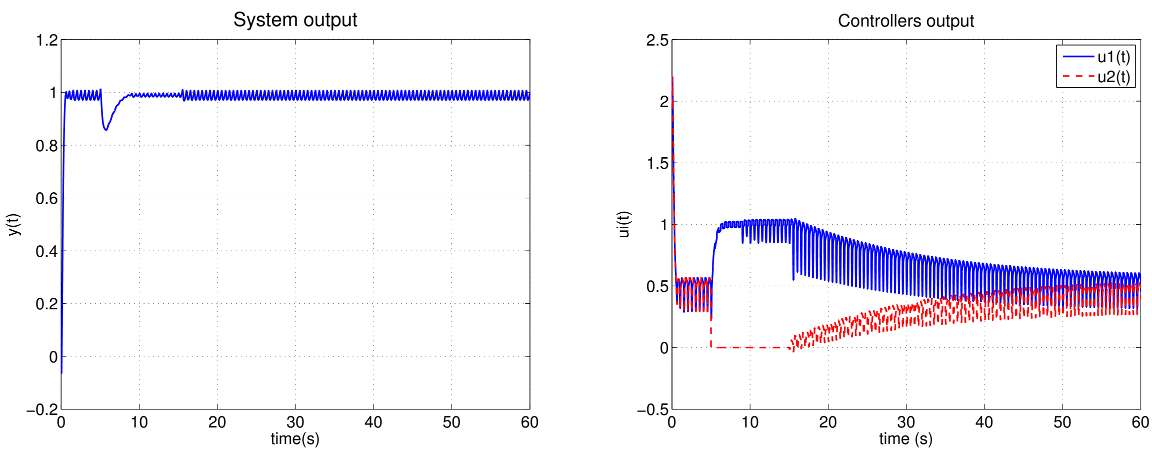


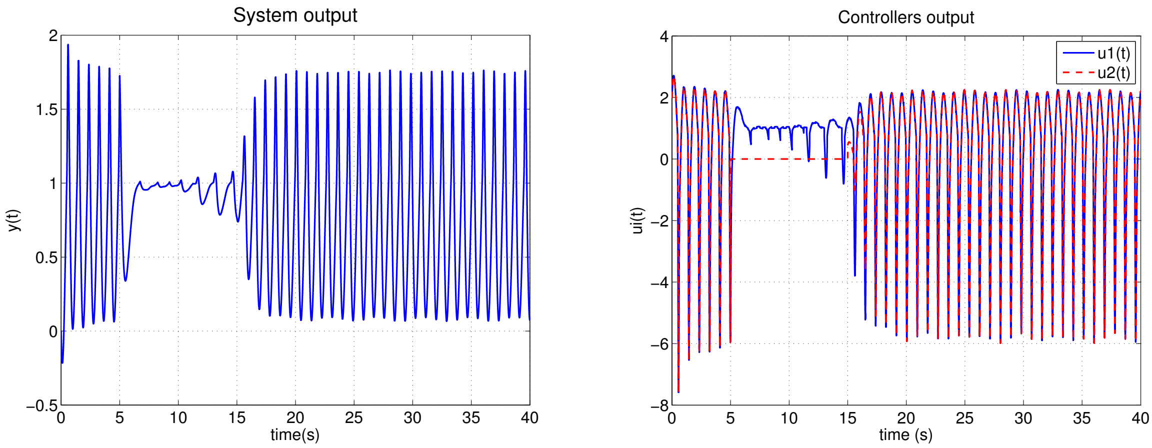
Figure 5.2: Representation of controller C_1 on Simulink.



(a) System output.

(b) Controllers output

Figure 5.3: Simulation results for a first order plant with time delay given by $\tau = 0.1$.



(a) System output. (b) Controllers output

Figure 5.4: Simulation results for a first order plant with time delay given by $\tau = 0.3$.

5.1.1 Smith Predictor Control

To control a system which incorporates time-delay, the compensation scheme known as *Smith Predictor Control* is frequently used. The method proposed by O.J.M.Smith in 1959 [2] is shown in the block diagram in Fig. 5.5. The Smith Predictor aims to minimize the undesired effects of the time delay. Since the time delay is approximated by a polynomial with poles in the RHP, system stability is affected and thus, the transient response for a system with time delay will be worse than for the system without the delay.

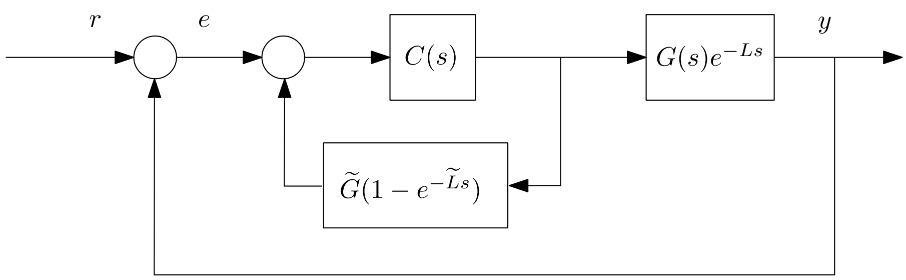


Figure 5.5: Smith Predictor Control scheme for a plant with time delay.

In the scheme presented, G represents a stable and strictly proper rational function and L a positive constant which stands for the time delay. $\widetilde{G}$ and $\widetilde{L}$ represents the nominal versions of G and L. The Smith Predictor, as can be seen in the Figure, adds a feedback loop around the controller to make the predicted output more precise, which is some times referred to as "delay cancellation".

5.1.2 Decentralized Reliable Balanced Control with Smith Predictor

In this section a new control structure is proposed for the time delay plant represented by (5.1). This new proposal combines the DRBC method with the Smith Predictor Control as shown in Fig. 5.6. The Simulink diagram for the new solution is represented in 5.7.

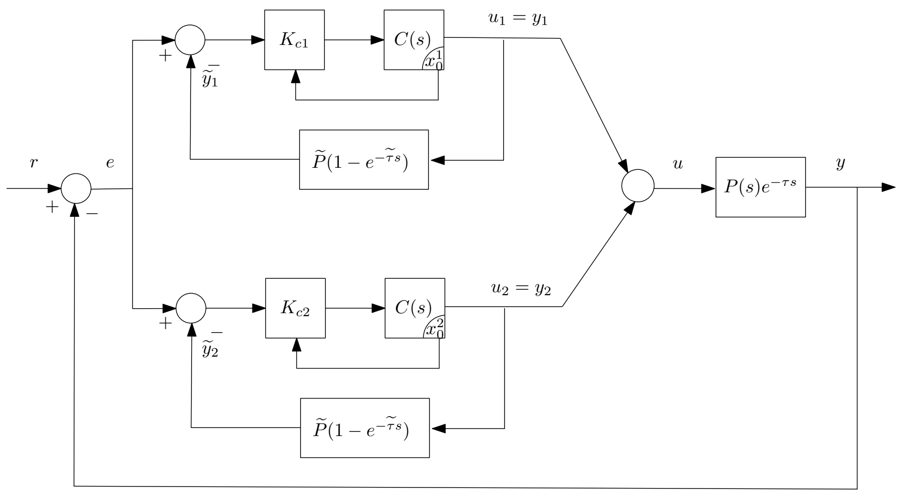


Figure 5.6: Smith Predictor added to DRBC method for first order plants with time delay.

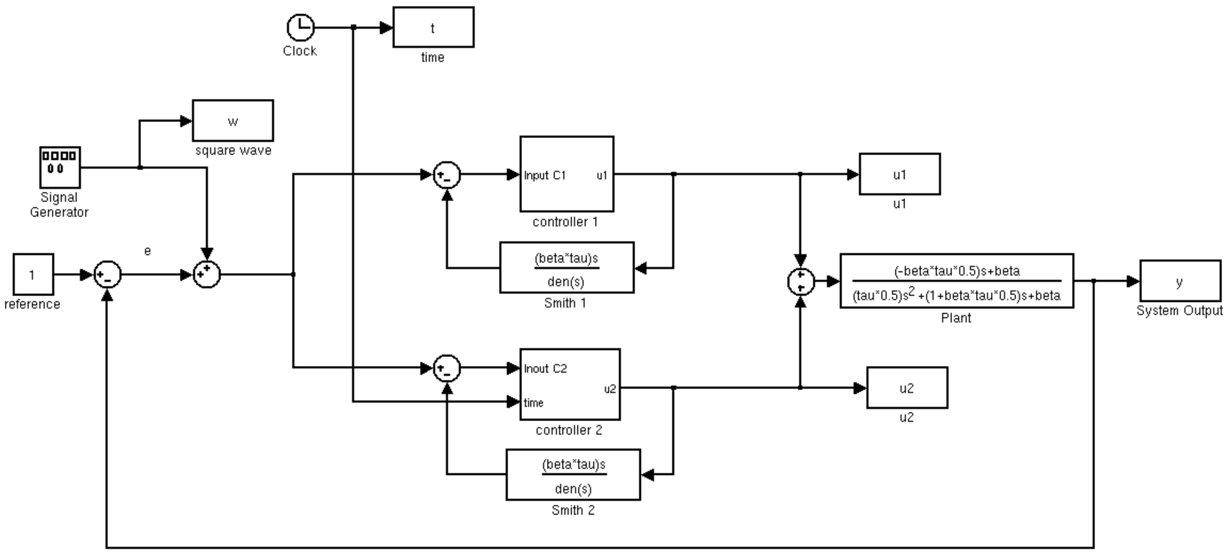
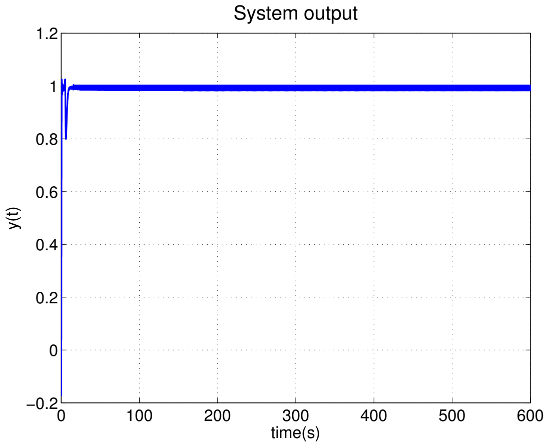
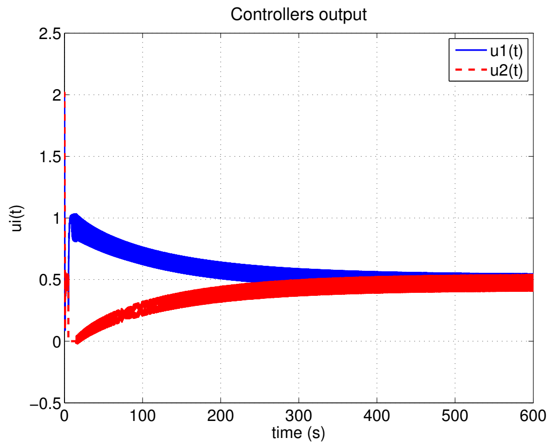


Figure 5.7: Blocks diagram (from Simulink) of the first order plant with time delay with DRBC and Smith Predictor.

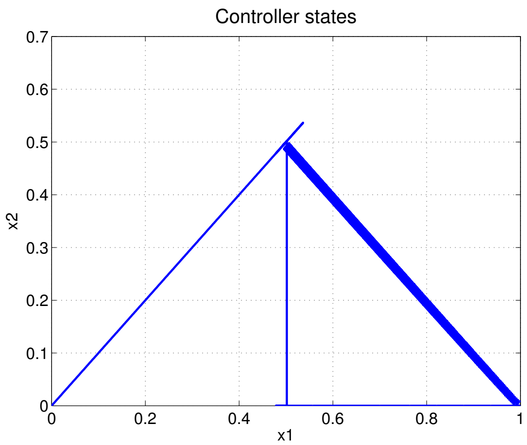
The simulation results presented in Fig.5.8 were generated by using the same control parameters as in Fig. 5.4. However, for the Smith predictor, the time delay has a estimation error of 0.05. The idea is to simulate a real scenario where the delay and/or the plant are unknown



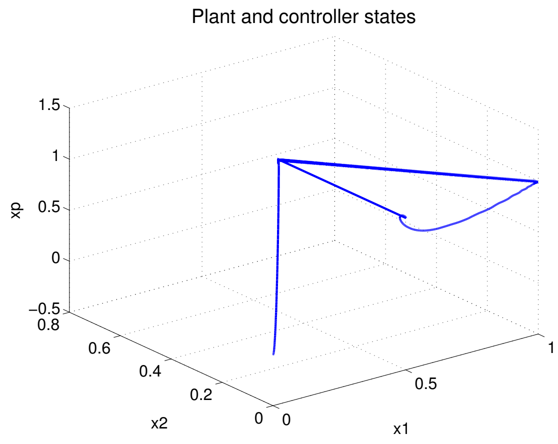
(a) System output.



(b) Controllers output



(c) Controller states



(d) Plant and controller states

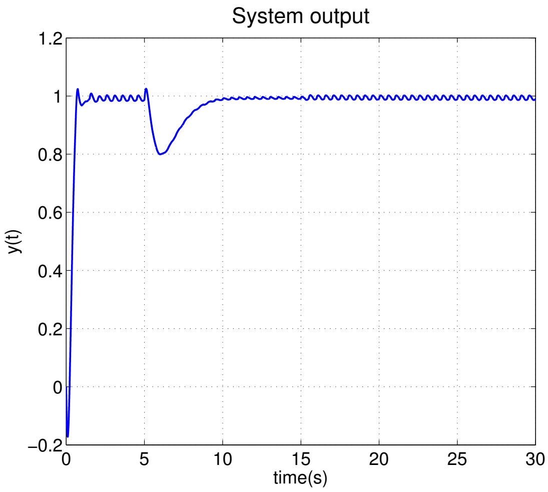
Figure 5.8: Simulation results for a first order plant with time delay given by $\tau = 0.3$ and estimation delay given by $\tau = 0.35$. The Smith Predictor added to the DRBC comprises the control law for this system.

and so, the Smith predictor is built with an estimate of the plant delay which is erroneous.

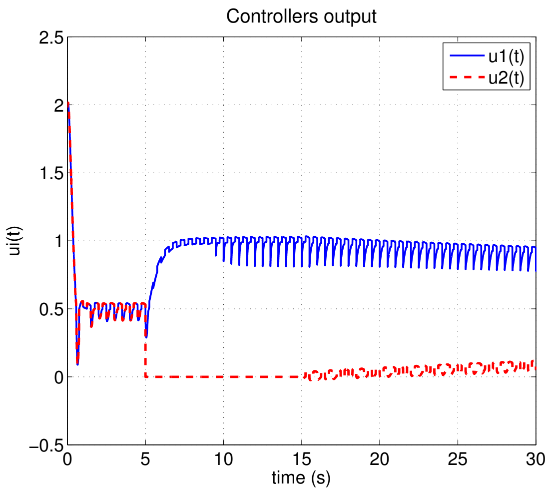
Figure 5.9 shows the system behavior for the first $30s$ of simulation. As can be seen, there is an impact on the system caused by the inexact estimation of τ however, the system maintains stable and it is capable of tracking the reference and guaranteeing the consensus between the controller outputs.

A new analysis by assuming errors in the estimation of the plant parameter β is presented in Fig.5.10. The estimated plant is given by:

$$P(s) = \frac{\widetilde{\beta}}{s + \widetilde{\beta}} = \frac{0.9}{s + 0.9}$$



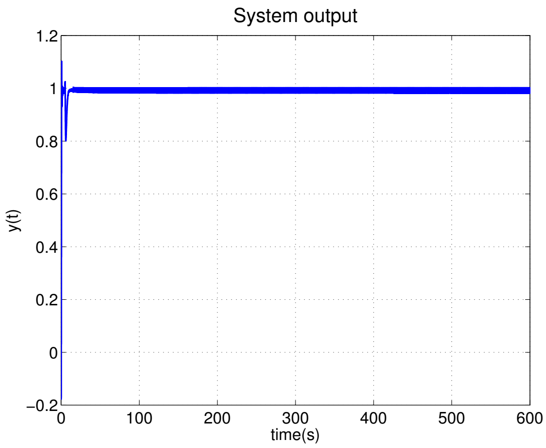
(a) System output.



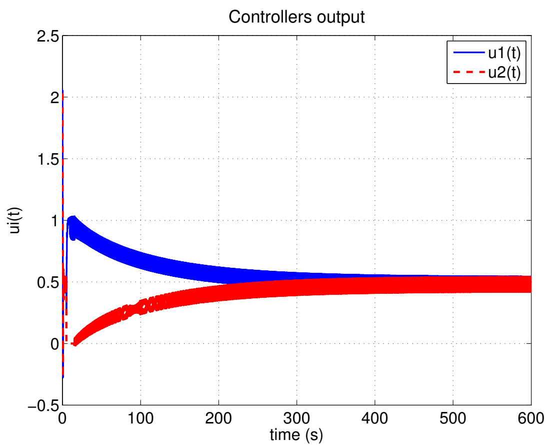
(b) Controllers output

Figure 5.9: Simulation results for a first order plant with time delay estimation given by $\tau = 0.35$ from $t = 0s$ to $t = 30s$. The Smith Predictor added to the DRBC comprises the control law for this system.

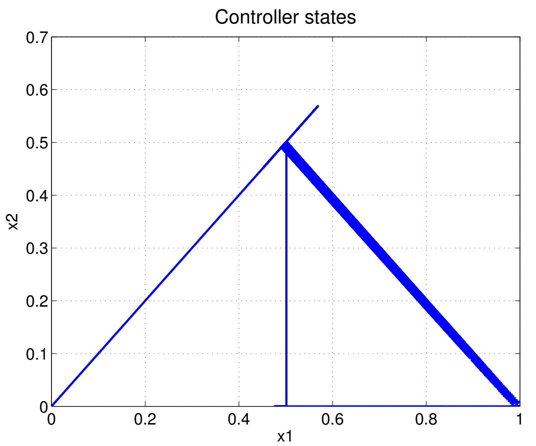
In Fig.5.11 a zoom is applied with the aim of showing the system behavior during the first $30s$. Even with the estimation error of β, the system is capable of guaranteeing the reference tracking and satisfying the consensus condition. Thus, the DRBC appears to be robust to small errors in the estimation of the delay. Systematic robustness studies of the Smith predictor have been carried out and could be applied to obtain precise bounds on the estimation error [3, 14, 13, 8, 15].



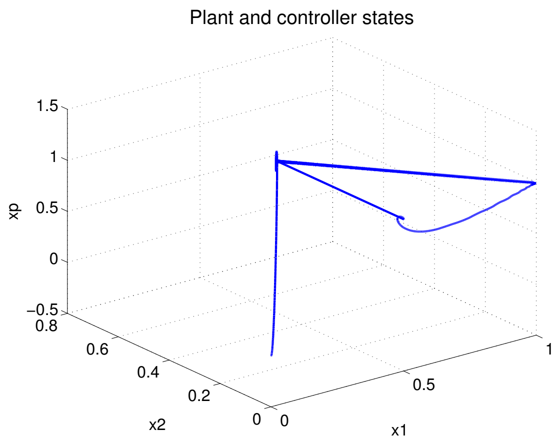
(a) System output.



(b) Controllers output

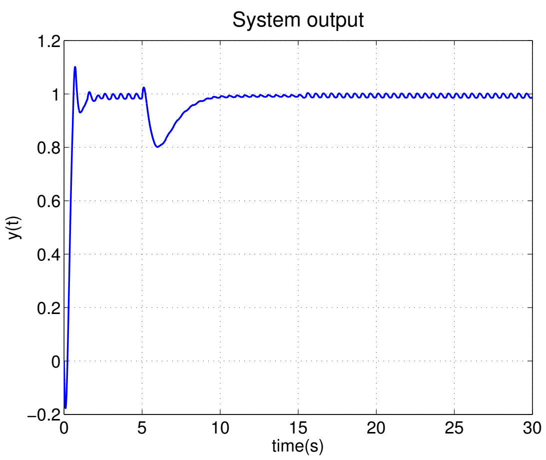


(c) Controller states

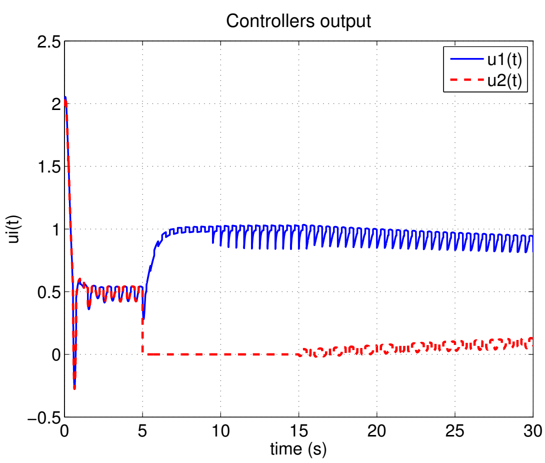


(d) Plant and controller states

Figure 5.10: Simulation results for a first order plant with time delay given by $\tau = 0.3$ and estimation plant parameter given by $\beta = 0.9$. The Smith Predictor added to the DRBC comprises the control law for this system.



(a) System output.



(b) Controllers output

Figure 5.11: Simulation results for a first order plant with time plant parameter estimation given by $\beta = 0.9$ from $t = 0s$ to $t = 30s$. The Smith Predictor added to the DRBC comprises the control law for this system.

Chapter 6

Conclusions

The purpose of the current study was to design a decentralized fault tolerant control for first order plants, based on a set of controllers placed in parallel, and for which it is desired that the control actions be equally divided between the functioning controllers. The class of faults is defined by the total failure of a controller. Faults occurrence can occur in multiple controllers simultaneously. A decentralized solution is required, meaning that each controller has no information about the status of the other controllers.

A linear solution is attempted first. In the presence of failures, this solution satisfies the regulation objectives but there is no consensus between the controller actions if different initial conditions are chosen. A nonlinearity was added to the first proposal and the results revealed that consensus between controller outputs would only be achieved if the control law depends on the actual state of the controller.

The Balanced Redundant Control is then presented in Chapter 5. In this solution, for negative errors the control action depends on the controller state and so the consensus of u_1 and u_2 becomes possible by modifying this state. Results revealed that an external signal is required to force the error to become negative from time to time. A zero-mean square wave with a small amplitude was then chosen to do this. The results show that the modified scheme would simultaneously satisfy all the problem objectives: when failures occur, the system is capable of tracking the reference and when a controller is added to the system, the consensus between the controllers output is attained.

To complement this study, time delays were considered and as expected, even for small delays the system could become unstable. To compensate the time delay, a Smith Predictor was added to each controller resulting from the DRBC method. The results suggest that the robustness and consensus are guaranteed.

This study has provided a solution for completely decentralized fault-tolerant control solution also achieving a consensus between the controller outputs. The method proposed is only applicable for first-order plants, including those with delays. The proposed control law minimizes the actuator efforts and is therefore expected to increase lifetime of the latter.

6.1 Future work

Future work should explore the validity of the *Balanced Redundant Control* solution for other classes of plants. Studies about the stability of this solution and adaptations to other plants may be interesting to generalize or to find further limitations.

The redundant balanced control problem formulated in this project is of interest for more general plants and future work should certainly consider the generalization of the ideas considered here. Within the scope of this work, the treatment of stability issues was incomplete and a complete stability proof would be very useful. Issues of practical implementation, for example in plants with nonnegative states are also expected to be topics of future research. Finally, the theoretical question of achieving consensus with minimal information exchange between agents is also richly deserving of further study.

Bibliography

[1] Control allocation survey. *Automatica*, 49(5):1087 – 1103, 2013.

[2] N. Abe and K. Yamanaka. Smith predictor control and internal model control - a tutorial. In *SICE 2003 Annual Conference*, volume 2, pages 1383 – 1387 Vol.2, 2003.

[3] A. Bhaya and C.A. Desoer. Controlling plants with delay. *International Journal of Control*, 41(3):813–830, 1985.

[4] Marc Bodson and Susan A. Frost. Control allocation with load balancing. *AIAA Guidance, Navigation, and Control Conference*, August 2009.

[5] N. Ertugrul, W. Soong, G. Dostal, and D. Saxon. Fault tolerant motor drive system with redundancy for critical applications. In *Power Electronics Specialists Conference, 2002. pesc 02. 2002 IEEE 33rd Annual*, volume 3, pages 1457 – 1462 vol.3, 2002.

[6] Jinyoung Lee et al. A constrained consensus problem using MPC. *International Journal of Control, Automation and Systems*, 9(5):952 – 957, 2011.

[7] B.A. Francis and W.M. Wonham. The internal model principle of control theory. *Automatica*, 12(5):457 – 465, 1976.

[8] C.E. Garcia and M. Morarl. Internal model control. 1. a unifying review and some new results. *Industrial and Engineering Chemistry Process Design and Development*, 21(2):308–323, 1982.

[9] A. N. Gundes and M. G. Kabuli. Reliable stabilization with integral action in decentralized control systems. *Automatica*, 32(7):1021 – 1025, 1996.

[10] Hongkeun Kim, Hyungbo Shim, and Jin Heon Seo. Output consensus of heterogeneous uncertain linear multi-agent systems. *IEEE Transactions on Automatic Control*, 56(1):200 – 206, 2011.

[11] MathPages. Failure rates, mtbfs, and all that, July 2013.

[12] MathPages. Infinite parallel redundancy, July 2013.

[13] J. E. Normey-Rico and E. F. Camacho. Dead-time compensators: A survey. *Control Engineering Practice*, 16(4):407–428, 2008.

[14] J.E. Normey-Rico and E.F. Camacho. Unified approach for robust dead-time compensator design. *Journal of Process Control*, 19(1):38–47, 2009.

[15] D.H. Owens and A. Raya. Robust stability of smith predictor controllers for time-delay systems. *IEE Proceedings D: Control Theory and Applications*, 129(6):298–304, 1982.

[16] Panagiotis Panagi and Marios M. Polycarpou. Decentralized fault tolerant control of a class of interconnected nonlinear systems. *IEEE Transactions On Automatic Control*, 56:178 – 184, 2011.

[17] Ron J Patton. Fault-tolerant control systems: The 1997 situation. *IFAC SAFEPROCESS'1997*, pages 1033 – 1054, 1997.

Chapter 7

Matlab Code
LTI Systems

```
function [M] = LTISSolution1 (T,Ti,Tf)

%---------------------
%   Initialization
%---------------------
t  = 0;
x1 = 0;
x2 = 0;
xp = 0;

dt = 0.001; %Euler step
N = T/dt;
M  = zeros(N,8);
%---------------------

%-----------------
%   Parameters
%-----------------
r = 1;
kI = 2;
alpha = 2;
beta = 1;
index = 0;
%-----------------

for i = 0:dt:T

    index = index + 1;
    e = r - xp;

    %----------------
    %Faillure in C2
    %-----------------
    if t≥Ti && t≤Tf
        f  = 0;
```

```
        x2 = 0;
    else
        f  = 1;
    end
    %----------------

    k1 = 1;
    k2 = 1;

    %-----------------------
    % Controllers output
    u1 = x1 + alpha*k1*e;
    u2 = x2 + alpha*k2*e*f;
    %-----------------------

    dx1 = kI*k1*e*dt;
    dx2 = kI*k2*e*dt*f;
    dxp = beta*(u1+u2-xp)*dt;

    x1 = x1+dx1;
    x2 = x2+dx2*f;
    xp = xp+dxp;

    t  = t+dt;

    %matrix M: t,x1,x2,xp,y,error
    M(index,:) = [t,x1,x2,xp,xp,e,u1,u2];

end
```

Chapter 8

Matlab Code
Switched Logic Solution

```
function [M] = SLSolution1 (T,Ti,Tf)

%--------------------
%   Initialization
%--------------------
t  = 0;
x1 = 0;
x2 = 0;
xp = 0;

dt = 0.001;
N = T/dt;
M  = zeros(N,8);
%--------------------

%----------------
%   Parameters
%----------------
kI = 2;
xmax = 1;
xmin = 0;
r = 1;
alpha = 2; %kp
beta = 1;
mu = 10;
index = 0;
%----------------

for i = 0:dt:T

    index = index + 1;
    e= r - xp;

    %--------------------
    % Faillure in C2
```

```
    if t≥Ti && t≤Tf
        f  = 0;
        x2 = 0;
    else
        f  = 1;
    end
    %--------------------

    if e> r/2
        k1 = 1;
        k2 = 1;
    else
        k1 = 1/2;
        k2 = 1/2;
    end

    %-----------------------
    % Controllers output
    u1 = x1 + alpha*k1*e;
    u2 = x2 + alpha*k2*e*f;
    %-----------------------

    dx1 = k1*e*dt*kI;
    dx2 = k2*e*dt*kI;
    dxp = beta*(u1+u2-xp)*dt;

    x1 = x1+dx1;
    x2 = x2+dx2*f;
    xp = xp+dxp;

    %----------------
    %Saturation
    %----------------
    if x1>xmax
        x1=xmax;
    end

    if x2>xmax
        x2=xmax;
    end

    if x1<xmin
        x1=xmin;
    end

    if x2<xmin
        x2=xmin;
    end
    %----------------

    t  = t + dt;

    %matrix M: t,x1,x2,xp,y,error
    M(index,:) = [t,x1,x2,xp,xp,e,u1,u2];
end
```

Chapter 9

Matlab Code
Decentralized Reliable Balanced Control

9.1 Decentralized Reliable Balanced Control without the square wave:

```
function [M] = BalancedSolutionNoSWpcode

%--------------------------
%   General Parameters
%--------------------------
T  = 25;    % Simulation max time
Ti = 5;
Tf = 15;

%--------------------
%   Initialization
%--------------------
t  = 0;
x1 = 0;
x2 = 0;
xp = 0;
dt = 0.001; %Euler step
N = T/dt;
M  = zeros(N,8);
%--------------------

%----------------
%   Parameters
%----------------
kI = 2;
xmax = 1;
xmin = 0;
r = 1;
alpha = 2;
beta = 1;
```

```
mu = 10;
%----------------

index = 0;

h=waitbar(0,'Waiting...');

for i = 0:dt:T

    if i==fix(i)
        waitbar(i/T);
    end

    index = index + 1;
    error = r - xp;
    e = error;

    %----------------
    %Faillure in C2
    %----------------
    if t≥Ti && t≤Tf
        f  = 0;
        x2 = 0;
    else
        f  = 1;
    end
    %----------------

    if e≥ 0
        k1 = 1;
        k2 = 1;
    else
        k1 = 1 + mu*x1/xmax;
        k2 = 1 + mu*x2/xmax;
    end

    %-----------------------
    % Controllers output
    u1 = x1 + alpha*k1*e;
    u2 = x2 + alpha*k2*e*f;
    %-----------------------

    dx1 = kI*k1*e*dt;
    dx2 = kI*k2*e*dt;
    dxp = beta*(u1+u2-xp)*dt;

    x1 = x1+dx1;
    x2 = x2+dx2*f;
    xp = xp+dxp;

    %----------------
    %Saturation
    %----------------
    if x1>xmax
        x1=xmax;
```

```
    end

    if x2>xmax
        x2=xmax;
    end

    if x1<xmin
        x1=xmin;
    end

    if x2<xmin
        x2=xmin;
    end
    %----------------

    t   = t+dt;

    %matrix M: t,x1,x2,xp,y,error,u1,u2
    M(index,:) = [t,x1,x2,xp,xp,error,u1,u2];

end
```

9.2 Decentralized Reliable Balanced Control with the square wave:

```
function [M] = BalancedSolutionpcode

%--------------------------
%   General Parameters
%--------------------------
T   = 100;    % Simulation max time
Ti = 5;
Tf = 15;
% Square wave
Amplitude = 0.01;
freq = 2;

%--------------------
%   Initialization
%--------------------
t   = 0;
x1 = 0;
x2 = 0;
xp = 0;
dt = 0.001; %Euler step
N = T/dt;
M   = zeros(N,8);
%--------------------

%----------------
%   Parameters
```

```
%----------------
kI = 2;
xmax = 1;
xmin = 0;
r = 1;
alpha = 2;
beta = 1;
mu = 10;
%----------------

%--------------------
%   Square Wave
%--------------------
time = 0:dt:T; % Sampling frequency
d = Amplitude*square(2*pi*freq*time);
%--------------------

index = 0;

h=waitbar(0,'Waiting...');

for i = 0:dt:T

    if i==fix(i)
        waitbar(i/T);
    end

    index = index + 1;
    error = r - xp;
    e = error + d(index);

    %----------------
    %Faillure in C2
    %----------------
    if t≥Ti && t≤Tf
        f  = 0;
        x2 = 0;
    else
        f  = 1;
    end
    %----------------

    if e≥ 0
        k1 = 1;
        k2 = 1;
    else
        k1 = 1 + mu*x1/xmax;
        k2 = 1 + mu*x2/xmax;
    end

    %-----------------------
    % Controllers output
    u1 = x1 + alpha*k1*e;
    u2 = x2 + alpha*k2*e*f;
    %-----------------------
```

```

    dx1 = kI*k1*e*dt;
    dx2 = kI*k2*e*dt;
    dxp = beta*(u1+u2-xp)*dt;

    x1 = x1+dx1;
    x2 = x2+dx2*f;
    xp = xp+dxp;

    %-----------------
    %Saturation
    %-----------------
    if x1>xmax
        x1=xmax;
    end

    if x2>xmax
        x2=xmax;
    end

    if x1<xmin
        x1=xmin;
    end

    if x2<xmin
        x2=xmin;
    end
    %-----------------

    t  = t+dt;

    %matrix M: t,x1,x2,xp,y,error,u1,u2
    M(index,:) = [t,x1,x2,xp,xp,error,u1,u2];

end
```