

UNIVERSIDAD DE ALMERIA

ESCUELA SUPERIOR DE INGENIERÍA

MULTIVARIABLE CONTROLLER FOR
STATIONARY FLAT
PLATE SOLAR COLLECTORS

Curso 2018/2019

Alumno/a:

Andrea Tosi

Director/es:

Dr. D. Antonio Visioli
Dr. D. Manuel Berenguel





UNIVERSIDAD DE ALMERÍA
Escuela Superior de Ingeniería

Trabajo Fin de Grado
Ingeniería Electrónica Industrial



**UNIVERSITÀ
DEGLI STUDI
DI BRESCIA**

**UNIVERSITÀ DEGLI STUDI DI
BRESCIA**

Dipartimento di Ingegneria
Meccanica Industriale
Corso di Laurea Magistrale in Ingegneria
dell' Automazione Industriale

Multivariable controller for stationary flat plate solar collectors

Controlador multivariable para colectores solares
estacionarios de placa plana.

Doble Título UNIBS-UAL Mechatronics for Industrial Automation

Autor: Andrea Tosi

Director: Dr. D. Antonio Visioli
Codirector: Dr D. Manuel Berenguel

**Almería (España), Diciembre 2018
Curso 2018-2019**

Giunto alla fine del mio percorso di studi mi risulta difficile trovare le parole adatte per ringraziare tutte le persone che mi sono state affianco in questi anni di studio. Innanzitutto desidero ringraziare il professor Antonio Visioli il quale, in questi anni, oltre ad essere stato un eccellente professore, ha saputo trasmettermi la passione per il proprio lavoro ed ha, senza alcun dubbio, dato una svolta positiva alla mia vita grazie ai progetti Erasmus da lui sostenuti. Credo che poche persone possano ritenersi sinceramente fortunate per aver incontrato un buon professore. Io, in tutta onestà, non so se il fato in questi casi riveste un ruolo decisivo ma posso affermare con decisione che la professionalità, il carisma, la simpatia e le vaste conoscenze del professor Antonio Visioli sono state decisive per la mia formazione universitaria come ingegnere e personale come uomo. Occorre ricordare che questa Tesi di Laurea Magistrale è stata svolta all'estero e ciò non sarebbe stato possibile senza il supporto del professor Manuel Berenguel e della ricercatrice Lidia Rocha. In particolar modo desidero ringraziare la ricercatrice Lidia Rocha per le ore che mi ha dedicato, sia presso la Plataforma Solar De Almería, sia presso il centro di ricerca CIESOL. Dopo aver citato il personale dell'ateneo, non posso far altro che rivolgere i miei ringraziamenti più sentiti e sinceri ai miei genitori. Nonostante le normali discussioni tra genitori e figli, sono state le persone che più di ogni altro hanno creduto in me. La fiducia incondizionata che hanno sempre riposto nei miei confronti è stata determinante ed indispensabile per poter portare a termine nel migliore dei modi questo percorso di studi. Non smetterò mai di essere loro riconoscente per l'amore che mi hanno donato ed i sacrifici che hanno fatto per la mia formazione. Consapevole del mio temperamento ringrazio anche tutti i miei amici e la mia fidanzata per avermi saputo sopportare durante i periodi di stress e per essermi stati sempre vicino.

Grazie a tutti...

Abstract

La Plataforma Solar de Almeria (PSA) è il più grande laboratorio di ricerca sull'energia solare concentrata d'Europa. È situata ad Almería, una provincia spagnola all'interno della quale si trova il deserto di Tabernas. Questa Tesi di Laurea Magistrale nasce dalla volontà di sviluppare un sistema di controllo per il campo solare termico AQUASOL-II (PSA). Quest'installazione fornisce l'acqua in ingresso ad un impianto di desalinizzazione MED (Multiple Effect Distillation), il cui rendimento è legato alla temperatura dell'acqua in ingresso allo stesso.

Tale installazione è stata realizzata nel 2006 con l'intenzione di incrementare le ricerche sull'utilizzo dell'energia solare termica negli impianti di desalinizzazione. Nello specifico è stato integrato un campo di pannelli solari termici da un impianto MED in modo da sfruttare l'energia solare per la produzione dell'acqua calda necessaria al processo di desalinizzazione. L'obiettivo principale era quello di creare un sistema di controllo in grado di mantenere l'acqua, in uscita dai vari pannelli solari del sistema, ad una temperatura desiderata. Come variabile controllata per tale scopo è stato scelto il flusso. L'energia solare prelevata all'interno dei vari loop di pannelli solari è legata al flusso d'acqua e di conseguenza è possibile affermare che la temperatura, in uscita da ogni singolo loop, dipende dal flusso che passa all'interno del loop stesso.

Si andrà ora nel dettaglio ad esporre i vari passaggi che sono stati svolti.

Modellizzazione: Come primo passo si è cercato di modellizzare il sistema flusso-pompe. La campagna sperimentale svolta ha visto una serie di scalini di diverso valore applicati alle pompe. Questo ha portato alla luce alcune non linearità nel sistema, nonché la presenza

di zeri positivi nelle funzioni di trasferimento (sistema non lineare e non a fase minima). Per risolvere i problemi dati dalle non linearità sono stati sviluppati due modelli. Nel primo caso, sfruttando il principio del *gain scheduling*, si è cercato di sviluppare un sistema che facesse intervenire nel modello, a seconda della situazione, le funzioni di trasferimento maggiormente adeguate. Nel secondo caso è stato diviso il modello in due parti, una composta da funzioni di trasferimento in grado di descriverne la dinamica e l'altra composta da funzioni polinomiali che ne definiscono il guadagno. Le non linearità del sistema in esame infatti erano contenute nel guadagno statico, mentre la dinamica era modellizzabile, con discreti risultati, tramite funzioni di trasferimento.

La bontà dei modelli è stata infine verificata applicandovi gli stessi ingressi applicati al campo reale e confrontando i risultati.

Una volta sviluppati dei modelli in grado di descrivere le relazioni tra il flusso nei vari loop e le pompe presenti nel circuito idraulico, è stato necessario fornirsi di un modello in grado di mettere in relazione le variabili di flusso e di temperatura. In questo caso è stato utilizzato, con alcune modifiche, un modello già presente nella letteratura. Tale modello è basato su equazioni differenziali ed è in grado di considerare i contributi sull'uscita dati dai disturbi sul carico. È stato infatti possibile effettuare simulazioni basate su valori reali di radiazione, temperatura ambiente e temperatura dell'acqua in ingresso al sistema.

Controllo: Dato che l'obiettivo finale era il controllo di temperatura e che il sistema era divisibile in due parti (pompe-flusso flusso-temperatura), ci si è concentrati sullo sviluppo di un controllo in cascata multivariabile composto da PID.

In una prima fase sono stati sviluppati due controllori di flusso per i due modelli precedentemente esposti. Entrambi i controllori sviluppati dovevano essere in grado di disaccoppiare i vari loop (MIMO controller) tenendo in considerazione che il sistema in esame era non a fase minima e non lineare. Una volta implementati, i controllori sono stati testati sia in simulazione che sul campo, ed è stato scelto il controllore basato sul secondo modello per formare l'anello interno del controllore in cascata.

L'anello esterno, dato dalla relazione temperatura-flusso, doveva essere in grado di fornire

Abstract

il corretto set-point all'anello interno. In questo caso il problema maggiore è stato dato dai forti disturbi sul carico presenti nel sistema. Occorre pensare infatti che la grandezza di maggior interesse in questo caso è stata la radiazione solare, la quale, subisce forti e repentine variazioni in giornate nuvolose. Si è ritenuto doveroso considerare anche altre grandezze come la temperatura esterna e la temperatura in ingresso dell'acqua nel circuito. Per far fronte a tutti questi eventi esterni, in grado di rendere inefficace l'azione di controllo dei PID è stata inserita nel controllore l'azione di un feed forward.

Il controllore in cascata così sviluppato è stato infine testato in simulazione con risultati soddisfacenti.

Contents

1	Introduction	1
1.1	Multiple-Effect Distillation (MED)	3
1.1.1	Functioning	3
1.1.2	Configuration	4
1.2	Solar collectors	6
1.3	Plataforma Solar de Almeria	8
1.3.1	AQUASOL-II	8
2	Models	11
2.1	Introduction	11
2.2	Flow model 1 - gain scheduling	14
2.2.1	Introduction	14
2.2.2	Transfer function - under 50%	18
2.2.3	Transfer function - upper 50%	23
2.2.4	Indirect transfer function	28
2.2.5	Model scheme	37
2.2.6	RGA matrix	42
2.2.7	Validation	43
2.3	Flow model 2 - non-linear gain	46
2.3.1	Introduction	46
2.3.2	Transfer function	47
2.3.3	Model scheme	51
2.3.4	RGA matrix	54
2.3.5	Validation	54
2.4	Temperature model	57

CONTENTS

3	Flow controller	61
3.1	Introduction	61
3.2	Gain scheduling controller	61
3.2.1	Controller scheme	62
3.3	Non-linear controller	69
3.3.1	Controller scheme	69
4	Temperature controller	73
4.1	Flow controller - inner loop	74
4.2	Temperature controller - external loop	77
5	Control - simulations and experiments	83
5.1	Introduction	83
5.2	Flow control - gain scheduling control	84
5.2.1	Test 1 - tuning Kappa Tau $M_s = 2.0$	84
5.2.2	Test 2 - tuning Kappa Tau $M_s = 1.4$	87
5.2.3	Test 3 - C.H.R. tuning set-point following	90
5.2.4	Test 4 - C.H.R. tuning load disturbance rejection	92
5.3	Flow control - non-linear control	95
5.3.1	Test 1 - tuning Kappa Tau $M_s = 2.0$	95
5.3.2	Test 2 - tuning Kappa Tau $M_s = 1.4$	98
5.3.3	Test 3 - C.H.R. tuning set-point following	100
5.3.4	Test 4 - C.H.R. tuning load disturbance rejection	103
5.4	Temperature control	105
5.4.1	Load disturbances	106
5.4.2	Test 1	109
5.4.3	Test 2	113
	Conclusions and future works	118
A	LabVIEW interface	119
A.1	Front Panel	119
A.2	Block diagram	121

List of Figures

1.1	Scheme of MED plant, water cycle in the different effects	4
1.2	Forward feed MED plant	5
1.3	Parallel feed MED plant	5
1.4	Flat plate solar collectors scheme, main part connections	7
1.5	Flat plate solar collectors photo	7
1.6	AQUASOL-II solar field facility at PSA (Spain) - Block diagram	9
1.7	AQUASOL-II plant , stationary flat plate solar collectors configuration . .	9
1.8	AQUASOL-II solar field facility at PSA (Spain) - Photo	10
2.1	AQUASOL-II solar field data for the model identification, 15-03-2018 . . .	13
2.2	Gain scheduling model, basic scheme	14
2.3	MIMO scheme for each part of the gain scheduling model	17
2.4	Transfer function with $U_0 < 50\%$: G00 - step response , pole-zero maps . .	20
2.5	Transfer function with $U_0 < 50\%$: G10 - step response , pole-zero maps . .	20
2.6	Transfer function with $U_0 < 50\%$: G30 - step response , pole-zero maps . .	20
2.7	Transfer function with $U_0 < 50\%$: G40 - step response , pole-zero maps . .	21
2.8	Transfer function with $U_0 < 50\%$: G50 - step response , pole-zero maps . .	21
2.9	Transfer function with $U_0 < 50\%$: G11 - step response , pole-zero maps . .	21
2.10	Transfer function with $U_0 < 50\%$: G33 - step response , pole-zero maps . .	22
2.11	Transfer function with $U_0 < 50\%$: G44 - step response , pole-zero maps . .	22
2.12	Transfer function with $U_0 < 50\%$: G55 - step response , pole-zero maps . .	22
2.13	Transfer function with $U_0 \geq 50\%$: G00 - step response , pole-zero maps . .	25
2.14	Transfer function with $U_0 \geq 50\%$: G10 - step response , pole-zero maps . .	25
2.15	Transfer function with $U_0 \geq 50\%$: G30 - step response , pole-zero maps . .	25
2.16	Transfer function with $U_0 \geq 50\%$: G40 - step response , pole-zero maps . .	26
2.17	Transfer function with $U_0 \geq 50\%$: G50 - step response , pole-zero maps . .	26

LIST OF FIGURES

2.18	Transfer function with $U_0 \geq 50\%$: G11 - step response , pole-zero maps . .	26
2.19	Transfer function with $U_0 \geq 50\%$: G33 - step response , pole-zero maps . .	27
2.20	Transfer function with $U_0 \geq 50\%$: G44 - step response , pole-zero maps . .	27
2.21	Transfer function with $U_0 \geq 50\%$: G55 - step response , pole-zero maps . .	27
2.22	Indirect transfer function: G01 - step response , pole-zero maps	31
2.23	Indirect transfer function: G31 - step response , pole-zero maps	31
2.24	Indirect transfer function: G41 - step response , pole-zero maps	32
2.25	Indirect transfer function: G51 - step response , pole-zero maps	32
2.26	Indirect transfer function: G03 - step response , pole-zero maps	32
2.27	Indirect transfer function: G13 - step response , pole-zero maps	33
2.28	Indirect transfer function: G43 - step response , pole-zero maps	33
2.29	Indirect transfer function: G53 - step response , pole-zero maps	33
2.30	Indirect transfer function: G04 - step response , pole-zero maps	34
2.31	Indirect transfer function: G14 - step response , pole-zero maps	34
2.32	Indirect transfer function: G34 - step response , pole-zero maps	34
2.33	Indirect transfer function: G54 - step response , pole-zero maps	35
2.34	Indirect transfer function: G05 - step response , pole-zero maps	35
2.35	Indirect transfer function: G15 - step response , pole-zero maps	35
2.36	Indirect transfer function: G35 - step response , pole-zero maps	36
2.37	Indirect transfer function: G45 - step response , pole-zero maps	36
2.38	Gain scheduling model, simulink scheme	37
2.39	Input and output initial condition strategy	39
2.40	General principal pump scheme, inner part of the scheme in Fig.2.38	40
2.41	tf00 block, inner part of the scheme in Fig.2.40	40
2.42	General loops pump scheme, inner part of the scheme in Fig.2.38	41
2.43	tf11 block,, inner part of the scheme in Fig.2.42	41
2.44	tf10 block, inner part of the scheme in Fig.2.42	41
2.45	Gain scheduling approach - Model validation, comparison between real and simulated data	44
2.46	Gain scheduling approach - Model error	45
2.47	Non-linear model, principal pump blocks	52
2.48	Non-linear model, transfer function 00	52
2.49	Non-linear model, loops pump blocks	53
2.50	Non-linear model, transfer function 01	53

LIST OF FIGURES

2.51	Non-linear model - validation, comparison between real and simulated data	55
2.52	Non-linear model - error	56
2.53	Temperature model scheme, controlled variable and load disturbance . . .	59
3.1	Gain scheduling controller - Simulink scheme	63
3.2	Controller Blocks, relative to Fig. 3.1	64
3.3	Bumpless blocks, relative to Fig. 3.1	66
3.4	Switch controller system, relative to Fig. 3.1	67
3.5	Non-linear controller - Simulink scheme	69
3.6	Non-linear controller - controller blocks relative to Fig. 3.5	70
4.1	Cascade control scheme for temperature control	73
4.2	Controller signals, inner loops	75
4.3	Anti-windup structure with variable limits	76
4.4	Temperature controller scheme: PIDs action, FF action and anti windup .	78
5.1	Gain scheduling control - Tuning Kappa Tau $M_s = 2.0\%$ - Flow, comparison between simulated and real signal	85
5.2	Gain scheduling control - Tuning Kappa Tau $M_s = 2.0\%$ - Control signal, comparison between simulated and real signal	86
5.3	Gain scheduling control - Tuning Kappa Tau $M_s = 1.4\%$ - Flow, comparison between simulated and real signal	88
5.4	Gain scheduling control - Tuning Kappa Tau $M_s = 1.4\%$ - Control signal, comparison between simulated and real signal	89
5.5	Gain scheduling control - Tuning CHR Set-Point following - Flow, compar- ison between simulated and real signal	91
5.6	Gain scheduling control - Tuning CHR Set-Point following - Control signal, comparison between simulated and real signal	92
5.7	Gain scheduling control - Tuning CHR load disturbance rejection - Flow, comparison between simulated and real signal	93
5.8	Gain scheduling control - Tuning CHR load disturbance rejection - Control signal, comparison between simulated and real signal	94
5.9	Non-linear control - Tuning Kappa Tau $M_s = 2.0\%$ - Flow, comparison between simulated and real signal	96

LIST OF FIGURES

5.10	Non-linear control - Tuning Kappa Tau $M_s = 2.0\%$ - Control signal, comparison between simulated and real signal	97
5.11	Non-linear control - Tuning Kappa Tau $M_s = 1.4\%$ - Flow, comparison between simulated and real signal	99
5.12	Non-linear control - Tuning Kappa Tau $M_s = 1.4\%$ - Control signal, comparison between simulated and real signal	100
5.13	Non-linear control - Tuning CHR Set-Point following - Flow, comparison between simulated and real signal	101
5.14	Non-linear control - Tuning CHR Set-Point following - Control signal, comparison between simulated and real signal	102
5.15	Non-linear control - Tuning CHR load disturbance rejection - Flow, comparison between simulated and real signal	104
5.16	Non-linear control - Tuning CHR load disturbance rejection - Control signal, comparison between simulated and real signal	105
5.17	Irradiation data for the simulation	107
5.18	Ambient temperature data for the simulation	107
5.19	Inlet temperature loop 1 data for the simulation	108
5.20	Inlet temperature loop 3 data for the simulation	108
5.21	Inlet temperature loop 4 data for the simulation	109
5.22	Inlet temperature loop 5 data for the simulation	109
5.23	Simulation 1 - temperature control loop 1	111
5.24	Simulation 1 - temperature control loop 3	111
5.25	Simulation 1 - temperature control loop 4	112
5.26	Simulation 1 - temperature control loop 5	112
5.27	Simulation 2 - temperature control loop 1	113
5.28	Simulation 2 - temperature control loop 3	114
5.29	Simulation 2 - temperature control loop 4	114
5.30	Simulation 2 - temperature control loop 5	115
A.1	LabVIEW - Front Panel	120
A.2	LabVIEW - Block diagram - pt.3.1	122
A.3	LabVIEW - Block diagram - pt.3.2	123

List of Tables

5.1	Tuning gain scheduling controller - Kappa Tau Methods $M_s = 2.0$, PIDs parameters value	85
5.2	Tuning gain scheduling controller - Kappa Tau Methods $M_s = 1.4$, PIDs parameters value	87
5.3	Tuning gain scheduling controller - CHR tuning Set-Point following, PIDs parameters value	90
5.4	Tuning gain scheduling controller - CHR tuning load disturbance rejection, PIDs parameters value	93
5.5	Tuning non-linear controller - Kappa Tau Methods $M_s = 2.0$, PIDs parameters value	95
5.6	Tuning non-linear controller - Kappa Tau Methods $M_s = 1.4$, PIDs parameters value	98
5.7	Tuning non-linear controller - CHR tuning Set-Point following, PIDs parameters value	101
5.8	Tuning non-linear controller - CHR tuning load disturbance rejection, PIDs parameters value	103
5.10	Cascade control tuning	106

List of Acronyms

CHR Chien Hrones Reswich.

FF Feed Forward.

IC Initial Condition.

MED Multiple Effect Distillation.

MIMO Multiple Input Multiple Output.

MISO Multiple Input Single Output.

PSA Plataforma Solar de Almeria.

SAT Saturation.

SISO Single Input Single Output.

SP Set Point.

SS Steady State.

Ta Settling time at 5%.

Chapter 1

Introduction

In the last few years the human activity has significantly increased, and therefore the request of fresh water has increased as well. Particular relevance is the increase of water used by a single person every day. The severity of this scenario is alarming, the 97% of the world water being saline water and the most part of the rest being frozen in glaciers, ice and snow. Moreover, the resources of fresh water are decreasing. For all these reasons, water desalination is a very topical issue. Desalination is a process that permits to obtain fresh water from salt water. This is possible because the process separates dissolved salts and other minerals from water. Seawater desalination has the potential to reliably produce enough potable water to support large populations located near the coast. Studying this plant is consequently necessary to reduce the cost of desalinated water, which is still too high for many people.

There are many methods to implement a desalination plant but all of them can be categorized in two main families: thermal techniques and the membrane-based technique. Thermal techniques use energy to boil or freeze the salt water. These two physical transformations are able to separate salt from water. Unfortunately, they need a lot of energy and research is still looking for a less wasteful of energy method. In this field, renewable energies are particularly important. They permit to obtain thermal energy to boil water

without electrical or fossil fuel consume.

The membrane-based techniques used special membrane to filter the salt water. In this way the salt is retained by the membranes and fresh water is produced. These techniques require less energy but they have other technical problems related to the membranes.

Thermal techniques, as listed below, can be:

- *vacuum distillation*: it consists in boiling salt water in an ambient where the atmospheric pressure is reduced. Pressure reduction is essential to decrease the boiling temperature of water. Lower boiling temperatures result in a lower consumption of energy
- *multi stage flash distillation*: the distillation process is realized with a series of flash evaporations
- *multiple effect distillation (MED)*: it consists of multiple stages of effects. In each stage the water is heated by the steam in the tube. This heating produces steam, which flows into the tubes of the next stage (effect), heating and causing the evaporation of more water.
- *vapor compression distillation*: the compressed vapor is used to boil the salt water
- *freeze thaw*: opposite to the previous method, in this technique freezing sea water is used to obtain fresh water
- *solar evaporation*: it is the natural cycle. The sun evaporates the water and the result of it is collected on a cold surface.

The membrane-based technique is also known as:

- *reverse osmosis*: the salt water is pushed through a semipermeable membrane (with the membrane rejecting salts). This technique to produce fresh water typically uses less energy than thermal desalination process, but it depends on the salt concentration in the water. Reverse osmosis and Nanofiltration are the leader techniques at the moment.

- Other methods are implemented in the desalination plant but they are not mentioned here because they are rarely used.

This thesis deals with a control problem on a MED desalination plant. In the next paragraph it is explained more specifically the MED desalination method.

1.1 Multiple-Effect Distillation (MED)

MED desalination process was developed in 1950 and it is based on the physical principle of distillation. This means that it needs energy to produce steam, which places MED process among the thermal techniques. In this process the main problem is the huge quantity of energy consumption. In order to exploit it and reduce its waste, a multi effect distillation plant was developed. MED is more energy-efficient than other evaporation techniques but it's also the most sophisticated. In this type of desalination plant is possible to used some sustainable source of energy, for example exhaust steam from the power station turbines. This opportunity makes these plants more competitive than others on the technical side, but requires higher costs and a wider range of instruments (a broader equipment).

1.1.1 Functioning

As shown in Fig. 1.1, MED evaporator consists of several consecutive cells (called effects) maintained at a progressive decreasing level of pressure and temperature from the first hot cell to the last cold one. The working sequence is described below:

1. The high temperature steam is introduced inside the first effects tubes
2. The tubes are cooled externally thanks to a flow of salt water
3. Inside the tubes, the steam results in a distillate
4. At the same time part of the saltwater evaporates as a consequence of the condensation heat

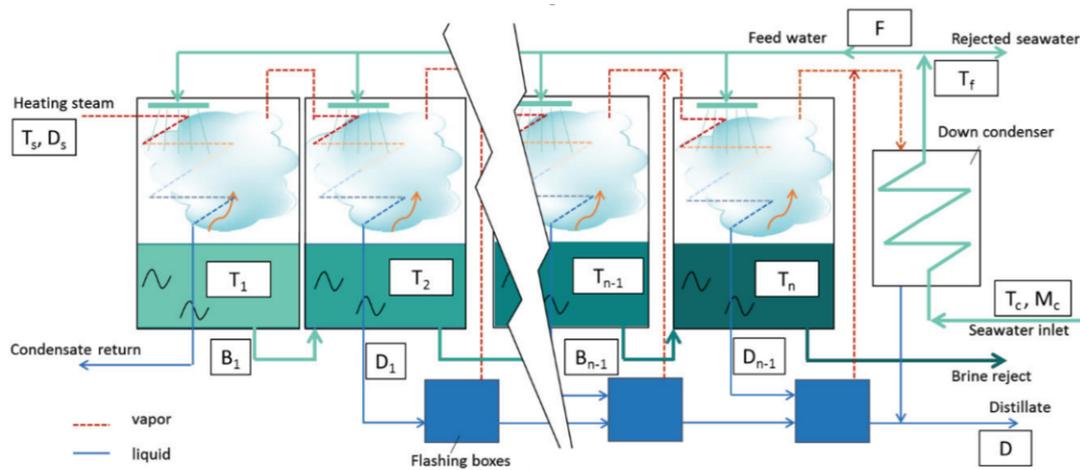


Figure 1.1: Scheme of MED plant, water cycle in the different effects

5. This 'second stage of vapor' is at a lower temperature than the first heated steam and it is used into the second effect cell to repeat the process.
6. This additional vapor will condense into a distillate inside the next cell
7. In the last cell, the produced steam condense on a conventional shell and on tube heat exchangers.

The salt doesn't evaporate and what is obtained in a condensed form is fresh water. On the bottom of each effect tank remains a brine solution (salt water) and which can be thrown back into the sea. In each tank the evaporation process is possible because temperature and pressure decrease together, the boiling temperature decreasing according to the pressure. For the reasons just mentioned, the vapor boiled off in one vessel can be used to heat the next one and only the first one, being characterised by a higher pressure, requires an external source of heat.

1.1.2 Configuration

FORWARD FEED MED In a plant with N effects developed according to the forward feed configuration (see Fig. 1.2), it is present $N - 1$ brine preheaters. These preheaters

use a part of the vapor produced in each effects to increase the inlet brine temperature. The flow of the brine and the flow of the vapor in the preheaters are opposite. Part of the vapor condensing in the preheaters and this condensate is collected as total distillate product after the last effect.

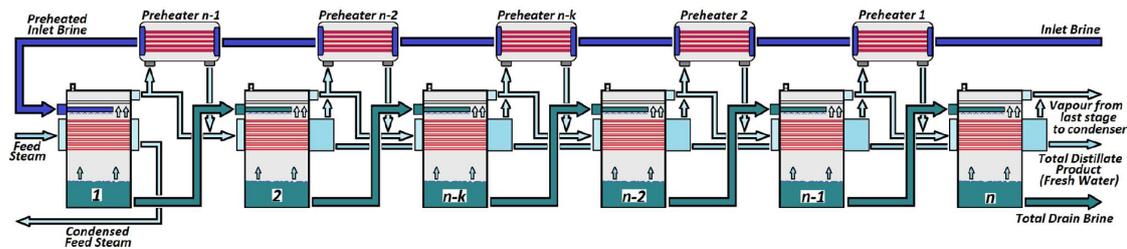


Figure 1.2: Forward feed MED plant

PARALLEL FEED MED In this configuration the input brine flow is split in N inlet flows, where N is the number of effects. The drain brine collected in the N tanks flows to the $N + 1$ effect's brine tank and different pressure in the effects permits to produce vapor in the brine pool too.

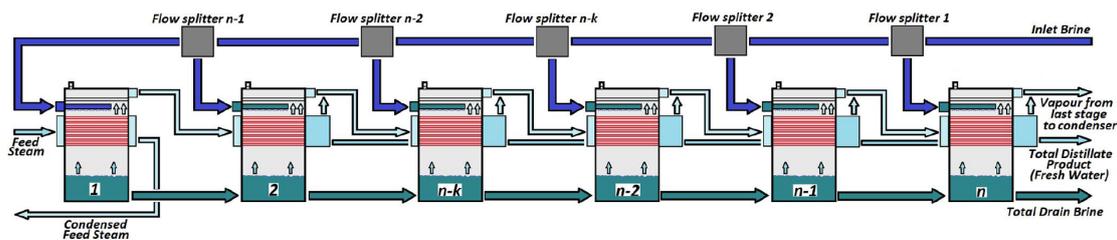


Figure 1.3: Parallel feed MED plant

In both the configuration it is necessary to produced the feed steam for the first effect of the MED with an external source of energy. It is very important to produce an adequate vapor in this phase of the project because its temperature and pressure heavily affect the performance of the plant.

1.2 Solar collectors

Solar collectors transform solar radiation into heat and transfer that heat to a medium (water, solar fluid, or air). Then solar heat can be used for heating water, to heating or cooling systems. Solar cooling technologies demand high temperatures and not all the type of solar collectors are capable of producing them. Flat-plate collectors are the most widely used kind of collectors in the world for domestic water-heating systems and solar space heating/cooling. The first accurate model of flat plate solar collectors was developed by Hottel and Whillier in the 1950's. A flat-plate solar collectors consist in:

- absorber: it is usually a lamina made in high thermal conductivity metal with tubes. This tubes, located in the absorber, converts into heat the solar radiation and transfer this heat to a fluid flowing through the collectors. Its surface is coated to maximize radiant energy absorption and to minimize radiant emission
- transparent cover sheets: it allow sunlight to pass through the absorber but also insulate the space above the absorber to prevent cool air to flow into this space.
- insulated box: to reduces heat losses

There is two main categories: sun tracking and stationary. The first follow the sun during the day so they always have the maximum incident irradiation. The second one don't do that. Another way of categorizing them is between: non concentrating or stationary and concentrating. A non concentrating collector has the same area for intercepting and for absorbing solar radiation. Concentrating solar collector usually has concave reflecting surfaces to intercept and focus the sun's beam radiation to a smaller receiving area. In this thesis flat plate solar collectors are used. They are shown in Fig. 1.4. In that scheme is possible to see the parts described above.

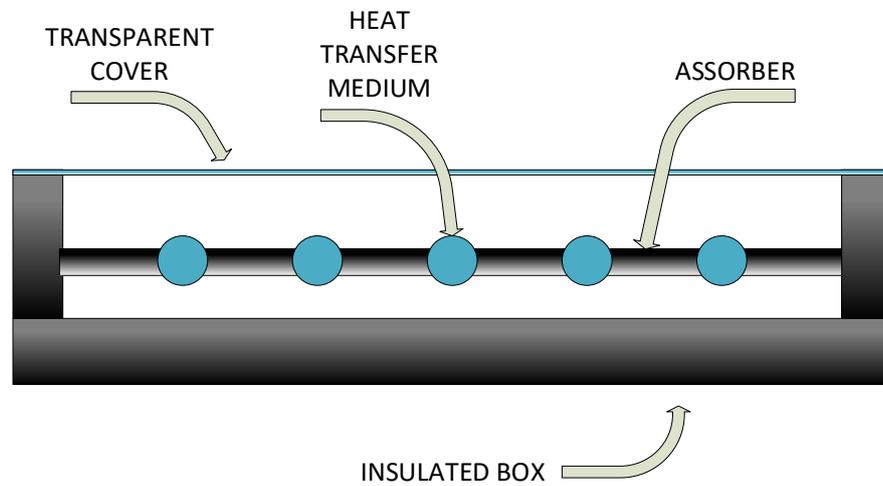


Figure 1.4: Flat plate solar collectors scheme, main part connections



Figure 1.5: Flat plate solar collectors photo

1.3 Plataforma Solar de Almeria

Located in the Taberna desert, thirty kilometers north of Almeria, there is the PSA. The *Plataforma Solar de Almeria - PSA* is the biggest research center on concentrated solar energy in Europe. This research center belongs to CIEMAT (*Centro de Investigaciones Energeticas, Medioambientales y Tecnologicas*) which depends on MINECO (*Ministerio de Economia y Competitividad*). The fields mainly studied are those connected to the concentration of solar thermal energy and to solar photochemistry.

Here a MED water desalination unit is located. The main objective of this unit is to widen the knowledge in the field of desalination technologies supported by renewable energies. The installation is divided into the following blocks:

- SOL-14 solar thermal seawater desalination plant
- stationary collector test platform

This thesis is focused on the control of the hydraulic circuit in the stationary flat plate solar collectors. This part of the system is called AQUASOL-II.

1.3.1 AQUASOL-II

The solar field AQUASOL-II (see Fig. 1.8) located into the Plataforma Solar de Almeria (PSA) in Spain, is composed of 60 stationary flat plate solar collectors (Wagner LBM 10HTF) with a total aperture area of 606 m^2 . Its distribution is as follows (see Fig. 1.7): there are five loops connected in parallel. Four of these (loops 2-5) have 14 flat-plate collectors each, and each loop has two rows connected in series with 7 collectors in parallel per row, while loop 1 has 4 flat-plate collectors connected in parallel. There is a main pumping system (Pump 0) and each loop has its own pumping system (Pump 1 - Pump 9). The main pump is necessary to help the other loops pump to reach a high level of flow. As Fig. 1.7 shows, the water enters into the main pipe and it is pushed by the main pump through the loops. In loops 2-5 the water is pushed across the flat plate solar collectors by

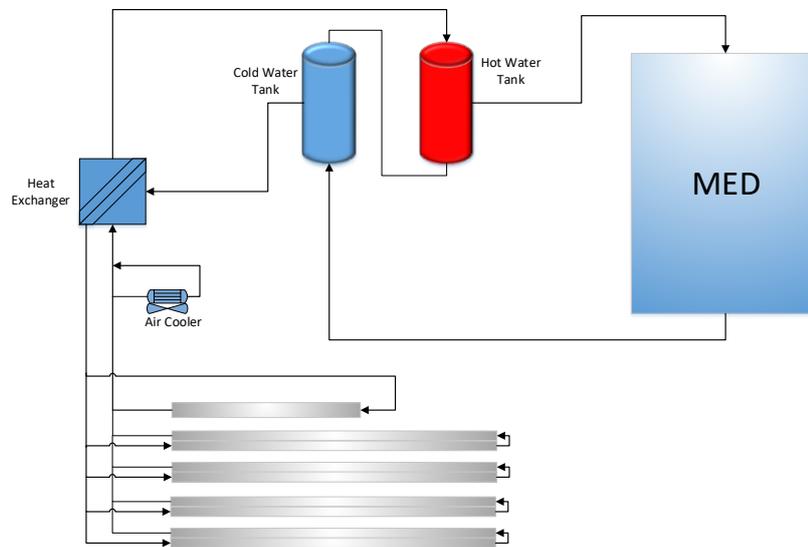


Figure 1.6: AQUASOL-II solar field facility at PSA (Spain) - Block diagram

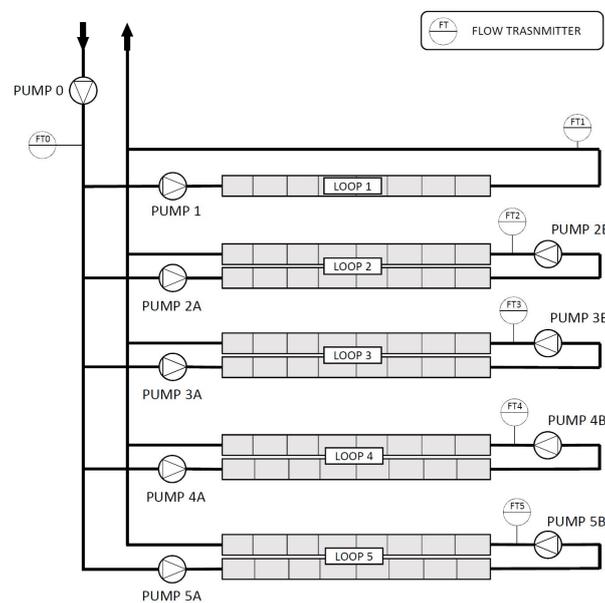


Figure 1.7: AQUASOL-II plant , stationary flat plate solar collectors configuration

two pumping systems (the one on the left and the one on the right) and it flows to the main pipe to continue its cycle into the plant. Each pumping system is composed of two pumps



Figure 1.8: AQUASOL-II solar field facility at PSA (Spain) - Photo

in parallel configuration but, from the control point of view, the pumps inside each loop are treated as a single one (having the same control signal). Therefore, the manipulated variable will be one for loop and each pump variable-frequency drive will receive the same value.

It is important to mention that, during the experimental campaigns performed to obtain the model and to test the controller, the loop number 2 was out of order. For this reason, this loop is not considered neither in the modelling stage, nor in the control design.

In the MED systems the water inlet temperature is very important. This thesis will deal with the solar field able to transfer this energy to the water. The ultimate goal is to control the outlet temperature of the water from each loop of the solar field.

First, the system will be modeled, and then the MIMO control system will be developed.

Chapter 2

Models

2.1 Introduction

In this chapter are treated the different methodologies to obtain the model of the system. With the aim of tuning a controller to regulate the water flow rates inside the solar loops, two hydraulic models of the system have been experimentally obtained. Since the plant is a multiple-input multiple-output (MIMO) system with 5 inputs (pump speeds) and 5 outputs (water flow rates), the basic transfer function model is $\mathbf{y}(s) = \mathbf{G}(s)\mathbf{u}(s)$, where \mathbf{y} and \mathbf{u} are 5x1 vectors and $\mathbf{G}(s)$ is a 5x5 transfer function matrix. There is interaction between inputs and outputs because a change in one of the inputs affects all the outputs. To obtain the values of matrix $\mathbf{G}(s)$, an experimental campaign was designed and performed; different steps were applied in the loops pumps to obtain the water flow rates responses via the reaction curve method. This technique is appropriate because the system has a stable behavior and it does not present a pole in zero in its transfers functions [1].

The experiments to find the data were planned as follows (see Fig. 1.7):

- The loop pumps (pump 1, pump 3a, pump 3b, pump 4a, pump 4b, pump 5a, pump 5b) were examined with steps from 40% to 90% of the input range.
- The experiments described in the previous point were repeated with different values

of the principal pump (pump 0) input. In particular, the principal pump was set to 20% 50% 80% of its input range.

For a good data set it was necessary to take the data when the water temperature was close to the operating point as the density of water changes according to its temperature. To obtain it, it was necessary to make the experiments in a day characterised by good solar radiation and the inlet temperature was controlled through an air-cooler located in the main pipe of the solar field. In order to use as less as possible the air-cooler, the experiment with the principal pump at 20% was done in the morning, when the solar radiation was not too high and the other one was done in the afternoon. The reason for doing this was the fact that lowering the speed of the pumps causes the increase of the water temperature in the loops. Because of this, the temperature of the water at the inlet of the solar field was maintained at $50^{\circ}C$ and $60^{\circ}C$. With this value of temperature the safety condition was guaranteed and it was possible to carry out the experiment without any problem.

in Fig. 2.1 the data found with the experiments described above are presented. It is possible to observe an incongruous shape of input and output in the initial and in the final part of the experiments. This is a consequence of the procedures of launch and shutdown. It is easily visible that this is a MIMO system because the input of each loop significantly affects each output. On closer inspection, the figure shows the presence of positive zeros in the non-direct transfer functions.

During the experimental campaign performed to obtain the transfer functions models, it was observed that the static gain of the process change with the operating conditions due to the non-linearities of the actuators (pump). In particular the gain of the models of the loop pumps, depend on the working point of the main pump. This is reasonable because the energy losses in the pipe, as the Bernulli equation states, depends on the square of the water speed.

In the next sections two system models can be found, the first using the gain scheduling

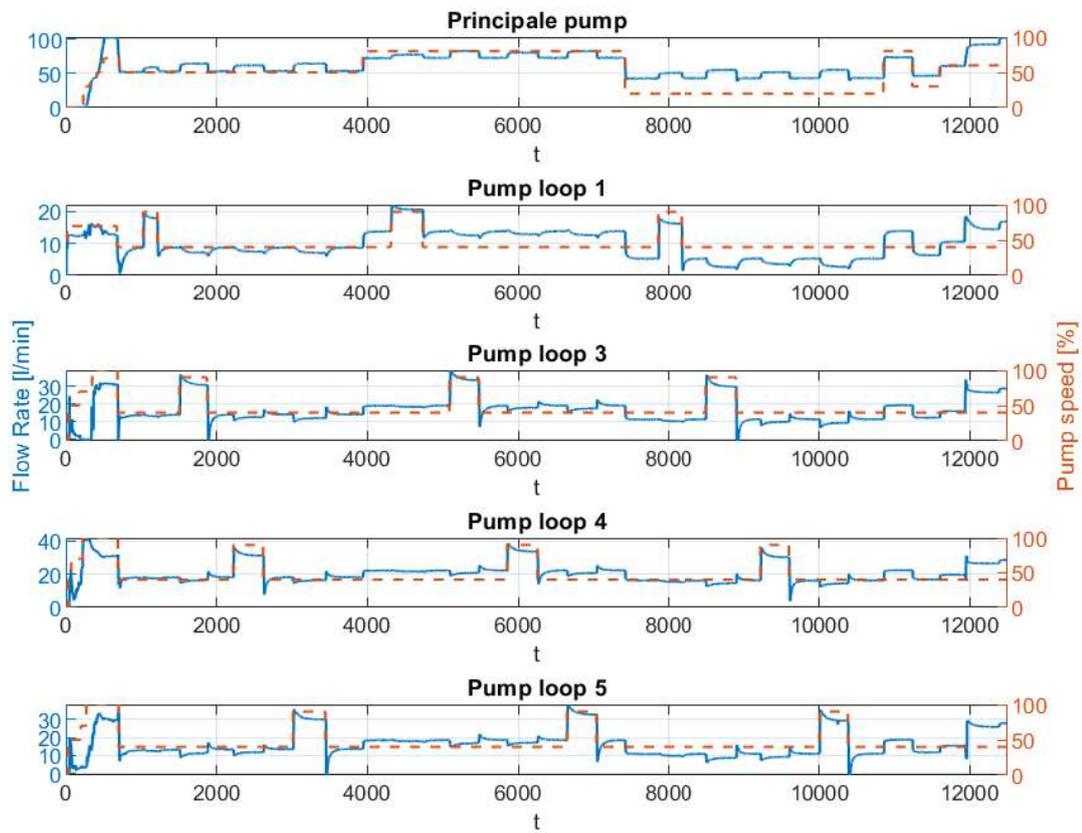


Figure 2.1: AQUASOL-II solar field data for the model identification, 15-03-2018

principle (piecewise linear model) and the second composed by a linear and a non-linear part.

The Flow-Temperature model will be presented in the last section of this chapter. Unlike the pump-flow model it was not designed from scratch but a pre-existing model was modified in order to be useful for this thesis.

2.2 Flow model 1 - gain scheduling

2.2.1 Introduction

This first model will be used to design a Gain Scheduling Controller. To this end, it will be used to tune PIDs and carry out simulations. The first idea for creating a model of the system is to use simple transfer functions for each loops, 1 direct and 4 indirect transfer functions. Nevertheless, the solution of having a simple transfer functions on each loop can be too inaccurate, the static gain of each loop is not constant. For this reasons the system is not linear. In order to maintain a simple model, it was developed a procedure that allows us to use transfer functions. Clearly, an error will be introduced, but it is considered acceptable for this controller purposes. After an accurate study it is possible to affirm that the static gain of each loop depends on the principal pump input. the model was split in two different parts, one working when the input on the principal pump is under 50% and the other one working when the input is over that level. This approach can be labeled as *gain scheduling*. In this case the scheduling happen in the model in order to have always the most precise transfer function in the simulation.

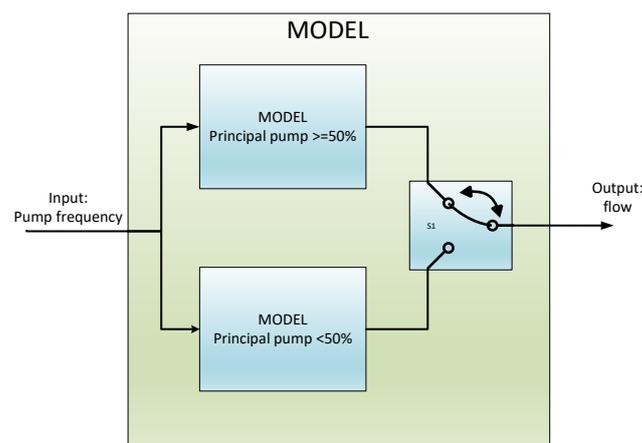


Figure 2.2: Gain scheduling model, basic scheme

As it was said before, the switch statement of the model is given by the signal U_0 (principal pump input) so it presents two situations (see Fig. 2.2):

- situation 1: $U_0 < 50\%$
- situation 2: $U_0 \geq 50\%$

In the model described in this chapter, not all the transfer functions change from the first to the second situation. Since the purpose is to avoid a too complex model, the indirect transfer functions (out of the principal pump) are kept constant. This is acceptable because the variations in it are negligible.

The procedure followed to find each transfer function is:

1. to define an adequate number of pole and zero,
2. to determine the delay,
3. to find the pole and zero with the *Ident Matlab Toolbox*,
4. to find the static gain in a define operating range,
5. to merge the information on pole, zero, delay and gain to obtain a suitable transfer function.

To obtain the necessary data, steps were used on the loops pumps and all of these steps were repeated for different values of the principal pump operating point. In particular, the data used to find the part of the model with the principal pump input under the 50% are those with the principal input fixed at 20% and 50%. For the upper part data is used with the principal pump input at 50% and 80% (see Fig. 2.1). The gain is calculated as the average of the two values obtained. In this way it is possible to reduced the error along the whole range. The data was also filtered in order to delete the signal noise but it did not change the results significantly. This is not surprising given the fact that the signal noise doesn't achieve high values and filtering the signal noise doesn't therefore make sense.

The nomenclature used to distinguish each transfer function G_{ij} is composed by two numbers:

- first (i): affected loop
- second (j): pump activated

In particular the principal pipe is denoted by the number 0.

As already mentioned, the model is split in two parts and each of them is composed as Fig. 2.3. In the same image, the main pump is treated as an independent loop. If the model were perfect, its output would be equal to the sum of the other loop pumps outputs. This can also be deduced from the scheme in Fig. 1.7.

In the following sections (Chap. 2.2.2, Chap. 2.2.3 and Chap. 2.2.4) are presented all the transfer functions. Several tests were made with a different number of poles and zeros. Finally, the best and simplest transfer functions were chosen. It makes no sense to use transfer functions with many poles and zeros to have only a marginal benefit.

Below, the number of poles and zero for each transfer function:

- G_{i0} , with $i=0,1,3,4,5$ - 1 pole , 0 zero
- G_{ij} , with $i = j$ $i,j=1,3,4,5$ - 2 poles , 1 zeros
- G_{0j} , with $j=1,3,4,5$ - 2 poles , 0 zeros
- G_{ij} , with $i \neq j$ $i,j=1,3,4,5$, 2 poles , 1 zeros

In Section 2.2.2, 2.2.3, 2.2.4 in the plot are used the following symbols:

- 'X' zeros
- 'O' pole

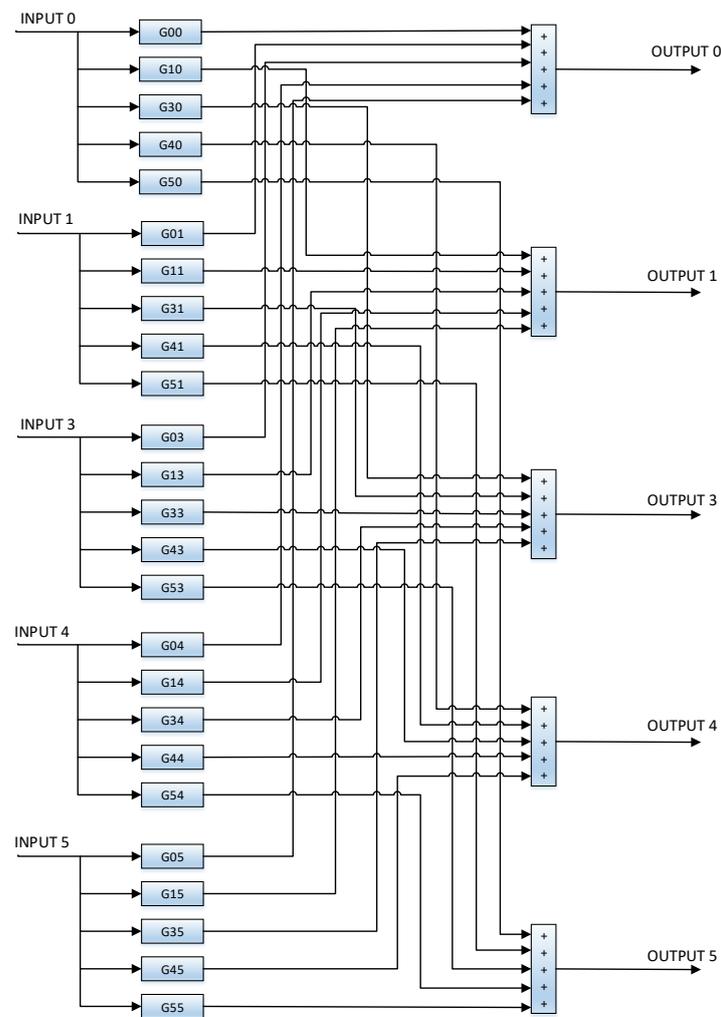


Figure 2.3: MIMO scheme for each part of the gain scheduling model

2.2.2 Transfer function - under 50%

The principal characteristics of the transfer functions developed for an input lower than 50% on the principal pump are presented below. The transfer function's gain presented in this section was found by calculating the average value between the gain obtained with the principal pump (fixed) at 20% and the one obtained with the principal pump at 50%.

- G_{00} - See Fig. 2.4

Rise Time	8.9174 s
Setting Time	19.8776 s
Overshoot	0 %
Undershoot	0 %

$$G_{00}(s) = \frac{0.08664}{s + 0.2464} e^{-4s}$$

- G_{10} - See Fig. 2.5

Rise Time	16.2668 s
Setting Time	32.9613 s
Overshoot	0 %
Undershoot	0 %

$$G_{10}(s) = \frac{0.01569}{s + 0.1351} e^{-4s}$$

- G_{30} - See Fig. 2.6

Rise Time	5.1030 s
Setting Time	14.0855 s
Overshoot	0 %
Undershoot	0 %

$$G_{30}(s) = \frac{0.03493}{s + 0.4306} e^{-5s}$$

- G_{40} - See Fig. 2.7

Rise Time	5.1030 s
Setting Time	14.0855 s
Overshoot	0 %
Undershoot	0 %

$$G_{40}(s) = \frac{0.027}{s + 0.4306} e^{-5s}$$

- G_{50} - See Fig. 2.8

Rise Time	5.8605 s
Setting Time	15.4339 s
Overshoot	0 %
Undershoot	0 %

$$G_{50}(s) = \frac{0.03289}{s + 0.375} e^{-5s}$$

- G_{11} - See Fig. 2.9

Rise Time	5.2407 s
Setting Time	103.5082 s
Overshoot	15.5201 %
Undershoot	0 %

$$G_{11}(s) = \frac{0.0593s + 0.001289}{s^2 + 0.2629s + 0.006407} e^{-6s}$$

- G_{33} - See Fig. 2.10

Rise Time	4.6855 s
Setting Time	212.9448 s
Overshoot	31.1090 %
Undershoot	0 %

$$G_{33}(s) = \frac{0.103s + 0.001089}{s^2 + 0.2189s + 0.00306} e^{-5s}$$

- G_{44} - See Fig. 2.11

Rise Time	2.8683 s
Setting Time	193.5813 s
Overshoot	36.4728 %
Undershoot	0 %

$$G_{44}(s) = \frac{0.1241s + 0.001431}{s^2 + 0.3321s + 0.005299} e^{-4s}$$

- G_{55} - See Fig. 2.12

Rise Time	4.3798 s
Setting Time	188.4421 s
Overshoot	30.4972 %
Undershoot	0 %

$$G_{55}(s) = \frac{0.1078s + 0.001293}{s^2 + 0.2357s + 0.003719} e^{-5s}$$

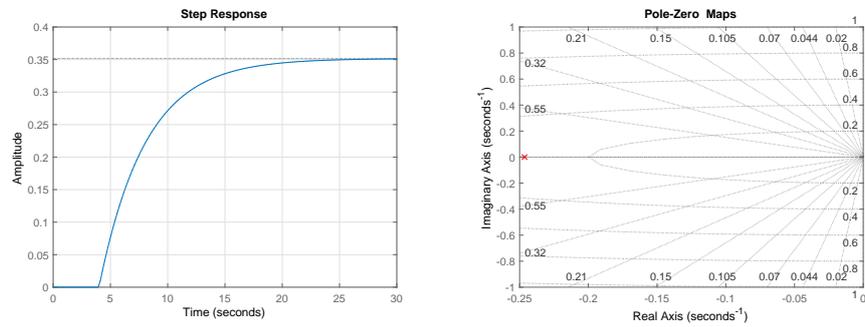


Figure 2.4: Transfer function with $U_0 < 50\%$: G00 - step response , pole-zero maps

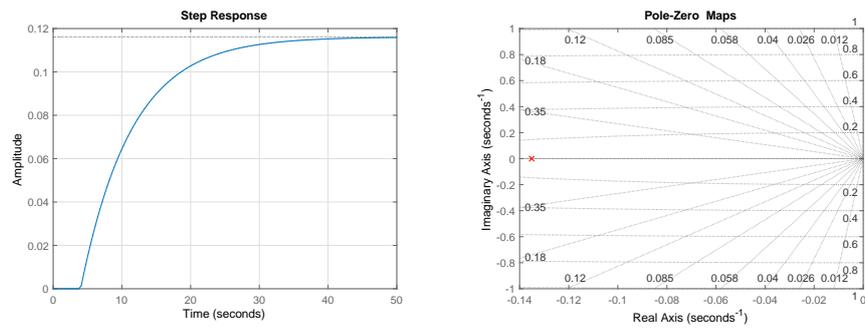


Figure 2.5: Transfer function with $U_0 < 50\%$: G10 - step response , pole-zero maps

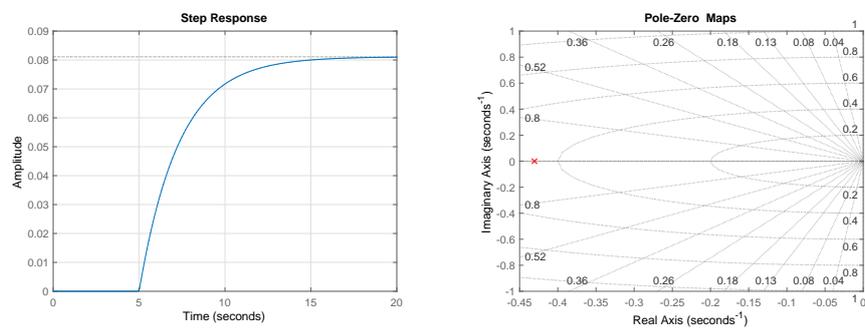


Figure 2.6: Transfer function with $U_0 < 50\%$: G30 - step response , pole-zero maps

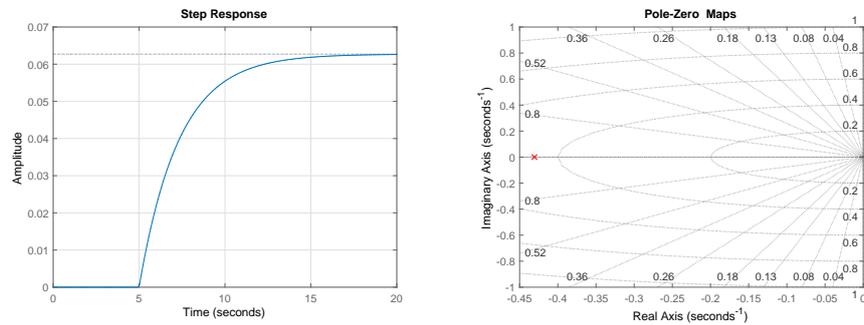


Figure 2.7: Transfer function with $U_0 < 50\%$: G40 - step response , pole-zero maps

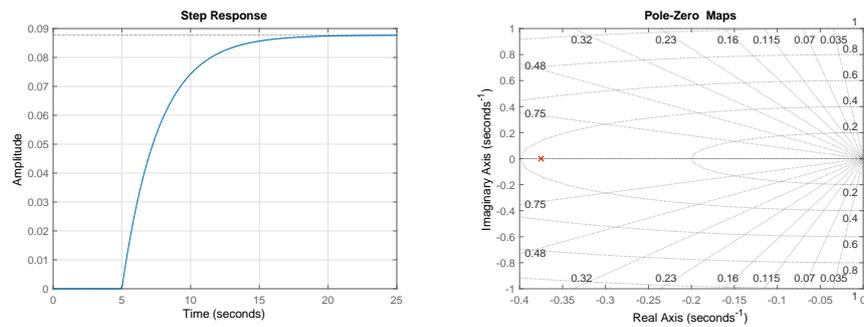


Figure 2.8: Transfer function with $U_0 < 50\%$: G50 - step response , pole-zero maps

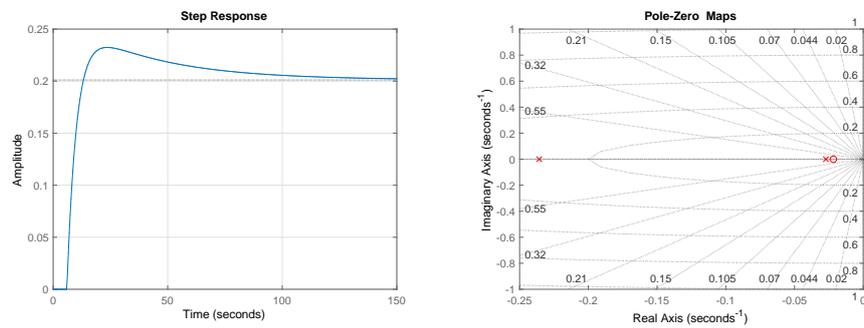


Figure 2.9: Transfer function with $U_0 < 50\%$: G11 - step response , pole-zero maps

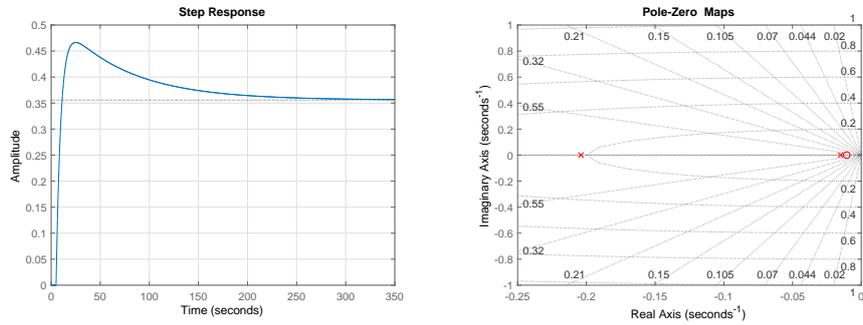


Figure 2.10: Transfer function with $U_0 < 50\%$: G33 - step response , pole-zero maps

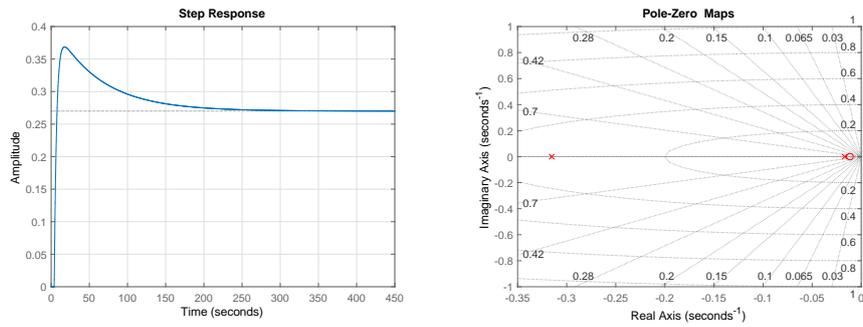


Figure 2.11: Transfer function with $U_0 < 50\%$: G44 - step response , pole-zero maps

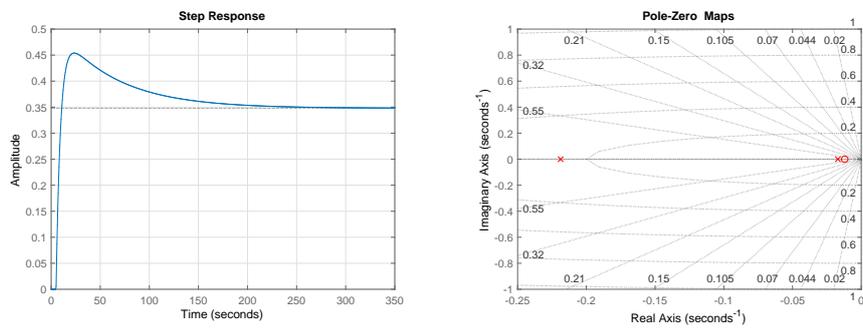


Figure 2.12: Transfer function with $U_0 < 50\%$: G55 - step response , pole-zero maps

2.2.3 Transfer function - upper 50%

The principal characteristics of the transfer functions developed for an input upper than 50% on the principal pump are presented below. The transfer function's gain presented in this section was found by calculating the average value between the gain obtained with the principal pump (fixed) at 50% and the one obtained with the principal pump at 80%.

- G_{00} - See Fig. 2.13

Rise Time	8.9174 s
Setting Time	19.8776 s
Overshoot	0 %
Undershoot	0 %

$$G_{00}(s) = \frac{0.147}{s + 0.2464} e^{-4s}$$

- G_{10} - See Fig. 2.14

Rise Time	16.2668 s
Setting Time	32.9613 s
Overshoot	0 %
Undershoot	0 %

$$G_{10}(s) = \frac{0.02374}{s + 0.1351} e^{-4s}$$

- G_{30} - See Fig. 2.15

Rise Time	5.1030 s
Setting Time	14.0855 s
Overshoot	0 %
Undershoot	0 %

$$G_{30}(s) = \frac{0.0831}{s + 0.4306} e^{-5s}$$

- G_{40} - See Fig. 2.16

Rise Time	5.1030 s
Setting Time	14.0855 s
Overshoot	0 %
Undershoot	0 %

$$G_{40}(s) = \frac{0.0648}{s + 0.4306} e^{-5s}$$

- G_{50} - See Fig. 2.17

Rise Time	5.8605 s
Setting Time	15.4339 s
Overshoot	0 %
Undershoot	0 %

$$G_{50}(s) = \frac{0.06759}{s + 0.375} e^{-5s}$$

- G_{11} - See Fig. 2.18

Rise Time	5.2407 s
Setting Time	103.5082 s
Overshoot	15.5201 %
Undershoot	0 %

$$G_{11}(s) = \frac{0.04732s + 0.001028}{s^2 + 0.2629s + 0.006407} e^{-6s}$$

- G_{33} - See Fig. 2.19

Rise Time	4.6855 s
Setting Time	212.9448 s
Overshoot	31.1090 %
Undershoot	0 %

$$G_{33}(s) = \frac{0.09103s + 0.0009621}{s^2 + 0.2189s + 0.00306} e^{-5s}$$

- G_{44} - See Fig. 2.20

Rise Time	2.8683 s
Setting Time	193.5813 s
Overshoot	36.4728 %
Undershoot	0 %

$$G_{44}(s) = \frac{0.1125s + 0.001297}{s^2 + 0.3321s + 0.005299} e^{-4s}$$

- G_{55} - See Fig. 2.21

Rise Time	4.3798 s
Setting Time	188.4421 s
Overshoot	30.4972 %
Undershoot	0 %

$$G_{55}(s) = \frac{0.09504s + 0.00114}{s^2 + 0.2357s + 0.003719} e^{-5s}$$

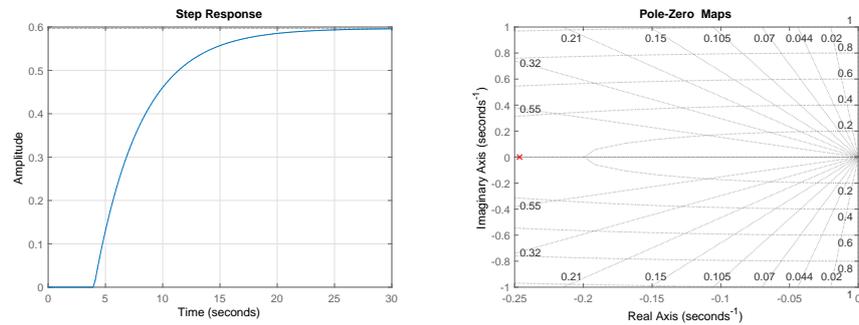


Figure 2.13: Transfer function with $U_0 \geq 50\%$: G00 - step response , pole-zero maps

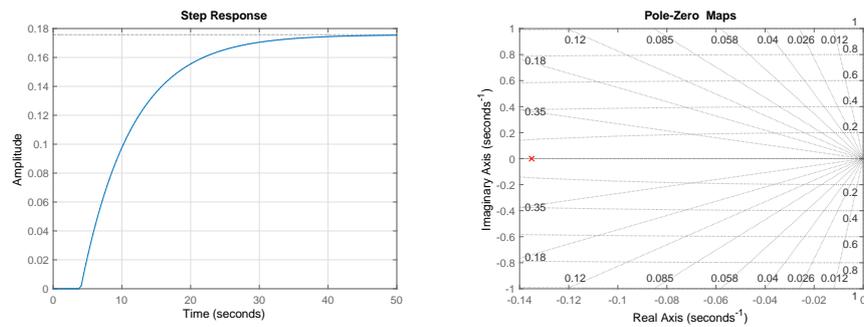


Figure 2.14: Transfer function with $U_0 \geq 50\%$: G10 - step response , pole-zero maps

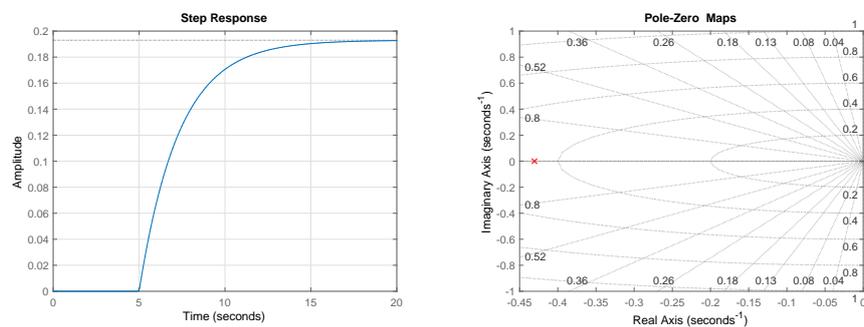


Figure 2.15: Transfer function with $U_0 \geq 50\%$: G30 - step response , pole-zero maps

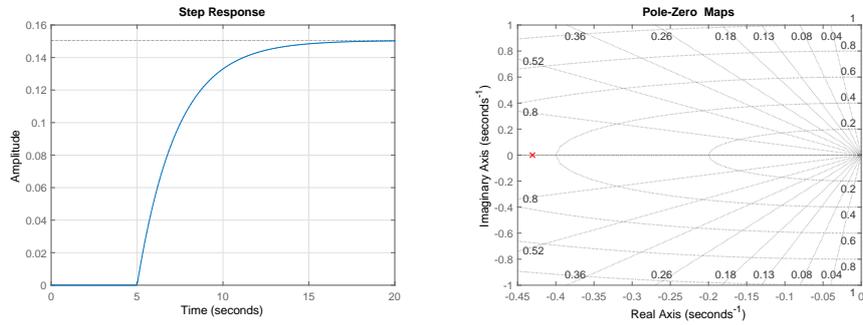


Figure 2.16: Transfer function with $U_0 \geq 50\%$: G40 - step response , pole-zero maps

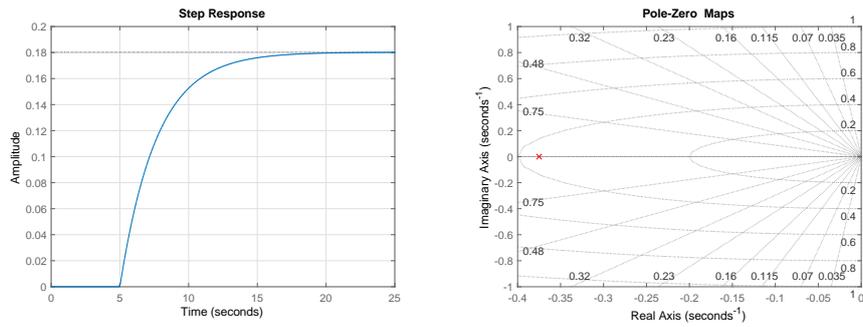


Figure 2.17: Transfer function with $U_0 \geq 50\%$: G50 - step response , pole-zero maps

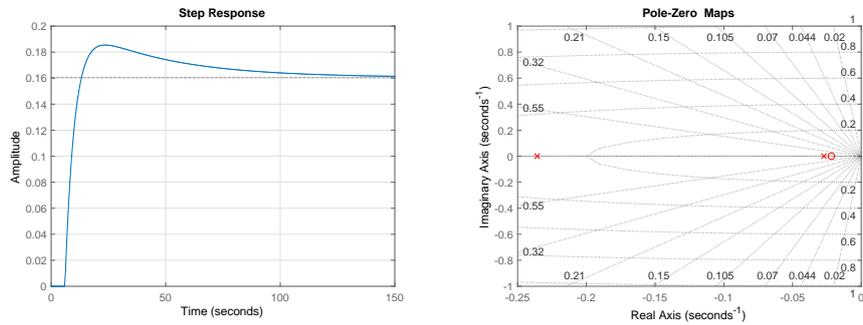


Figure 2.18: Transfer function with $U_0 \geq 50\%$: G11 - step response , pole-zero maps

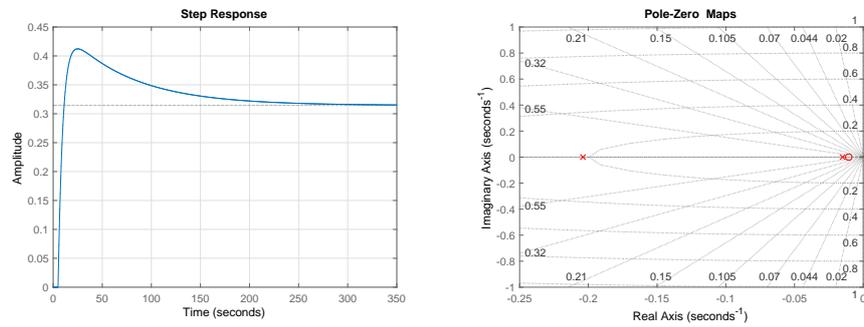


Figure 2.19: Transfer function with $U_0 \geq 50\%$: G33 - step response , pole-zero maps

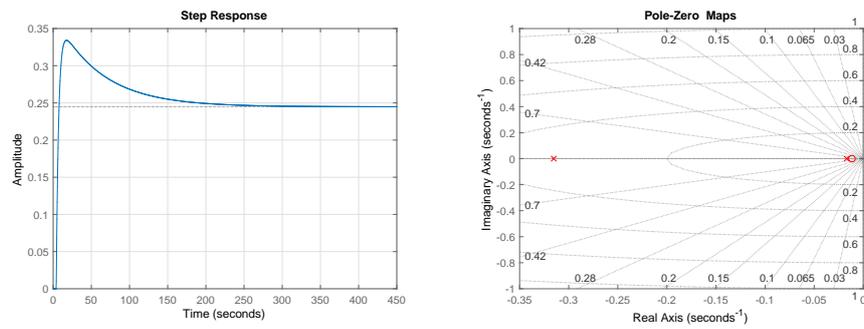


Figure 2.20: Transfer function with $U_0 \geq 50\%$: G44 - step response , pole-zero maps

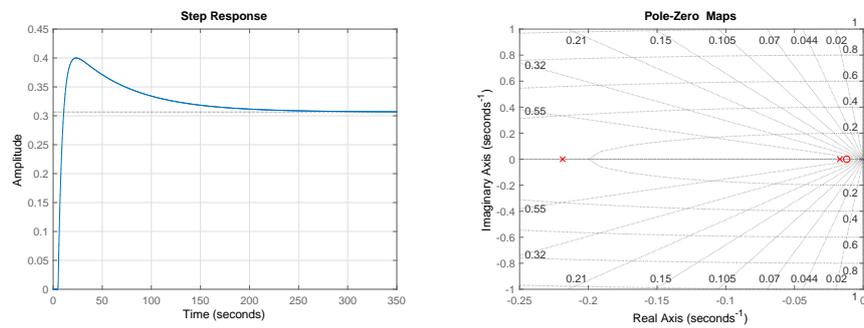


Figure 2.21: Transfer function with $U_0 \geq 50\%$: G55 - step response , pole-zero maps

2.2.4 Indirect transfer function

The indirect transfer functions are not split in two parts as the transfer functions in Cap. 2.2.2 and Cap. 2.2.3. It was decided to keep it constant because the variation is not high and a simple model is more desirable than a complicated one.

- G_{01} - See Fig. 2.22

Rise Time	22.7487 s
Setting Time	45.2797 s
Overshoot	0 %
Undershoot	0 %

$$G_{01}(s) = \frac{0.01174}{s^2 + 1.02s + 0.08994} e^{-4s}$$

- G_{31} - See Fig. 2.23

Rise Time	98.3117 s
Setting Time	191.1840 s
Overshoot	0 %
Undershoot	52.4732 %

$$G_{31}(s) = \frac{0.003695s - 0.0001137}{s^2 + 0.4204s + 0.008897} e^{-8s}$$

- G_{41} - See Fig. 2.24

Rise Time	132.8099 s
Setting Time	252.9202 s
Overshoot	0 %
Undershoot	32.9677 %

$$G_{41}(s) = \frac{0.001948s - 6.974 \times 10^{-5}}{s^2 + 0.3924s + 0.006218} e^{-8s}$$

- G_{51} - See Fig. 2.25

Rise Time	53.0542 s
Setting Time	109.6402 s
Overshoot	0 %
Undershoot	71.3679 %

$$G_{51}(s) = \frac{0.004941s - 0.0001899}{s^2 + 0.4566s + 0.01719} e^{-8s}$$

- G_{03} - See Fig. 2.26

Rise Time	22.5292 s
Setting Time	44.1482 s
Overshoot	0 %
Undershoot	0 %

$$G_{03}(s) = \frac{0.6223}{s^2 + 28.07s + 2.728} e^{-4s}$$

- G_{13} - See Fig. 2.27

Rise Time	99.4288 s
Setting Time	191.7309 s
Overshoot	0 %
Undershoot	21.9225 %

$$G_{13}(s) = \frac{0.003068s - 0.0001756}{s^2 + 0.2837s + 0.005784} e^{-5s}$$

- G_{43} - See Fig. 2.28

Rise Time	2.9758 s
Setting Time	258.6964 s
Overshoot	91.3554 %
Undershoot	0 %

$$G_{43}(s) = \frac{-0.01211s - 8.933 \times 10^{-5}}{s^2 + 0.1751s + 0.002649} e^{-4s}$$

- G_{53} - See Fig. 2.29

Rise Time	1.0921 s
Setting Time	255.7534 s
Overshoot	113.4982 %
Undershoot	0 %

$$G_{53}(s) = \frac{-0.03807s - 0.0002712}{s^2 + 0.4268s + 0.006808} e^{-8s}$$

- G_{04} - See Fig. 2.30

Rise Time	17.4954 s
Setting Time	35.1516 s
Overshoot	0 %
Undershoot	0 %

$$G_{04}(s) = \frac{363.7}{s^2 + 1.682 \times 10^4 s + 2112} e^{-4s}$$

- G_{14} - See Fig. 2.31

Rise Time	109.3793 s
Setting Time	205.5798 s
Overshoot	0 %
Undershoot	31.8276 %

$$G_{14}(s) = \frac{0.005862s - 0.0002739}{s^2 + 0.5889s + 0.01143} e^{-5s}$$

- G_{34} - See Fig. 2.32

Rise Time	1.0621 s
Setting Time	288.0732 s
Overshoot	148.2903 %
Undershoot	0 %

$$G_{34}(s) = \frac{-0.02288s - 0.000121}{s^2 + 0.3572s + 0.004963} e^{-4s}$$

- G_{54} - See Fig. 2.33

Rise Time	1.3287 s
Setting Time	243.6882 s
Overshoot	114.8105 %
Undershoot	0 %

$$G_{54}(s) = \frac{-0.0246s - 0.0001785}{s^2 + 0.3499s + 0.005768} e^{-4s}$$

- G_{05} - See Fig. 2.34

Rise Time	25.5439 s
Setting Time	48.4829 s
Overshoot	0 %
Undershoot	0 %

$$G_{05}(s) = \frac{0.009746}{s^2 + 0.597s + 0.04491} e^{-2s}$$

- G_{15} - See Fig. 2.35

Rise Time	108.5689 s
Setting Time	205.7725 s
Overshoot	0 %
Undershoot	28.4374 %

$$G_{15}(s) = \frac{0.003196s - 0.0001437}{s^2 + 0.3166s + 0.005998} e^{-3s}$$

- G_{35} - See Fig. 2.36

Rise Time	2.6297 s
Setting Time	239.6591 s
Overshoot	94.2502 %
Undershoot	0 %

$$G_{35}(s) = \frac{-0.01659s - 0.0001311}{s^2 + 0.1939s + 0.003195} e^{-4s}$$

- G_{45} - See Fig. 2.37

Rise Time	2.1006 s
Setting Time	279.6755 s
Overshoot	90.4788 %
Undershoot	0 %

$$G_{45}(s) = \frac{-0.01867s - 0.0001308}{s^2 + 0.2606s + 0.003662} e^{-5s}$$

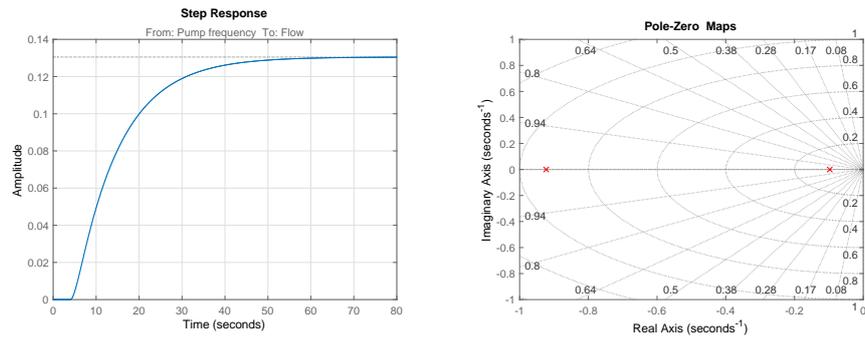


Figure 2.22: Indirect transfer function: G01 - step response , pole-zero maps

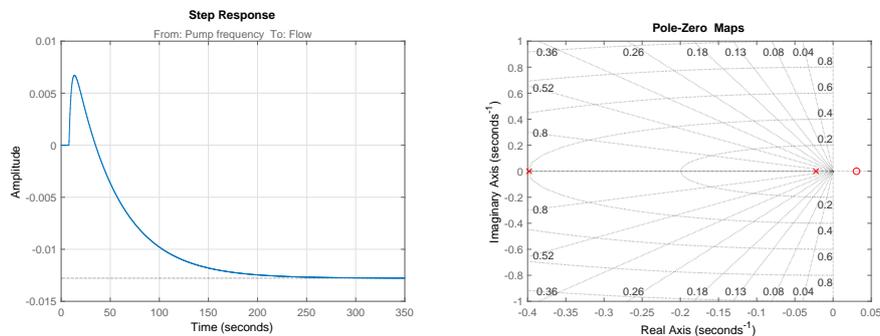


Figure 2.23: Indirect transfer function: G31 - step response , pole-zero maps

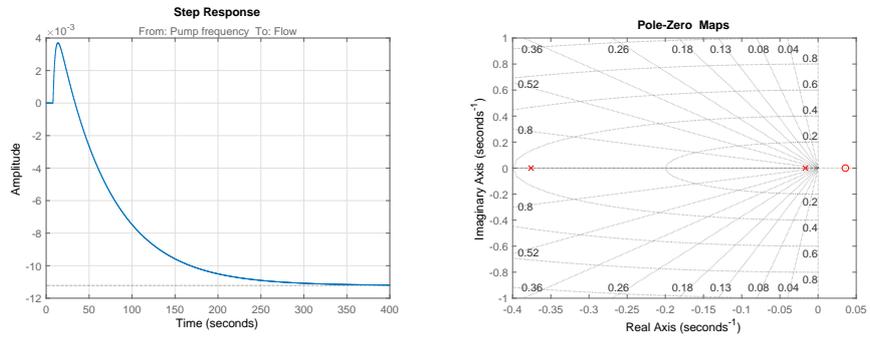


Figure 2.24: Indirect transfer function: G41 - step response , pole-zero maps

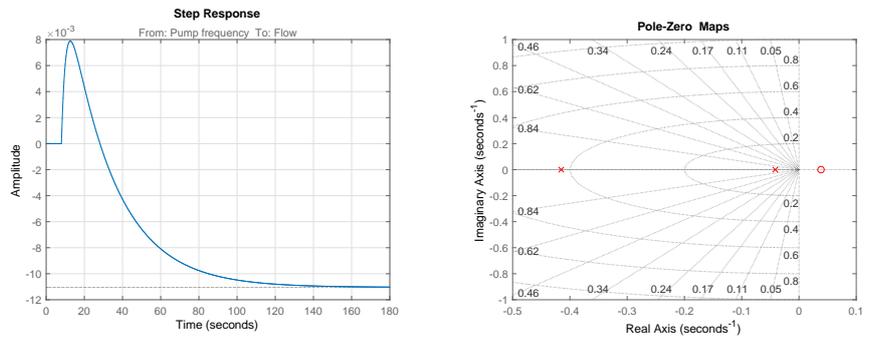


Figure 2.25: Indirect transfer function: G51 - step response , pole-zero maps

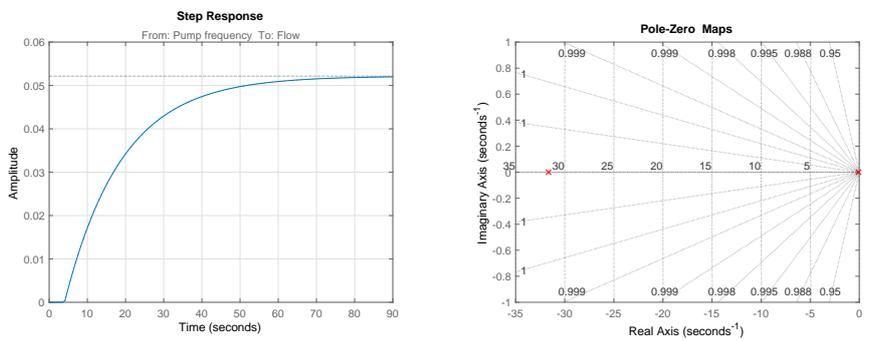


Figure 2.26: Indirect transfer function: G03 - step response , pole-zero maps

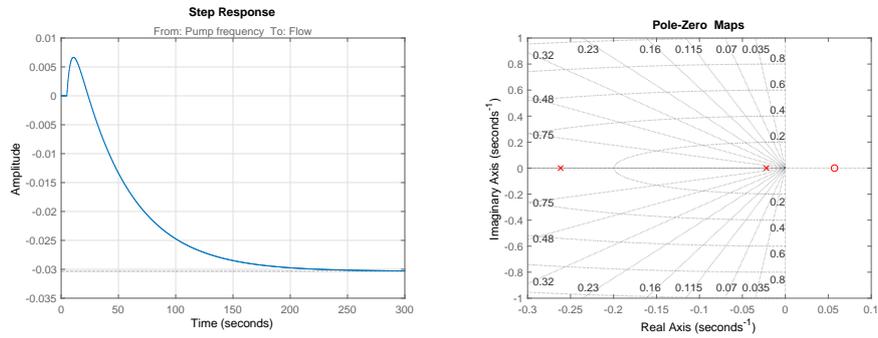


Figure 2.27: Indirect transfer function: G13 - step response , pole-zero maps

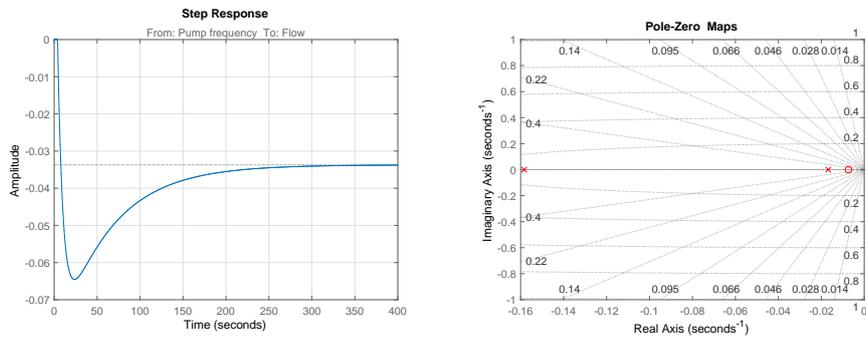


Figure 2.28: Indirect transfer function: G43 - step response , pole-zero maps

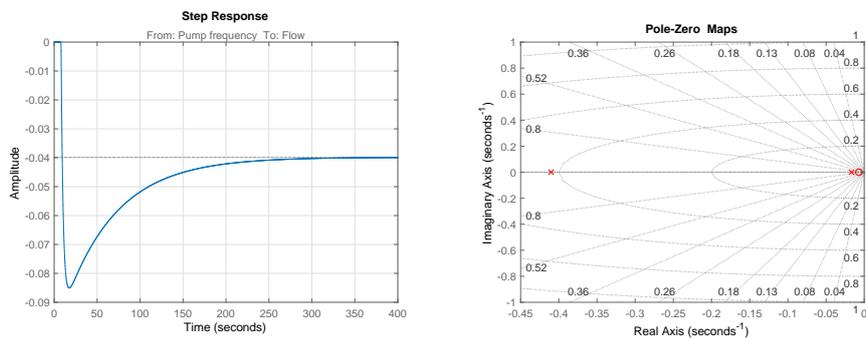


Figure 2.29: Indirect transfer function: G53 - step response , pole-zero maps

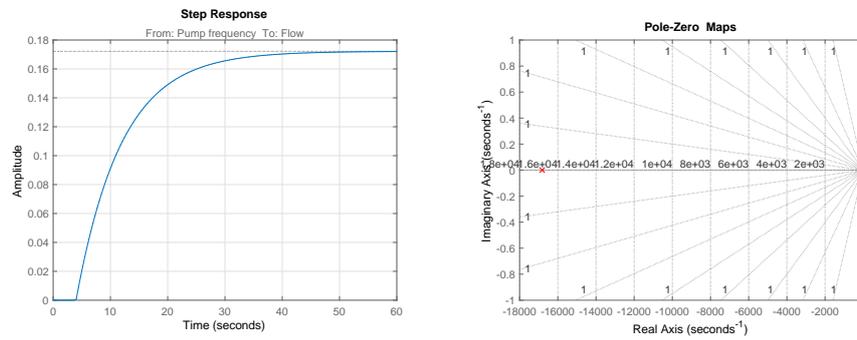


Figure 2.30: Indirect transfer function: G04 - step response , pole-zero maps

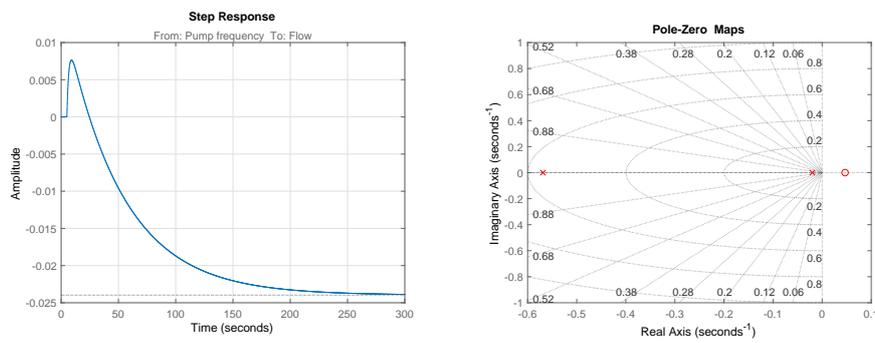


Figure 2.31: Indirect transfer function: G14 - step response , pole-zero maps

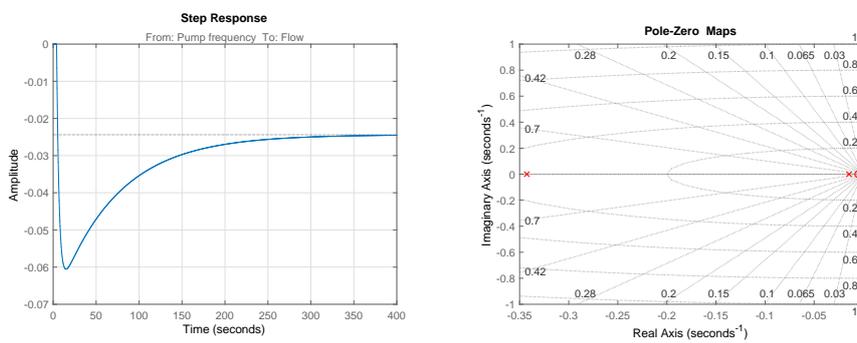


Figure 2.32: Indirect transfer function: G34 - step response , pole-zero maps

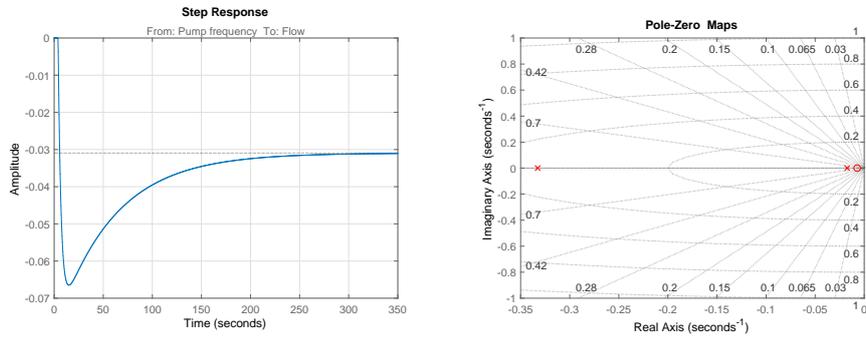


Figure 2.33: Indirect transfer function: G54 - step response , pole-zero maps

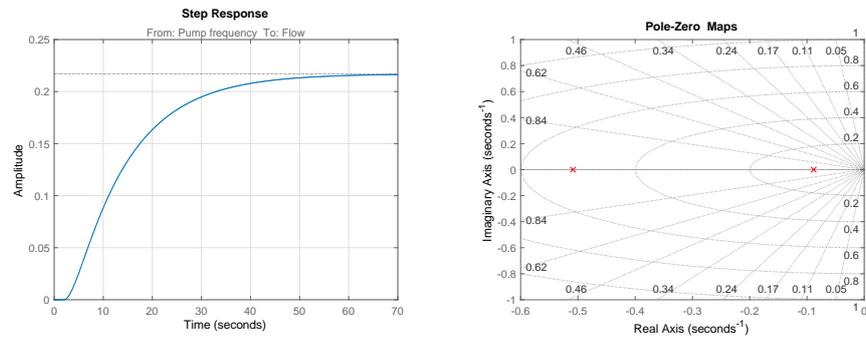


Figure 2.34: Indirect transfer function: G05 - step response , pole-zero maps

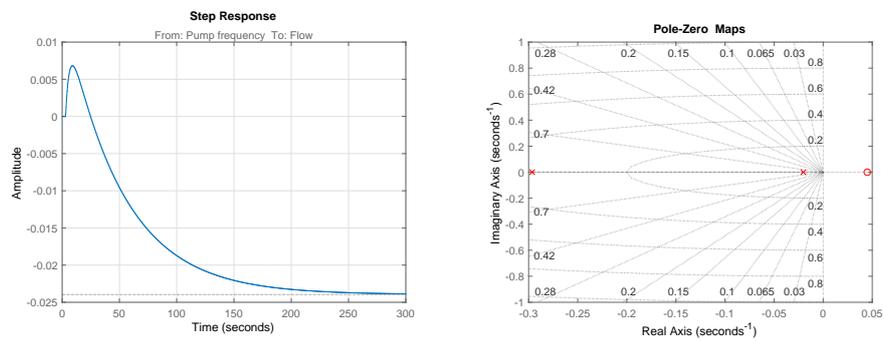


Figure 2.35: Indirect transfer function: G15 - step response , pole-zero maps

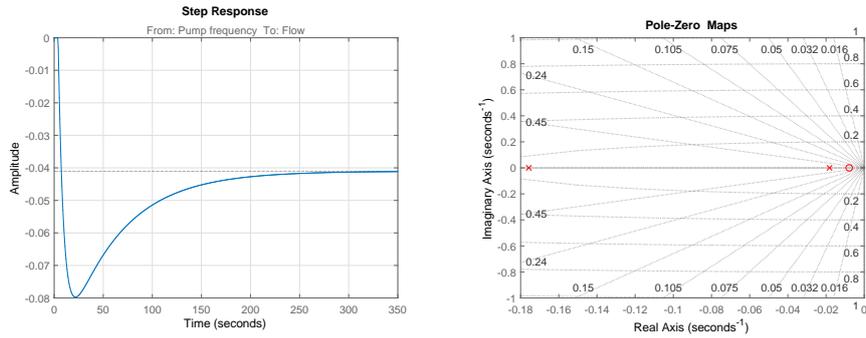


Figure 2.36: Indirect transfer function: G35 - step response , pole-zero maps

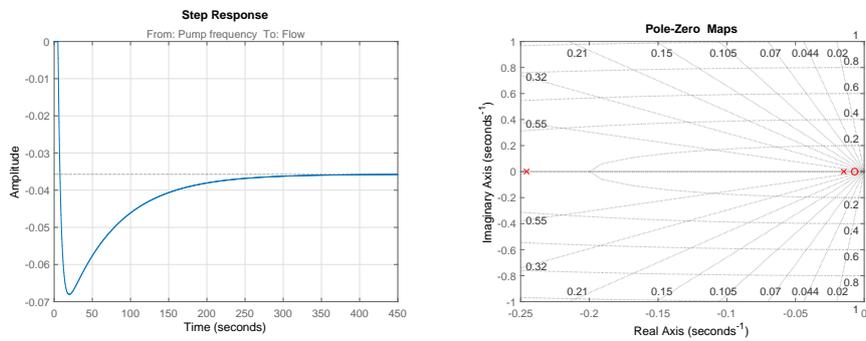


Figure 2.37: Indirect transfer function: G45 - step response , pole-zero maps

As can be seen from the Chap. 2.2.2 and Chap. 2.2.3, the step response information (rise time, setting time, overshoot, undershoot) are the same in the two different situation ($U_0 < 50\%$ and $U_0 \geq 50\%$). This make sense because only the transfer functions gains change a little in the two cases. Pole and zero remains the same. Since the changes in the transfer functions are not high, the step response of them are similar in the two case. With a high number of decimal numbers the differences would emerge but it has no sense to report it.

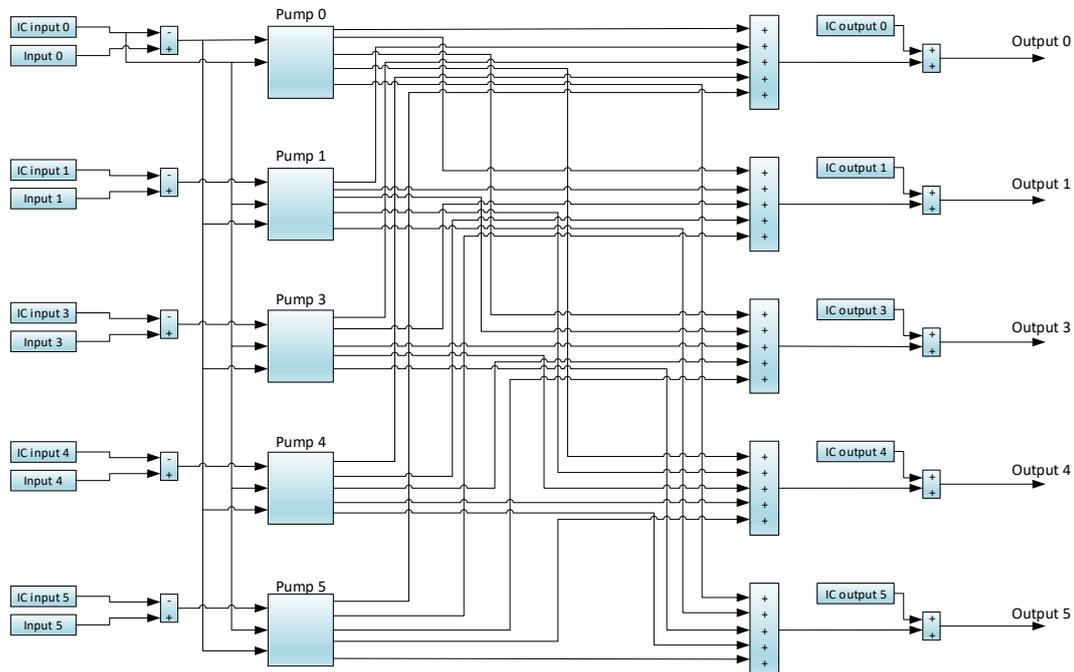


Figure 2.38: Gain scheduling model, simulink scheme

2.2.5 Model scheme

The transfer functions presented in Chap. 2.2.2, Chap. 2.2.3 and Chap. 2.2.4 are used in this section in the Simulink model blocks. Developing a Simulink scheme is necessary, if the purpose is to do a simulation. This will be necessary to test the controller, before placing it in action on the field, in order to prevent dangerous situations. The model will be used also to check the transfer function and the *gain scheduling* idea.

In Fig. 2.38 is presented the general plant. It is possible to evince that it is a MIMO system. It present 4 + 1 input and 4 + 1 output (four represent the loop pipes and one represents the principal one).

Each pump block produces five signals and the flow values are obtained as the sum of five of these (one for each pump block). This means that all the effects are added together in order to produce one output signal for each loop. The main blocks will be explained in the

paragraph below.

Initial condition In the scheme presented in Fig. 2.38 it is possible to see that the initial conditions are constant. They are fundamental because the transfer functions are found through different steps and they don't start from zero. In order to consider those effects, this strategy was followed (see Fig. 2.39):

1. The input in the transfer function of the model is calculated as the difference between the input signal and the initial input step condition (IC input)
2. The signal produced by this system is added to the initial step output condition (IC output) after the transfer function blocks.

In this way, the model will be more accurate in the central part of the operating range and the error will therefore be reduced also in its extremes. This happens because, as already said, the system is not linear. If the initial conditions was not considered in an appropriate way, an offset would appear on the output between the simulation and the reality. With this approach we can simplify the initial conditions problems of the MIMO system. The initial conditions must be added also in the switch system (discussed below). The vales used are the initial step values used in the experiment to find the data for modelling the system.

- $IC_{input,0} = 50\%$
- $IC_{input,j} = 40\%$, $j = 1, 3, 4, 5$
- $IC_{output,0} = 53.597$ l/min
- $IC_{output,1} = 8.539$ l/min
- $IC_{output,3} = 13.566$ l/min
- $IC_{output,4} = 17.894$ l/min
- $IC_{output,5} = 12.896$ l/min

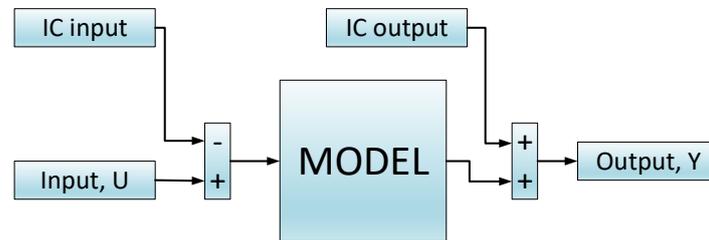


Figure 2.39: Input and output initial condition strategy

Principal pump (Pump 0) In the Fig. 2.40, it is shown the principal pump subsystem (Pump 0 in Fig. 2.38). It presents one transfer function blocks for each loop and what is inside it is show in Fig. 2.41. It evince the two different transfer function and the switch system. First the delay is placed and after the signal provides to control the switch and acts as input signal for the model transfer functions. At the beginning the initial condition was subtracted and in this part, for a correct switch, it is necessary to add back that values. Then to avoid situations of continuous changes it is posed a filter and a relay block. With this strategy it was solved the problem in cases in which the input 0 is around the 50%. The rele's threshold are set to 45% and 55%. The inaccuracy introduced is negligible compared to the gain error introduced by non-linearity.

Loops pump (Pump 1,3,4,5) In Fig. 2.42 are presented the loop pumps blocks. The loops 1, 3, 4 and 5 are made in the same way. Only the direct transfer functions are split in two different situation. All the indirect transfer functions are considered constant over the entire input range, so for these transfer functions the Gain Scheduling is not applied. In Fig. 2.43 it is possible to observe the switch system. It differs from the switch system shown before (for the loop 0) because in this case it is necessary to insert the principal

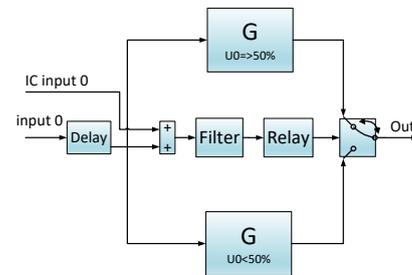
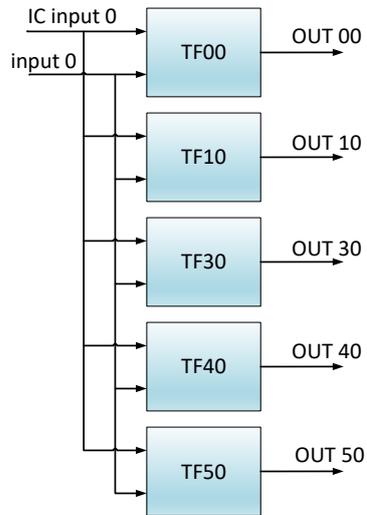


Figure 2.41: tf00 block, inner part of the scheme in Fig.2.40

Figure 2.40: General principal pump scheme, inner part of the scheme in Fig.2.38

pump signal. The rest is the same as it was discussed in the previous paragraph.

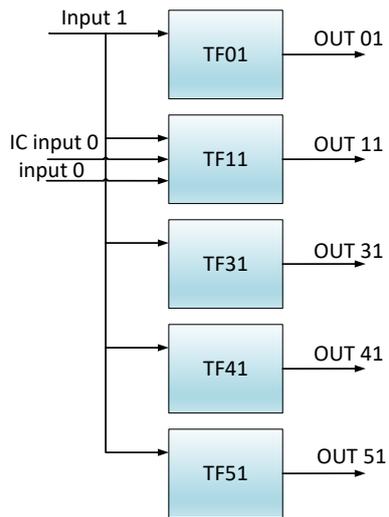


Figure 2.42: General loops pump scheme, inner part of the scheme in Fig.2.38

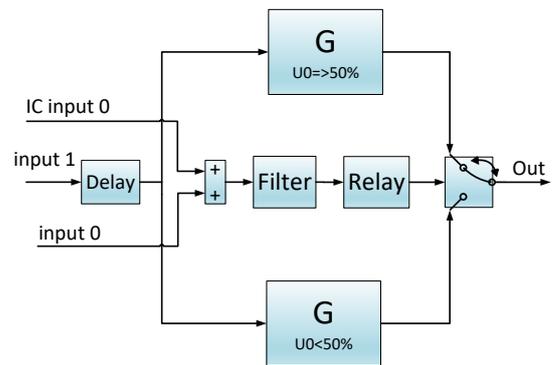


Figure 2.43: tf11 block,, inner part of the scheme in Fig.2.42

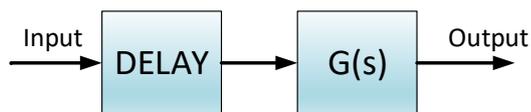


Figure 2.44: tf10 block, inner part of the scheme in Fig.2.42

2.2.6 RGA matrix

Since it is a MIMO system it is advisable to calculate the RGA matrix [2]. It is necessary to consider that the system is non-linear and that it doesn't present a constant static gain. The RGA matrix is calculated using the approximation introduced by the model. This model in particular presents two situations and each of them has constant gain. For this reason two RGA matrix are presented, one in Eqn. 2.1 and one in Eqn. 2.2.

$$RGA_{U_0 < 50\%} = \begin{bmatrix} 6.59 & -1.73 & -1.35 & -1.12 & -1.39 \\ -1.75 & 3.16 & -0.15 & -0.13 & -0.13 \\ -1.36 & -0.16 & 2.92 & -0.15 & -0.25 \\ -1.04 & -0.14 & -0.19 & 2.59 & -0.22 \\ -1.44 & -0.13 & -0.22 & -0.19 & 2.98 \end{bmatrix} \quad (2.1)$$

$$RGA_{U_0 \geq 50\%} = \begin{bmatrix} -4.23 & 1.41 & 1.39 & 1.13 & 1.30 \\ 1.30 & -0.78 & 0.18 & 0.16 & 0.14 \\ 1.47 & 0.14 & -1.01 & 0.16 & 0.24 \\ 1.12 & 0.12 & 0.20 & -0.64 & 0.21 \\ 1.35 & 0.12 & 0.24 & 0.20 & -0.90 \end{bmatrix} \quad (2.2)$$

The matrix in Eqn. 2.1 refers to the system when the input on the principal pump is under 50% and the matrix in Eqn. 2.2 when the input on the principal pump equal or over that value. From Eqn. 2.2 the system appears uncontrollable with simple PIDs. However, it must be considered that loop 0 is controlled in open loop. So, loop 0 is excluded from the calculation of the RGA matrix. The new RGA matrices, without loop 0 are shown in Eqn. 2.3 and Eqn. 2.4. From these matrices it is understandable that the system is well controllable.

$$RGA_{U_0 < 50\%} = \begin{bmatrix} 1.02 & -0.01 & -0.01 & -0.01 \\ -0.01 & 1.03 & -0.01 & -0.02 \\ -0.01 & -0.02 & 1.03 & -0.02 \\ -0.01 & -0.02 & -0.01 & 1.035 \end{bmatrix} \quad (2.3)$$

$$RGA_{U_0 \geq 50\%} = \begin{bmatrix} 1.02 & -0.01 & -0.01 & -0.01 \\ -0.01 & 1.04 & -0.02 & -0.02 \\ -0.01 & -0.02 & 1.04 & -0.02 \\ -0.01 & -0.02 & -0.02 & 1.05 \end{bmatrix} \quad (2.4)$$

From the RGA matrix, it is clear that the input-output pairs must be established as y_i-u_i , with $i = 0, 1, 3, 4, 5$. As expected, the water flow rate in each loop must be controlled with the pumping system of that loop. Although the interaction degree in this case is relatively low, with the aim of reducing it as much as possible, a decoupling net [3] has been included in the controller treated in the next chapter. [4, 5]

2.2.7 Validation

The model presented in Chap. 2.2 was tested to check its accuracy. The input signal used in the real plant was applied to the Simulink model and the results were compared.

In Fig. 2.45 are presented the validation experiment results. It is possible to see some similarities and some inequalities. In particular, the dynamics of the model follow the real dynamics of the plant. The poles and the zeros are placed properly and, due to this, the model behaviour follows the real one. As it could be expected, the main difference is in the gain. However, when the input signals achieve values close to the range limits some differences appear. The high error shows up as an off-set (< 2 l/min) when the direct transfer function works with the principal pump input at 20% (the lower extreme of the operating range). This error is acceptable since the goal is to control the system with PID and the usual operation point is higher. It can be said that the scheduling approximation implemented to simplify the non-linear gain is adequate.

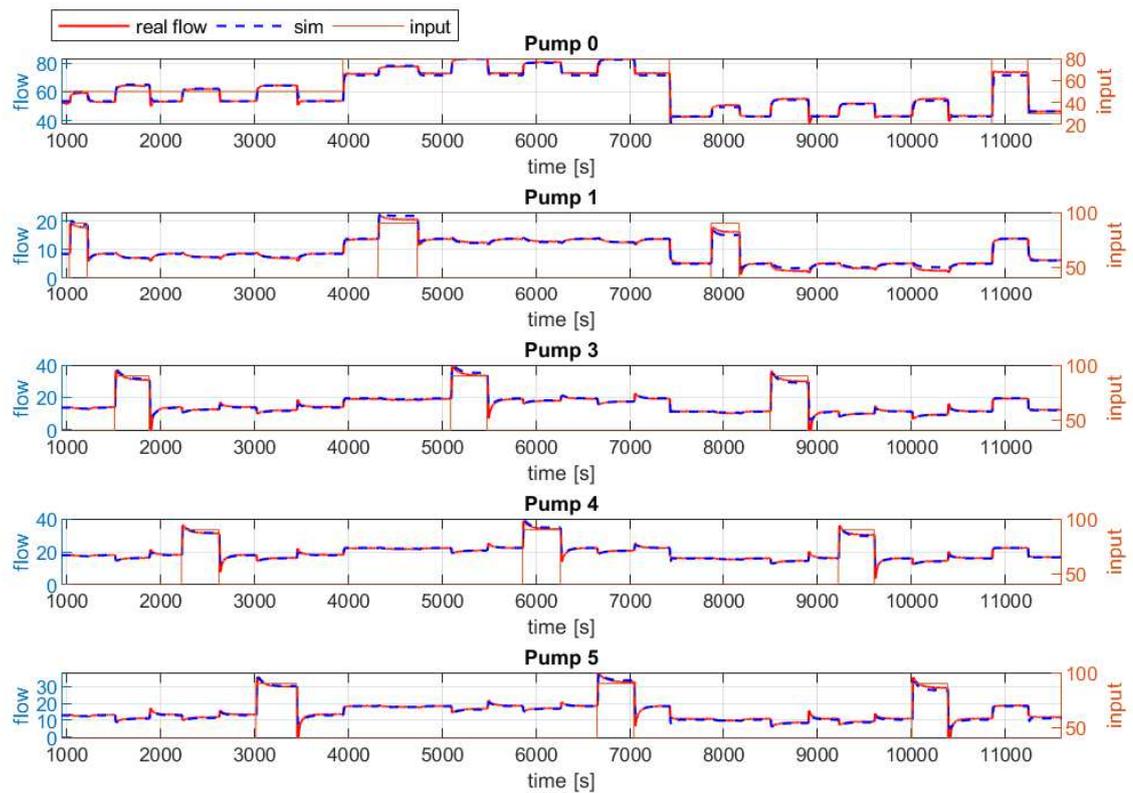


Figure 2.45: Gain scheduling approach - Model validation, comparison between real and simulated data

In Fig. 2.46 is shown the error introduced in each second for each loop. It is calculated as $e = y_{sim} - y_{real}$ where y_{sim} is the model output and y_{real} is the real output of the plant. The maximum error peaks occur at the input changes. Their value is quite high but short time. This simulation examines almost the entire operating range, so it makes sense to calculate the error average as Eqn. 2.5. Its value is shown below.

$$\bar{e} = \frac{|y_{sim} - y_{real}|}{n} \quad (2.5)$$

- Loop 0: $\bar{e}_0 = 0.7676$ l/min
- Loop 1: $\bar{e}_1 = 0.3730$ l/min

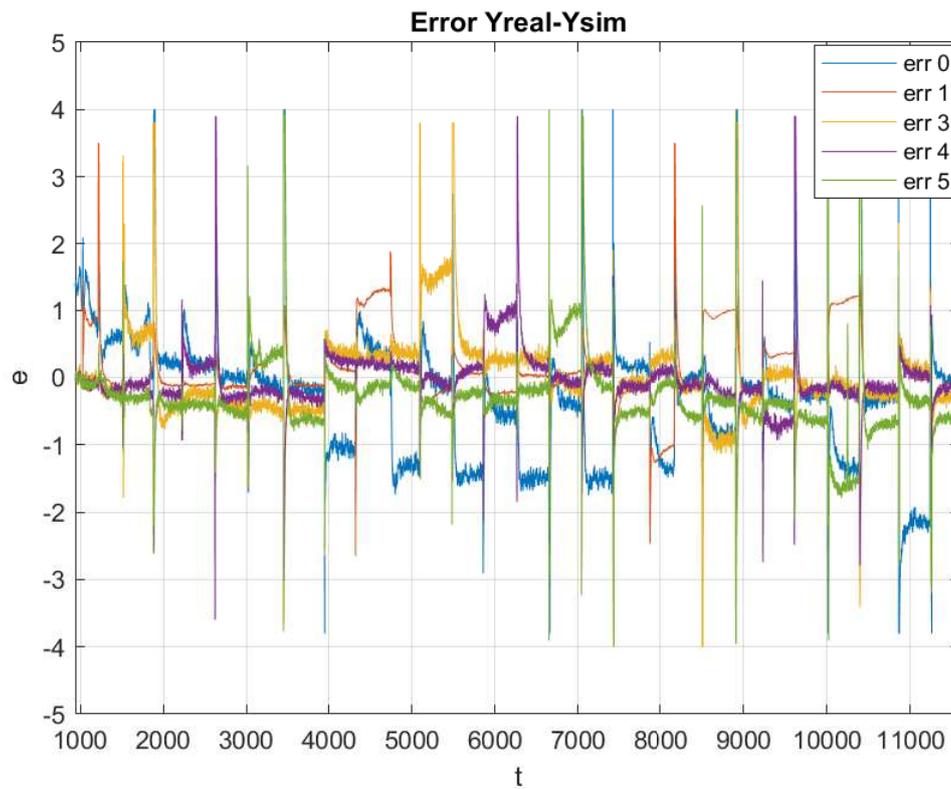


Figure 2.46: Gain scheduling approach - Model error

- Loop 3: $\bar{e}_3 = 0.4931$ l/min
- Loop 4: $\bar{e}_4 = 0.3030$ l/min
- Loop 5: $\bar{e}_5 = 0.5411$ l/min

2.3 Flow model 2 - non-linear gain

2.3.1 Introduction

The data presented in the previous chapter was used to develop another model for the hydraulic system. The gain scheduling approach presented in the Cap. 2.2 is one possible way to approximate the real system behavior. That approach presents two different situations and the transition happens steeply. The model presented in this section is a non-linear model. It was developed to have an accurate model without sharp transition. As what it was said before, the problematic part is in the gain and it was exploit to obtain the non-linear gain function. In order to model this kind of gain and to maintain the use of transfer functions, the strategy described below was used:

1. plotting the experimental loop pump gains according to the pump 0 input signal,
2. interpolating the figure found with a polynomial,
3. spreading the transfer function in two parts: one is a polynomial g_{ij} , which defines the gain and the other is a transfer function G_{ij} , witch describe the dynamics.

In this way we can treat the dynamics in an easy way and shift the non-linearity in the gain described by the polynomial. In the previous model the indirect transfer functions are kept constant, but in this chapter, with this technique it was also designed the indirect transfer function in order to obtain a higher accuracy in the model. Each transfer function G_{ij} has a unitary gain because the signal amplitude is imposed by the polynomial function g_{ij} . It is very important to take into account this point in the controller tuning. As it was done in the previous model, the flow in each pipe is the resulting sum of all the pump effects (direct transfer function and indirects). The water flow rate in each loop, y_i , where $i=0$ is for the principal pump and $i=1,3,4,5$ are for the loops, is therefore calculated as Eqn. 2.6

$$y_i = \sum_j y_{ij}, \quad i, j = 0, 1, 3, 4, 5, \quad (2.6)$$

where y_{ij} is the contribution to y_i due to the interaction between the loop i and the loop j , and they can be obtained from Eqn. 2.7

$$\frac{Y_{ij}(s)}{U_{0j}(s)} = g_{ij}G_{ij}, \quad g_{ij} = f(u_0, u_j), \quad (2.7)$$

being g_{ij} the polynomial that defines the gain and that depends on the inputs of the main, u_0 , and the loop j , u_j , pumps.

The data collected to find out the model transfer function was treated with the Ident Matlab toolbox. As it was already said, the steps don't start from zero, then in order to find the correct transfer function it was necessary to remove the output and the input initial value.

2.3.2 Transfer function

In this section are presented the polynomials and the transfer functions found for this model. With the nomenclature G_{ij} for $i,j=1,3,4,5$ are denoted the transfer functions (dynamics part), g_{ij} for $i,j=1,3,4,5$ it is the polynomial part (gain) and y_{ij} for $i,j=1,3,4,5$ defines the output signal.

Pump 0

$$G_{00}(s) = \frac{0.2464}{s + 0.2464} e^{-4s} \quad y_{00} = 0.0048u_0^2 + 0.0073u_0 + 41.5812 \quad (2.8)$$

$$G_{10} = \frac{0.1351}{s + 0.1351} e^{-4s} \quad y_{10} = 0.007u_0^2 + 0.068u_0 + 3.58 \quad (2.9)$$

$$G_{30} = \frac{0.4306}{s + 0.4306} e^{-5s} \quad y_{30} = 0.0011u_0^2 + 0.0171u_0 + 10.5324 \quad (2.10)$$

$$G_{40} = \frac{0.4306}{s + 0.4306} e^{-5s} \quad y_{40} = 0.001u_0^2 + 0.0031u_0 + 15.73 \quad (2.11)$$

$$G_{50} = \frac{0.3748}{s + 0.3748} e^{-5s} \quad y_{50} = 0.011u_0^2 + 0.0216u_0 + 9.9693 \quad (2.12)$$

Pump 1

$$G_{01} = \frac{0.08997}{s^2 + 1.02s + 0.08997} e^{-4s} \quad (2.13)$$

$$g_{01} = 8 \times 10^{-6}u_0^2 - 0.0018u_0 + 0.1938 \quad y_{01} = g_{10}u_1 \quad (2.14)$$

$$G_{11} = \frac{0.2946s + 0.006409}{s^2 + 0.2628s + 0.006409} e^{-6s} \quad (2.15)$$

$$g_{11} = -0.0014u_0 + 0.2473 \quad y_{11} = g_{11}u_1 \quad (2.16)$$

$$G_{31} = \frac{-0.2811s + 0.008885}{s^2 + 0.4214s + 0.008885} e^{-8s} \quad (2.17)$$

$$g_{31} = -5.7 \times 10^{-6}u_0^2 + 0.0007u_0 - 0.0325 \quad y_{31} = g_{31}u_1 \quad (2.18)$$

$$G_{41} = \frac{-0.2008s + 0.007278}{s^2 + 0.4699s + 0.007278} e^{-8s} \quad (2.19)$$

$$g_{41} = -5.3 \times 10^{-6}u_0^2 + 0.0006u_0 - 0.0260 \quad y_{41} = g_{41}u_1 \quad (2.20)$$

$$G_{51} = \frac{-0.4016s + 0.01599}{s^2 + 0.4264s + 0.01599} e^{-8s} \quad (2.21)$$

$$g_{51} = -6.2 \times 10^{-6} u_0^2 - 0.0007 u_0 - 0.0316 \quad y_{51} = g_{51} u_1 \quad (2.22)$$

Pump 3

$$G_{03} = \frac{2619}{s^2 + 2.675 \times 10^4 s + 2619} e^{-4s} \quad (2.23)$$

$$g_{03} = -4 \times 10^{-6} u_0^2 - 0.0003 u_0 + 0.2521 \quad y_{03} = g_{03} u_3 \quad (2.24)$$

$$G_{13} = \frac{-0.101s + 0.005787}{s^2 + 0.2839s + 0.005787} e^{-5s} \quad (2.25)$$

$$g_{13} = -8.2 \times 10^{-6} u_0^2 + 0.0013 u_0 - 0.0762 \quad y_{13} = g_{13} u_3 \quad (2.26)$$

$$G_{33} = \frac{0.2895s + 0.00306}{s^2 + 0.2189s + 0.00306} e^{-5s} \quad (2.27)$$

$$g_{33} = -0.0014 u_0 + 0.4028 \quad y_{33} = g_{33} u_3 \quad (2.28)$$

$$G_{43} = \frac{0.3592s + 0.002649}{s^2 + 0.175s + 0.002649} e^{-4s} \quad (2.29)$$

$$g_{43} = 5.6 \times 10^{-6} u_0^2 - 0.0005 u_0 - 0.0213 \quad y_{43} = g_{43} u_3 \quad (2.30)$$

$$G_{53} = \frac{0.9557s + 0.006807}{s^2 + 0.4268s + 0.006807} e^{-8s} \quad (2.31)$$

$$g_{53} = 5.6 \times 10^{-7} u_0^2 + 0.0002 u_0 - 0.0476 \quad y_{53} = g_{53} u_3 \quad (2.32)$$

Pump 4

$$G_{04} = \frac{3334}{s^2 + 2.724 \times 10^4 s + 3334} e^{-4s} \quad (2.33)$$

$$g_{04} = -6.8 \times 10^{-6} u_0^2 + 0.0003 u_0 + 0.1743 \quad y_{04} = g_{04} u_4 \quad (2.34)$$

$$G_{14} = \frac{-0.244s + 0.01141}{s^2 + 0.588s + 0.01141} e^{-5s} \quad (2.35)$$

$$g_{14} = -2.6 \times 10^{-6} u_0^2 + 0.0005 u_0 - 0.0444 \quad y_{14} = g_{14} u_4 \quad (2.36)$$

$$G_{34} = \frac{0.9392s + 0.004963}{s^2 + 0.3571s + 0.004963} e^{-4s} \quad (2.37)$$

$$g_{34} = 1.3 \times 10^{-6} u_0^2 - 3.3750 \times 10^{-5} u_0 - 0.0280 \quad y_{34} = g_{34} u_3 \quad (2.38)$$

$$G_{44} = \frac{0.4598s + 0.005299}{s^2 + 0.3321s + 0.005299} e^{-4s} \quad (2.39)$$

$$g_{44} = -0.0008 u_0 + 0.2984 \quad y_{44} = g_{44} u_4 \quad (2.40)$$

$$G_{54} = \frac{0.795s + 0.005769}{s^2 + 0.35s + 0.005769} e^{-4s} \quad (2.41)$$

$$g_{54} = -4.9 \times 10^{-7} u_0^2 + 0.0001 u_0 - 0.0377 \quad y_{54} = g_{54} u_4 \quad (2.42)$$

Pump 5

$$G_{05} = \frac{0.04489}{s^2 + 0.5968s + 0.04489} e^{-2s} \quad (2.43)$$

$$g_{05} = -3.7 \times 10^{-6}u_0^2 - 0.0003u_0 + 0.2469 \quad y_{05} = g_{05}u_5 \quad (2.44)$$

$$G_{15} = \frac{-0.1333s + 0.005995}{s^2 + 0.3164s + 0.005995}e^{-3s} \quad (2.45)$$

$$g_{15} = -5.8 \times 10^{-6}u_0^2 + 0.001u_0 - 0.07 \quad y_{15} = g_{15}u_5 \quad (2.46)$$

$$G_{35} = \frac{0.4043s + 0.003194}{s^2 + 0.1939s + 0.003194}e^{-4s} \quad (2.47)$$

$$g_{35} = 2.8 \times 10^{-6}u_0^2 - 0.0001u_0 - 0.0416 \quad y_{35} = g_{35}u_5 \quad (2.48)$$

$$G_{45} = \frac{0.5236s + 0.003655}{s^2 + 0.261s + 0.003655}e^{-5s} \quad (2.49)$$

$$g_{45} = 3.2 \times 10^{-6}u_0^2 - 0.0003u_0 - 0.0297 \quad y_{45} = g_{45}u_5 \quad (2.50)$$

$$G_{55} = \frac{0.31s + 0.003719}{s^2 + 0.2357s + 0.003719}e^{-5s} \quad (2.51)$$

$$g_{55} = -0.0014u_0 + 0.3951 \quad y_{55} = g_{55}u_5 \quad (2.52)$$

The general model scheme is the same as it was presented in Cap. 2.2 Fig. 2.38. The details are shown below.

2.3.3 Model scheme

Principal Pump To model the principal pump transfer functions (pump 0) it is not necessary to introduce an additional input signal because the gain changes according to

the input U_0 . In this case it is possible to model this gain as a second degree polynomial. The same things are applicable for the model's gain between the principal pump and each loop (indirect transfer function).

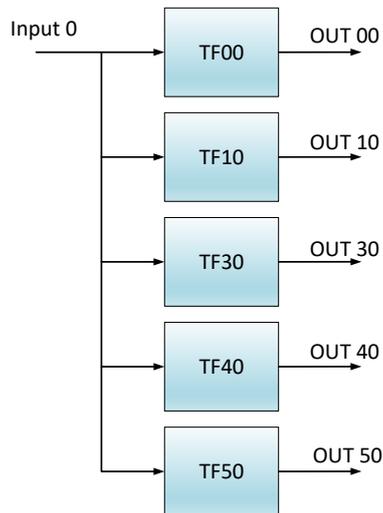


Figure 2.47: Non-linear model, principal pump blocks

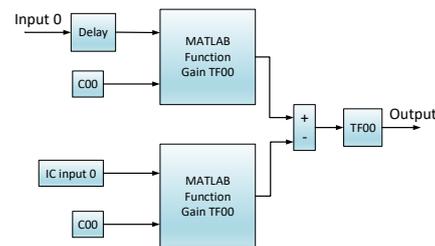


Figure 2.48: Non-linear model, transfer function 00

In Fig. 2.47 it is shown the principal pump blocks structure. Particular attention must be paid to the figure on the right (Fig. 2.48). The input signal follows the following path:

1. The input signal arrive in the transfer function blocks
2. A delay block translate the signal back in time
3. In the *Matlab Function Block* the polynomial which describes the gain is implemented. This step modifies the signal amplitude
4. The initial conditions are subtracted
5. The transfer function blocks ensure to give the desired dynamics to the signal.
6. The signal produced in the previous steps is placed in the output.

Significantly, the initial condition signal passes through the *Matlab Function Block* in order to modify its amplitude before being subtracted.

Loop Pumps These models are quite similar to the one shown above, but in the loops the equations (G_{ij} , being $i = j$) are first order for the gain in the direct transfer functions, while they are second order for the indirect ones (G_{ij} , being $i \neq j$). Already, from the Fig. 2.49 it is possible to note some difference compared to the principal pump scheme shown in Fig. 2.47. As input signal, it is necessary to introduced not only the loop pumps input but also the main pump operation point. This is necessary in the *Matlab Function Blocks* to calculate the transfer function gain g_{ij} (see Fig. 2.50). The scheme represented in Fig. 2.50 is the same for all the transfer function blocks of each loop (direct and indirect). The only thing that changes is the value of the constants in the polynomial and the position of the pole and the zero in the transfer function.

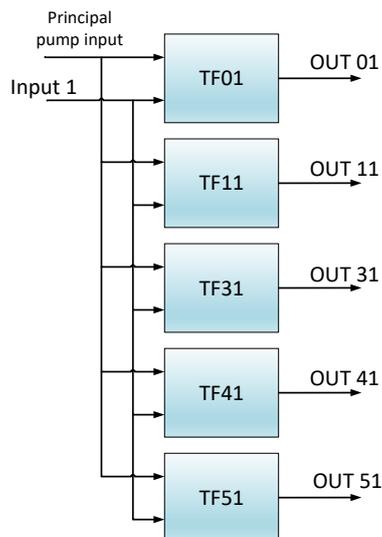


Figure 2.49: Non-linear model, loops pump blocks

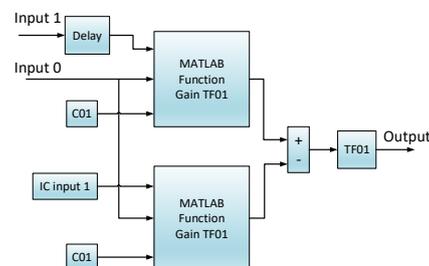


Figure 2.50: Non-linear model, transfer function 01

2.3.4 RGA matrix

In this case, due to the non-linearities, the RGA matrix is not constant, it depends on the operating point. Based on the experience at the facility, the principal pump uses to work at more than 50% of its speed. In order to follow as much as possible the real operation point, the RGA matrix was calculated when the pump 0 input is at 80%, obtaining the following one:

$$RGA_{80\%} = \begin{bmatrix} 81.35 & -17.2 & -20.8 & -21.8 & -20.6 \\ -15.0 & 23.6 & -2.5 & -2.5 & -2.6 \\ -20.9 & -1.9 & 30.2 & -3.1 & -3.3 \\ -24.0 & -1.6 & -2.8 & 32.5 & -3.2 \\ -20.5 & -1.9 & -3.2 & -4.1 & 30.6 \end{bmatrix} \quad (2.53)$$

$$RGA_{80\%} = \begin{bmatrix} 1.03 & -0.01 & -0.01 & -0.01 \\ 0.01 & 1.04 & -0.01 & -0.02 \\ -0.01 & -0.01 & 1.03 & -0.02 \\ -0.01 & -0.02 & -0.02 & 1.04 \end{bmatrix} \quad (2.54)$$

In Eqn. 2.53 it is shown the RGA considering the principal pump. In Eqn. 2.54 it is not considered (controlled in open loop). The input-ouput pairs must be established as y_i-u_i in this case too. Already, it is logical because the physical system is the same.

2.3.5 Validation

The model presented in this section was tested and compared with real data. The experimental data, used for this validation, are the same used for the validation of the model in Cap. 2.2.7. This allows us to make a comparison between the two models.

In Fig. 2.51 it is presented the comparison between the real plant data and the simulation. As it can be seen, the two signals are similar. So, with the same input signals, the model implemented in this section provides an output akin to the real hydraulic system.

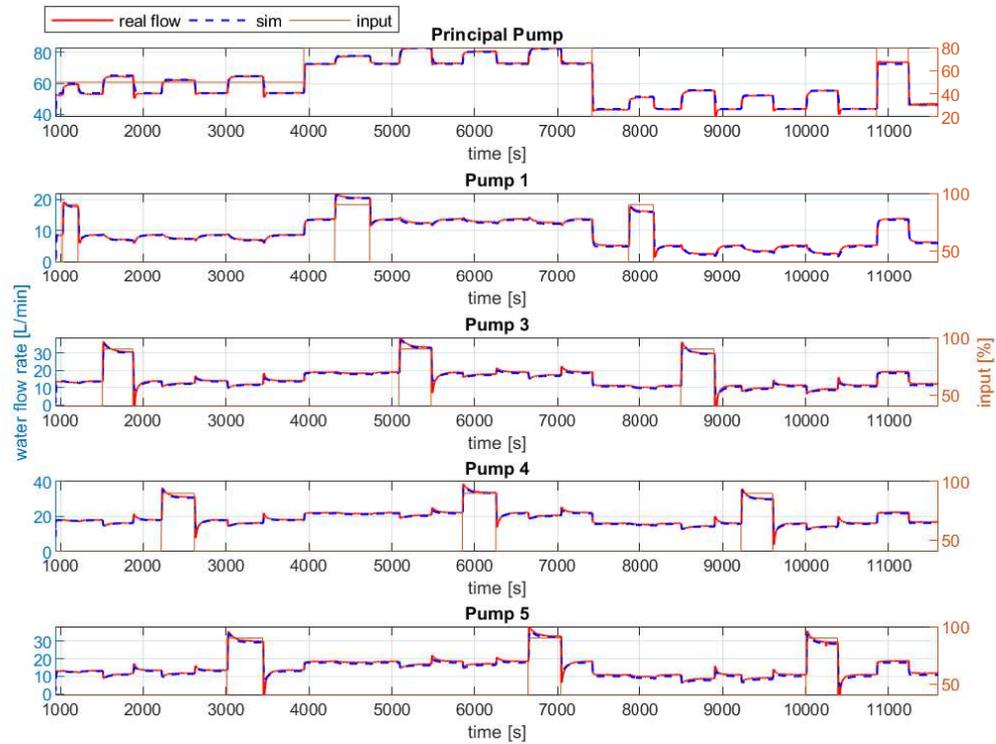


Figure 2.51: Non-linear model - validation, comparison between real and simulated data

In Fig. 2.52, the error along the whole simulation is shown. It is calculated as $e = y_{sim} - y_{real}$ where y_{sim} is the model output and y_{real} is the real output of the plant. The highest values can be observed at the input changes. These peaks are short-term and they cancel before the end of the transient. In stationary situation, the error remains almost constant. The average error occurring during the simulation is shown below:

- Loop 0: $\bar{e}_0 = 0.6826$ l/min
- Loop 1: $\bar{e}_1 = 0.3711$ l/min
- Loop 3: $\bar{e}_3 = 0.6841$ l/min
- Loop 4: $\bar{e}_4 = 0.4754$ l/min
- Loop 5: $\bar{e}_5 = 0.8229$ l/min

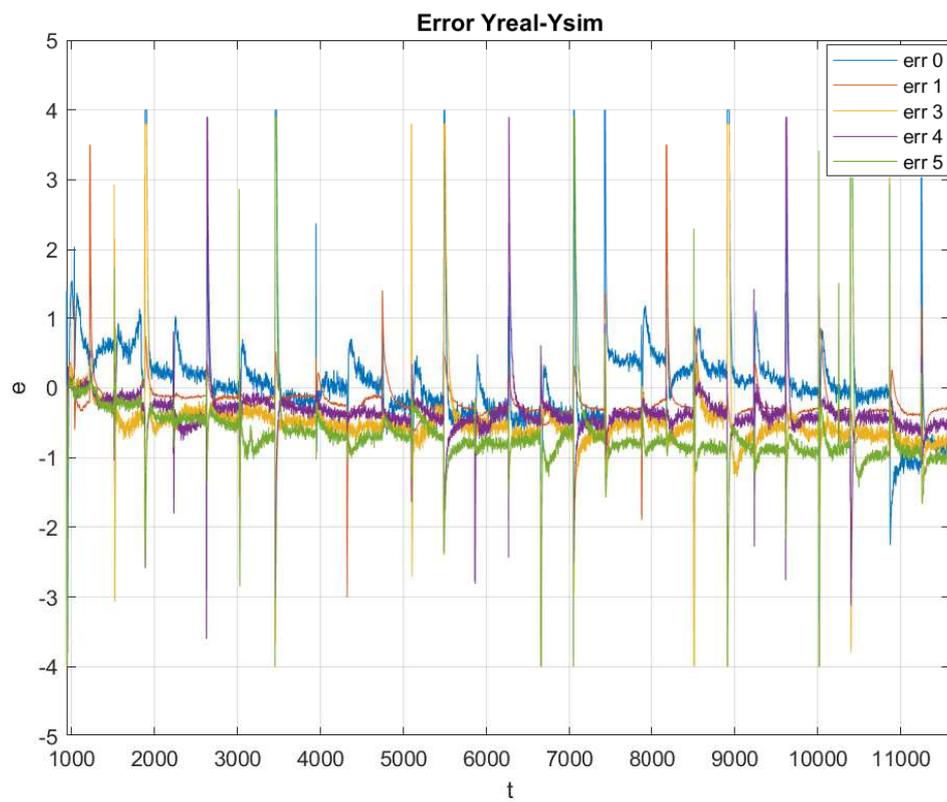


Figure 2.52: Non-linear model - error

2.4 Temperature model

In order to control the outlet temperature of each loop, it is necessary to use a model.

The model used in this thesis is presented into the paper *Modeling and simulation of a solar field based on flat-plate collectors* [6]. For more details it's possible to make reference to it. The idea is to have a model between flow (input) and temperature (output) of each loop. The outlet temperature of each loops is related to the flow but there are also other influential physical quantities. The model used must include them. In particular, in the plant are included a solar field, an air cooler to reduced the water temperature in the solar field and an heat exchanger to heat the water in the tanks. The connection between this equipment is shown in Fig. 1.6.

The equation used in this model concerns energy and mass balance. In particular the equation that describes the evolution of the outlet temperature is presented in Eqn. 2.55 and Eqn. 2.56

$$\rho c_p A_{cs} \frac{\partial T_{loopj,out}(t)}{\partial t} = \beta I(t - d_{j,tout-I}) - \frac{H}{L_{eq}} (\hat{T}(t) - T_a(t)) - c_p \frac{\rho}{c_f} Q_{loopj}(t - d_{j,tout-Q}) \frac{T_{loopj,out}(t) - T_{loopj,in}(t - d_{j,tout-tin})}{L_{eq}} \quad (2.55)$$

$$\hat{T}(t) = \frac{T_{loop,out}(t) + T_{loop,in}(t - d_{j,tout-tin})}{2} \quad (2.56)$$

where:

- ρ water density, 975 kg/m^3
- c_p specific heat capacity, $4180 \text{ J} \cdot \text{kg}^{-1} \cdot \text{°C}^{-1}$
- A_{cs} flat plate collector tube's cross area, m^2
- L_{eq} length of the equivalent flat plate collector tube, 1.95 m

- H thermal losses coefficient, $1.1 \text{ J} \cdot \text{s}^{-1} \cdot ^\circ \text{C}^{-1}$
- β parameter that modulates the solar radiation, 0.0189 m
- \hat{T} mean temperature of the equivalent flat plate collector tube, $^\circ \text{C}$
- c_f conversion factor, 12×10^6

This model can be used only if the following constraints are satisfied, Eqn. 2.57 Eqn. 2.58, Eqn. 2.59.

$$T_{loopj,out} > T_{loopj,in} \quad (2.57)$$

$$Q_{loopj} > 0 \quad (2.58)$$

$$I > 0 \quad (2.59)$$

From Eqn. 2.55 it is clear that the outlet temperature depends on irradiation, ambient temperature, loop flow, inlet temperature. All of these signal, except for the flow, can be treated as external disturbances in the controller. This model is a concentrated parameter model, so in order to reproduced the real behaviour of the plant, it is necessary to introduced delay in the signal. This is really important since the model provide the outlet temperature of the solar field and the delays have got a significant value. The delays are found experimentally and their value are:

- $d_{j,tout-Q} = 33$
- $d_{j,tout-tin} = 0.14Q_{loopj}^2 - 21.4Q_{loopj} + 915.8$
- $d_{j,tout-I} = 36$
- $d_{1,tout-Q} = 29$

- $d_{1,tout-tin} = 1.6Q_{loop1}^2 - 59.4Q_{loop1} + 719.4$
- $d_{1,tout-I} = 45$

with $j = 3, 4, 5$.

In Fig. 2.53 the scheme of the model is shown. As can be seen, the contributions of the air cooler and the heat exchanger are included in T_{in} .

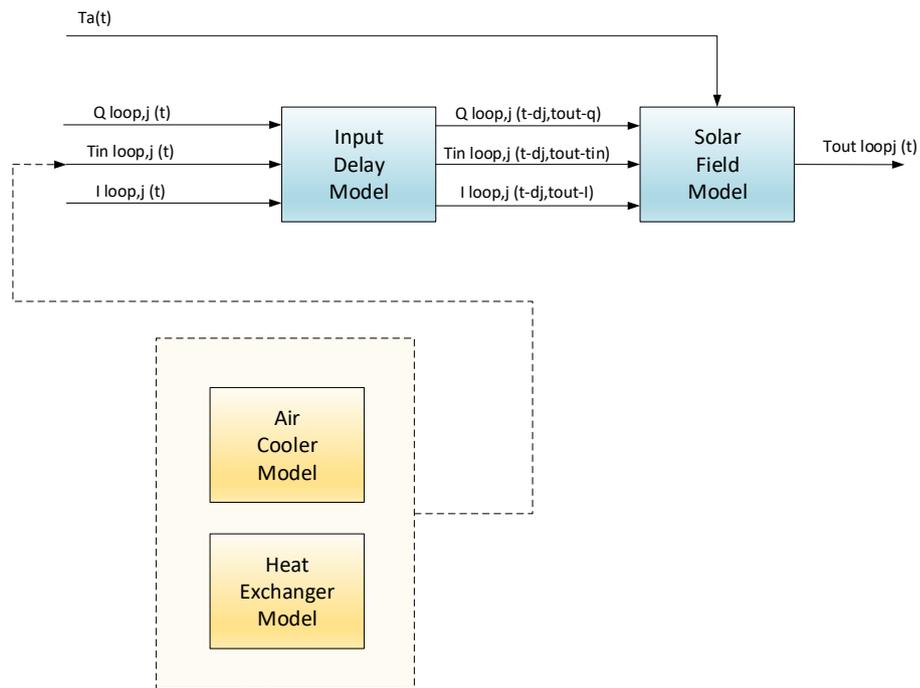


Figure 2.53: Temperature model scheme, controlled variable and load disturbance

Chapter 3

Flow controller

3.1 Introduction

In this chapter it is treated the flow control with the aim of obtaining the internal loop of a temperature *cascade control*. In this solar field, each pump of this system influences all the system's pipes and not only the loop where it is located. For this reason, a MIMO control system will be developed. The main purpose of the control is decoupling the loops in order to impose the required flow in each pipe. The flow control is faster than the temperature control and this is optimal since the goal is to control the temperature with a cascade control system. The time constants are equal to some seconds for the flow control and to a range of few minutes for the temperature control. Two different control systems are planned for the two different models presented in the Cap. 2. One exploits the *gain scheduling* idea, and one tries to modify the PIDs signals to compensate the non-linearity. These controllers are developed in order to be consistent with the two models.

3.2 Gain scheduling controller

In Cap. 2.2 it is shown the first idea for the model. The control structure presented in this chapter is planned to make the simulation with it. That model presents two situations

and the same scenario appears in the controller. The main idea is to change the PID's parameter according to the model. This controller presents two situations:

- situation 1: $U_0 < 50\%$
- situation 2: $U_0 \geq 50\%$

where U_0 is the principal pump control signal, and between these two situations, only the PID's parameter changes. To obtain a workable controller the following blocks are included:

- PIDs: to obtain the control signal
- Decouples: to make the interactions less relevant
- Anti windup: to solve the problems caused by the saturations
- Bump less structure: to solve the problems caused by the switch

All of this blocks will be treated in detail in the next section of this chapter.

3.2.1 Controller scheme

In Fig. 3.1 the Simulink control structure is presented, and the blocks mentioned before are placed as it is shown. Here, it is possible to see that we are in the presence of a MIMO control system. As input there are five set-points and four feedback signals (the principal pump is controlled in open loop), and as output there are five control signals. It can therefore be defined a *Multiple Input Multiple Output* controller 5×5 . In this diagram it can be seen that there are two distinct blocks called 'controller'. Only the output of one of these will constitute the real exit signal, so it is necessary to install an appropriate switch system. In the Anti-windup and Bumpless blocks the tracking time constant changes, so it is necessary to provide one different block for each controller part.

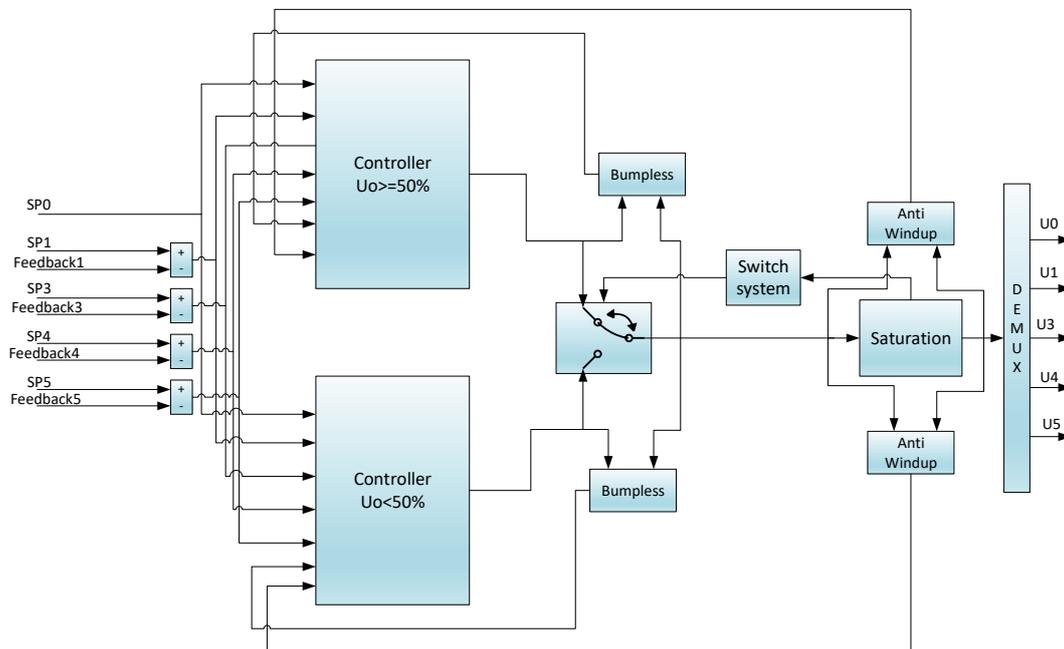


Figure 3.1: Gain scheduling controller - Simulink scheme

Controller blocks

In Fig. 3.2 the contents of the Controller block are highlighted. It is within these blocks that the control signal is produced. The various loops are controlled in closed loop while the main pump is controlled in open loop.

Principal pump The *principal pump* does not directly control the flow in the loops but it is necessary to support the other pumps. For this reason it is controlled in *open loop*. The set-point is defined as the sum of the set-points of each individual loop, $SP_0 = SP_1 + SP_3 + SP_4 + SP_5$. This makes sense because the sum of the loop set-points will constitute the flow that passes through the main tube. This value gives us the information that allows us to understand how much the main pump has to work. Since an open loop control is required to produce the control signal, a *look-up table* is used. The equation

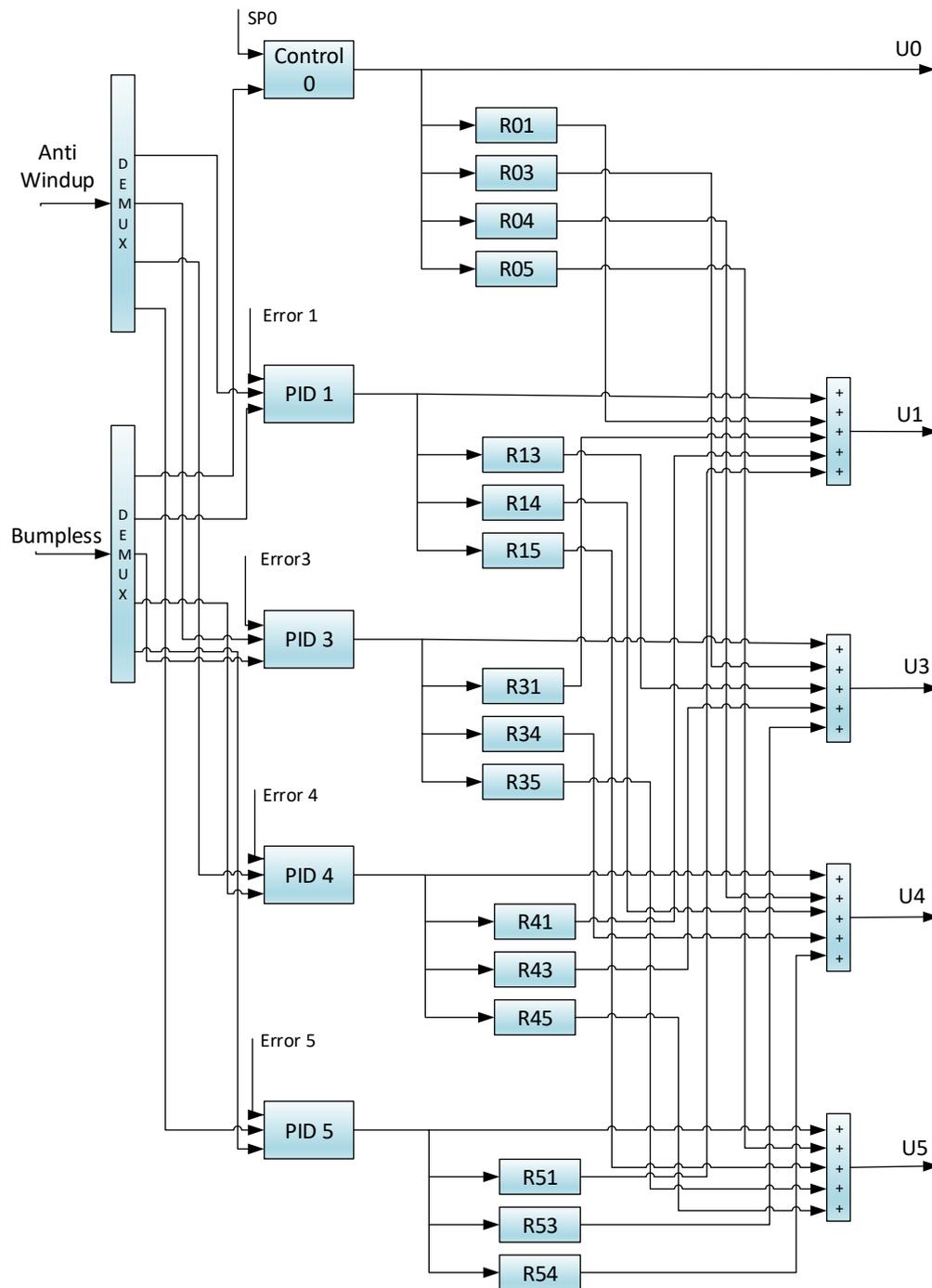


Figure 3.2: Controller Blocks, relative to Fig. 3.1

implemented in it is shown in Eqn.3.1

$$U'_0 = 0.8333 \times (SP_0) \quad (3.1)$$

The control signal passes from 0% when the SP is also at 0, to 100% when the SP is set to *maximum flow* in a linear way. As it can be deduced from the equation, the maximum flow is set equal to 120 l/min. To smooth the control signal during a set-point change, a filter was placed, Eqn. 3.2

$$TF_{filter} = \frac{0.09}{s + 0.09} \quad (3.2)$$

Loops pump All the pumps on the loop are controlled with the same control scheme and only the parameters change according to the model. In this case, the control is produced with a closed loop structure. The main element in this case is the PID. This block implements a PID in the noninteractive form. Due to the presence of positive zeros, the derivative action is switched off. The tuning was done in different ways and the result will be shown in the next chapter.

Decoupling In this sector, the functions are used for decoupling each loop are placed. These functions are not calculated in the ideal form because the transfer functions present delays and positive zeros. With an ideal decoupling these positive zeros become positive poles that destabilise the system. Simplified decoupling are calculated as Eq. 3.6, where the first number indicates the loop pump and the second the affected loop.

$$R_{01}(s) = -\frac{G_{01}(s)}{G_{11}(s)} \quad (3.3)$$

The effect of interaction can be reduced but not completely eliminated, due to these decouples. The signal produced by the decoupling is added to the signal produced by the PIDs and it is placed at the output of this control block.

Bumpless system

As it was said before, this control structure needs a bumpless system, in order to avoid steps on the control signal when the switch takes place.

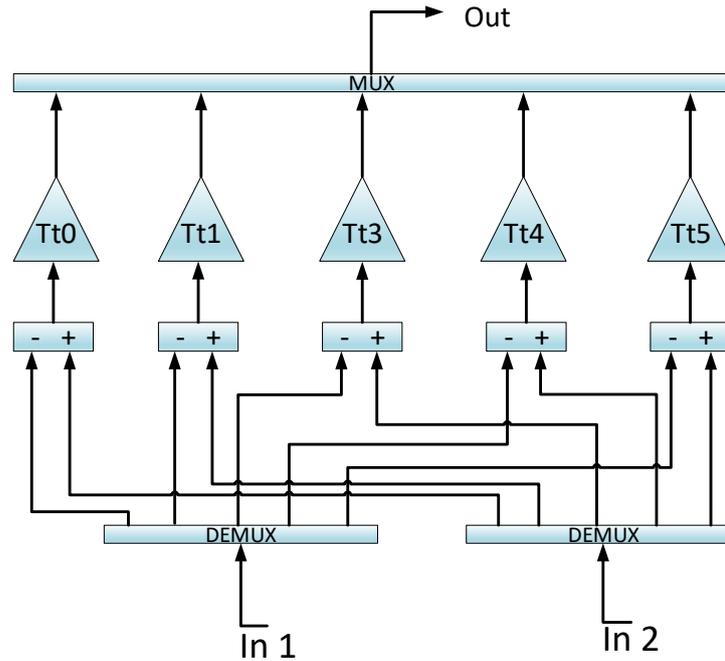


Figure 3.3: Bumpless blocks, relative to Fig. 3.1

In Fig. 3.3, the inside structure of the "Bumpless block" is shown. As input, the signals are taken before and after the switch system. In particular, the signal collected before with the minus sign and the signal collected after with the plus sign. In this way the sum will be zero for the part of the active output controller. The output signal is added to the integral action of the PIDs. In Fig. 3.3 it is possible to see five gain blocks. The gain values in these cases are called *Tracking Time Constants* and they are set equal to the integral time constant of the specific loops. So it is possible to establish this relationship, $Tt_i = Ti_i$, where Tt_i is the tracking time constant of the bumpless and Ti_i is the integral time constant of the PIDs. The subscript i indicates the loop involved.



Figure 3.4: Switch controller system, relative to Fig. 3.1

Switch system

In order to change the controller it is necessary to develop a system that does not give rise to ambiguous situations. This means that only one controller must be connected with the output port. The switch system is composed by one switch blocks and one switch controller block. This last block is composed as shown in Fig. 3.4.

The input signal is U_0 (principal pump control signal). This signal is produced by the controller itself so it's necessary to condition the signal in order to prevent ambiguous situation. First of all, the signal U_0 must be filtered to eliminate sudden changes. It would not make sense to change controllers every few instants. A relay block is placed after the filter. This allows to solve problems occurring when the signal U_0 is around the 50% (switch value). The threshold values are set to 45% and 55%. In this way the continuous changes are avoided and the error introduced in this way is negligible compared to the precision of the model. At the end, a delay of one second was placed in order to avoid a situation of initial indeterminacy (simulink problem).

Anti windup

In the controllers used maintain the integral action. It is consequently essential to use an anti windup structure in order to avoid the integral problem when the actuator saturates. The saturation limits of the control signal are set as follows:

- upper limit: 90%

- lower limit: 20%

These values were suggested by PSA staff and they were verified through an experimental test. The behavior of the pumps outside this range is strongly non-linear and controlling it would therefore be very difficult. The saturation blocks are placed directly inside the controller and the anti windup structure collects the signal before and after these. The tracking time constants are set equal to the PIs integral time constant. This structure is called *Back Calculation*. Note that, when the actuators are not saturated the anti windup structure has no effect on the output control signal.

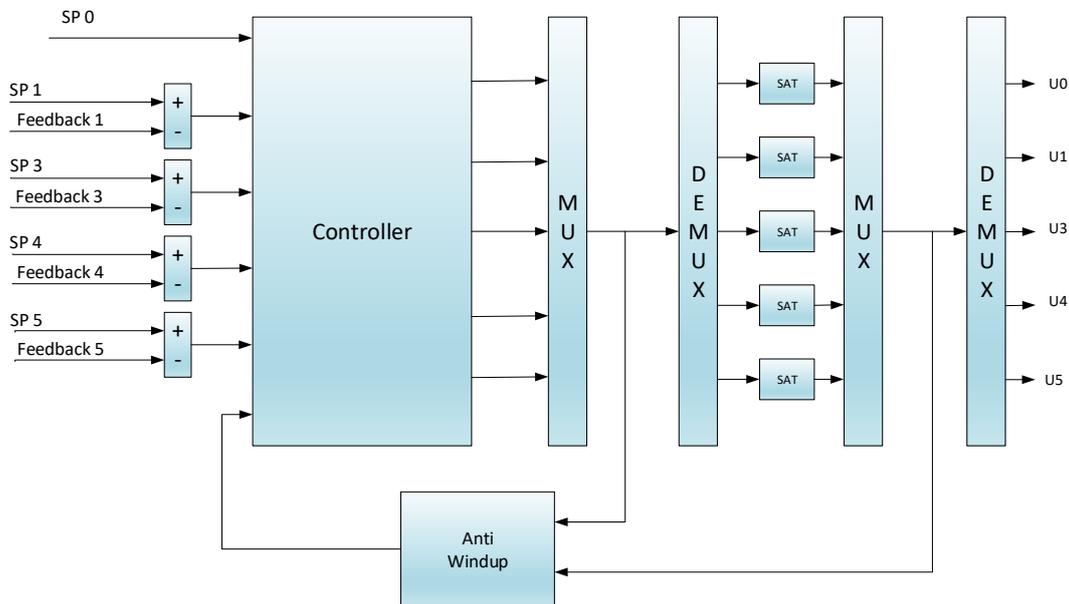


Figure 3.5: Non-linear controller - Simulink scheme

3.3 Non-linear controller

As it was described in Cap. 2.2, this hydraulic system is a non-linear system, so the MIMO controller includes some special features in order to deal with the non-linearities. For compensating these non-linearities, the inverse of the pump characteristics can be placed after the linear controller and before it is applied to the pump. With this idea, it is possible to obtain a considerable improvement in the performance of the close loop system.

In Fig. 3.5 the main controller signal are presented. Furthermore this time the controller is not divided into two parts. The controller will be analyzed in details below.

3.3.1 Controller scheme

In Fig. 3.6, the inside of the controller is shown.

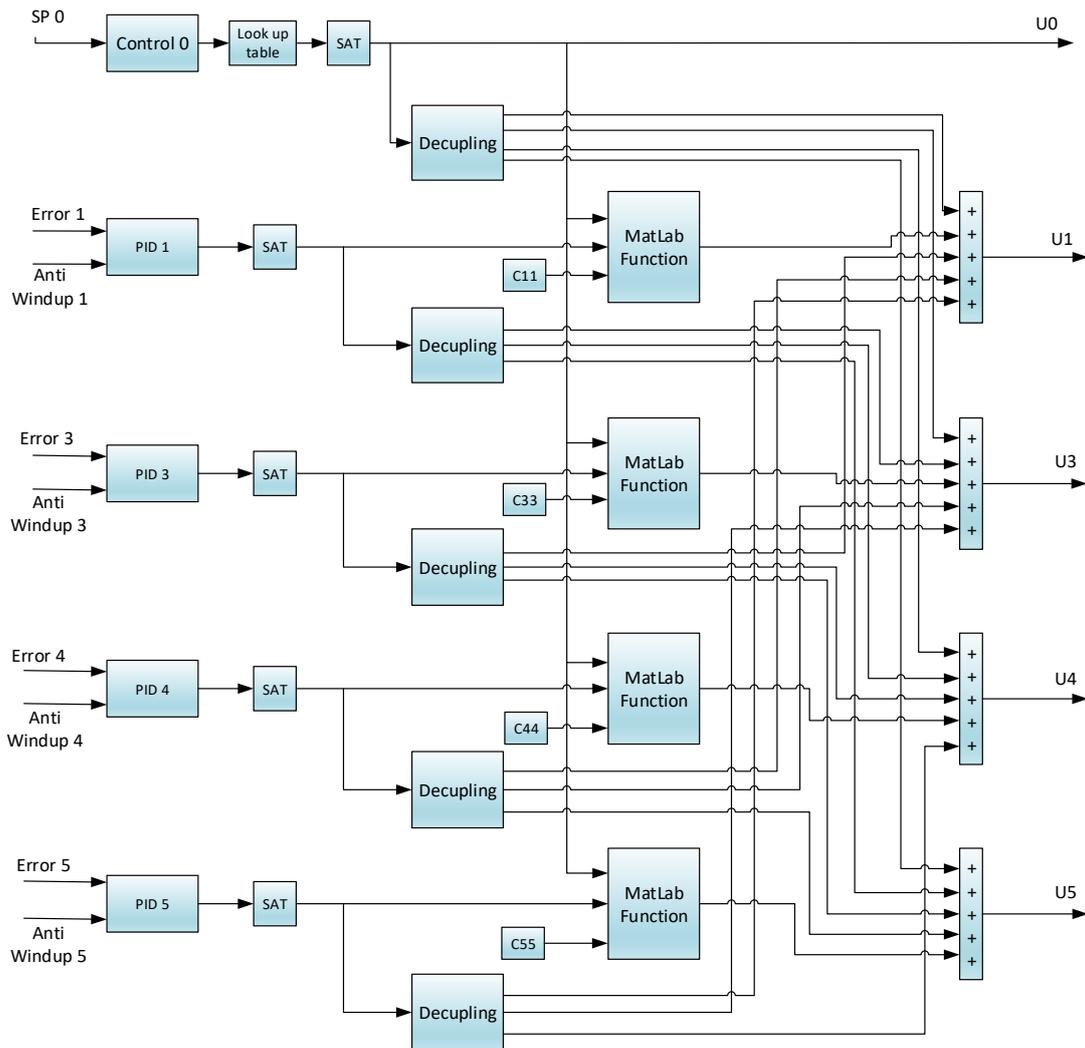


Figure 3.6: Non-linear controller - controller blocks relative to Fig. 3.5

Principal pump The principal pump does not directly control the flow in the loops but it is necessary to support the other pumps. For this reason it is controlled in *open loop*. In Fig. 3.6, the control scheme for the principal pump is shown. It is composed of three main blocks and the set-point is allocated as the sum of the set-points of each individual loop.

Control 0 In this block a linear function obtained experimentally (see Eq. 3.4) receives the set-point and provides a control signal. The control signal passes from 0% when the SP is also at 0, to 100% when the SP is set to *maximum flow* in a linear way. The maximum flow value is set to 120 l/min.

$$U'_0 = 0.8333 \times (SP_0) \quad (3.4)$$

In comparison with the controller in Cap. 3.2, the control strategy for the loop 0 is the same in these blocks.

Lookup Table In order to compensate the non-linearities of the system, a lookup table was used. It is implemented by a piecewise linear function (see Eq. 3.5) that approximates the inverse of the function which describes the gain of the model.

$$\begin{cases} U_0 = 2.91U'_0 - 106.34 & \text{if } U'_0 < 50\% \\ U_0 = 1.68U'_0 - 44.75 & \text{if } U'_0 \geq 50\% \end{cases} \quad (3.5)$$

Decoupling In this sector are placed the functions that are used for decoupling each loop. These functions are not calculated in the ideal form because the transfer functions present delays and positive zeros. These positive zeros, with an ideal decoupling become positive poles that destabilise the system. Simplified decouplings are calculated as Eq. 3.6, where the first number indicates the loop pump and the second the affected loop.

$$R_{01}(s) = -\frac{G_{01}(s)}{G_{11}(s)} \quad (3.6)$$

Loops pumps All the pumps on the loop are controlled with the same control scheme and only the parameters change according to the model.

PID block This block implements a PID in the non-interactive form. Due to the presence of positive zeros, the derivative action is switched off.

MATLAB Function This is the main difference between this control scheme and the typical MIMO control structure. The signal from PID passes through this block which modifies the signal to compensate the non-linearity. As it was said before, the no linearity gain of each loop is connected with the principal flow. As explained in [7], with this idea, it is possible to obtain a considerable improvement in the performance of the close-loop system. Inside of these blocks are contents these operations:

1. the value of the main flow is collected,
2. with that value the inverse of the model loop gain is calculated,
3. the signal coming from the PID is multiplied by the value obtained at the previous point and supplied at the output.

Due to the use of the *MATLAB Function* block, the gain of the system perceived by the PID in each moment is equal to one, then for the PID tuning, it is necessary to consider a unitary gain.

Chapter 4

Temperature controller

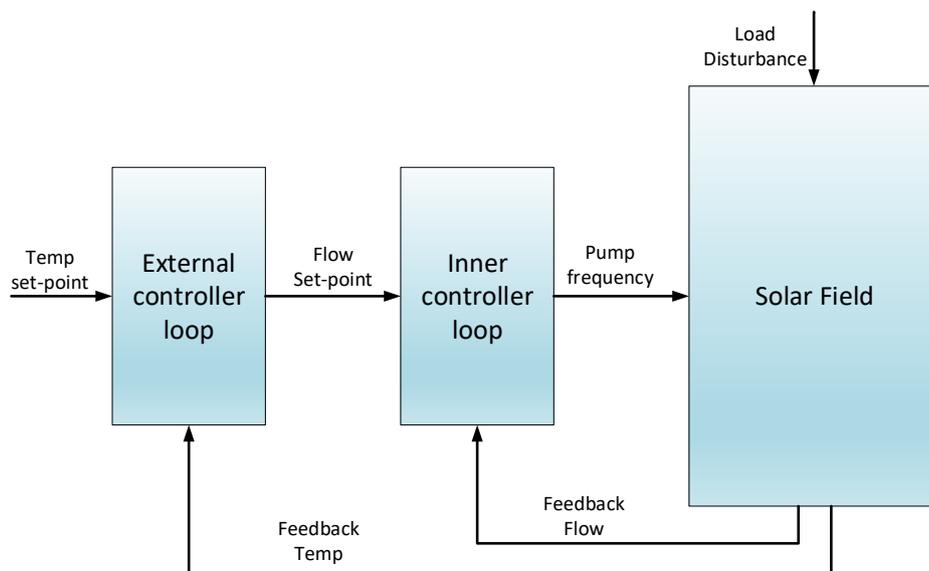


Figure 4.1: Cascade control scheme for temperature control

In the previous chapter the flow control was treated. As it was said in Chap. 2.4 the water flow affects the outlet temperature so the main idea is to use it as a controlled variable, in order to impose an outlet temperature in the solar panel [8]. It will be used a *cascade control*, which can be used since the system can be decomposed. In particular

the solar field model was decomposed in two parts, one describing the relation between temperature and flow and one describing the relation between pump and flow. The first was treated in Chap. 2.4 and the second in Chap. 2.3. To use this type of controller it is necessary to measure at least two input values (temperature and flow are mandatory) and with them, the controller provides the pump frequency signal. As it can be seen in the following section the controller developed in this thesis needs more input signals because it includes a feed forward control action and it is necessary to measure load disturbances.

The internal loop is composed by a flow controller (developed in the previous chapter) whose set-point value will be provided by the external loop.

In the following section the cascade control is treated in each part.

4.1 Flow controller - inner loop

The inner loop receives the flow set point signal produced by the external loop and it provides to set the pump frequency in order to get it. The flow controller treated in Chap. 3.3 is used here to constitute the inner loop. It is necessary to modify some parts in order to comply with the safety specifications and to make the flow controller compatible with the cascade controller. The main difference is in the anti-windup structure. The main goals in this case are:

- to connect the internal controller to the external controller without problems in the PID's integral part
- to avoid situation with a low level of pump's frequency.
- to increase the anti wind-up performance since the actuators will be in saturation a lot of time.

The control signal produced by this controller is the sum of PID actions and decouplers action, as Eqn. 4.1 (see Fig. 4.2):

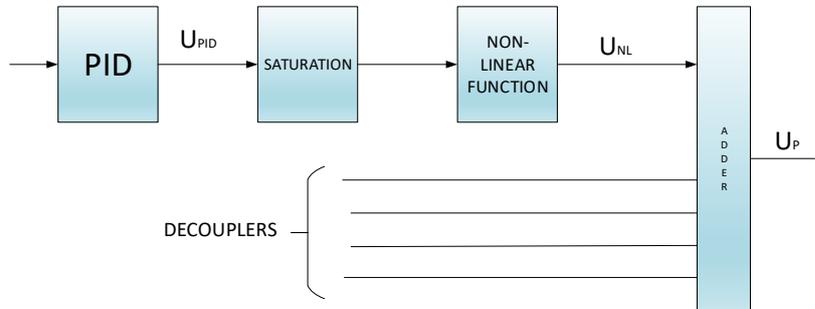


Figure 4.2: Controller signals, inner loops

$$U_P = \sum_{i=1}^N U_D + \frac{U_{PID}}{g} \quad (4.1)$$

where U_p is the controller output, U_D is the signal produced by decouplers, U_{PID} is the signal produced by PID and g is the gain produced by 'MATLAB Function' treated in Cap. 3.3. The limits for U_p now are set higher than what it was done in the previous flow controller. This choice was made because this controller will work all day long and its purpose is to control the temperature. With low levels of pump frequency not all the solar panel tubes will have the same flow and the same temperature. The limits value were chosen, according to what it was said by the PSA staff, equal to 40% lower limits and 90% upper limits.

- $U_{P-MAX} = 90\%$
- $U_{P-MIN} = 40\%$

In order to respect these limits, the Eqn. 4.2 and Eqn. 4.3 are implemented to determine

the PID's limits.

$$U_{PID-MAX} = (U_{P-MAX} - \sum_{i=1}^N U_D) \cdot g \quad (4.2)$$

$$U_{PID-MIN} = (U_{P-MIN} - \sum_{i=1}^N U_D) \cdot g \quad (4.3)$$

The PIDs limits are therefore not constant and it is necessary to calculate it at any moment. Their value is important to implement a correct anti windup.

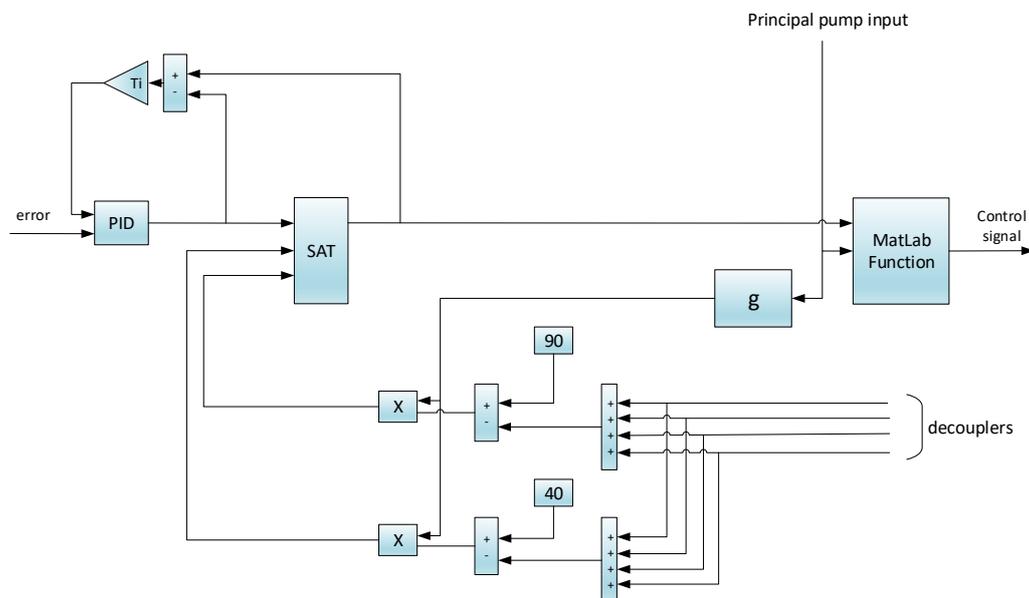


Figure 4.3: Anti-windup structure with variable limits

In Fig. 4.3 it is possible to see the main difference between the flow controller used as inner loop in this cascade control and the controller used in Cap. 3.3. The saturation blocks present three input. One is the PID output and the other to receive the value of the limits. An other added block is the 'g' block where is calculated the value of g .

Whit this technique the control signal produced by 'MatLab Function' can be added to the control signal produced by decoupling structure. This sum will never be outside the limits imposed. It turned out to be necessary to implement these limits because during the day, the pump will work in saturation for a lot of time and a typical anti-windup structure introduces a significant error, which is caused by 'MatLab function Block'.

4.2 Temperature controller - external loop

This external loop provides the flow set-point value for the inner loop. This controller works with PIDs. However, the PIDs alone would not be able to control load disturbances in a proper way so it is necessary to introduce a feed forward action. In Fig. 4.4 the general control scheme is shown. It is a MIMO controller with four inputs (loop temperature set point) and four outputs (desired flows). In this case decoupling structure are not present because the temperature in the loop j doesn't affect the temperature in the loop i , with $j, i = 1, 3, 4, 5$ and $j \neq i$. Actually, a small interaction is present because the outlet temperature in the loop j affects the inlet temperature in the loop i . However it is negligible and the inlet temperature effect is just considered in the feed forward action. As it was said for the internal loop, also the external loop needs a good anti windup system because the signal produced will be in saturation a lot of time during the day. This happens because in the morning the irradiation is low and the controller tries to reduce the flow to increase the outlet temperature.

Below, the various blocks are described.

Feed Forward action Since the load disturbances are present and their influence on the outlet temperature is strong, in order to increase the system performance it is necessary to use a feed forward action in the controller [9]. The FF controller permitted to prevent the error because it produced the action according to the load disturbance value. It doesn't wait for the error, between output and set point, as PIDs had done.

The Eqn. 4.4 was used to calculate the feedback control law. It is the energy balance

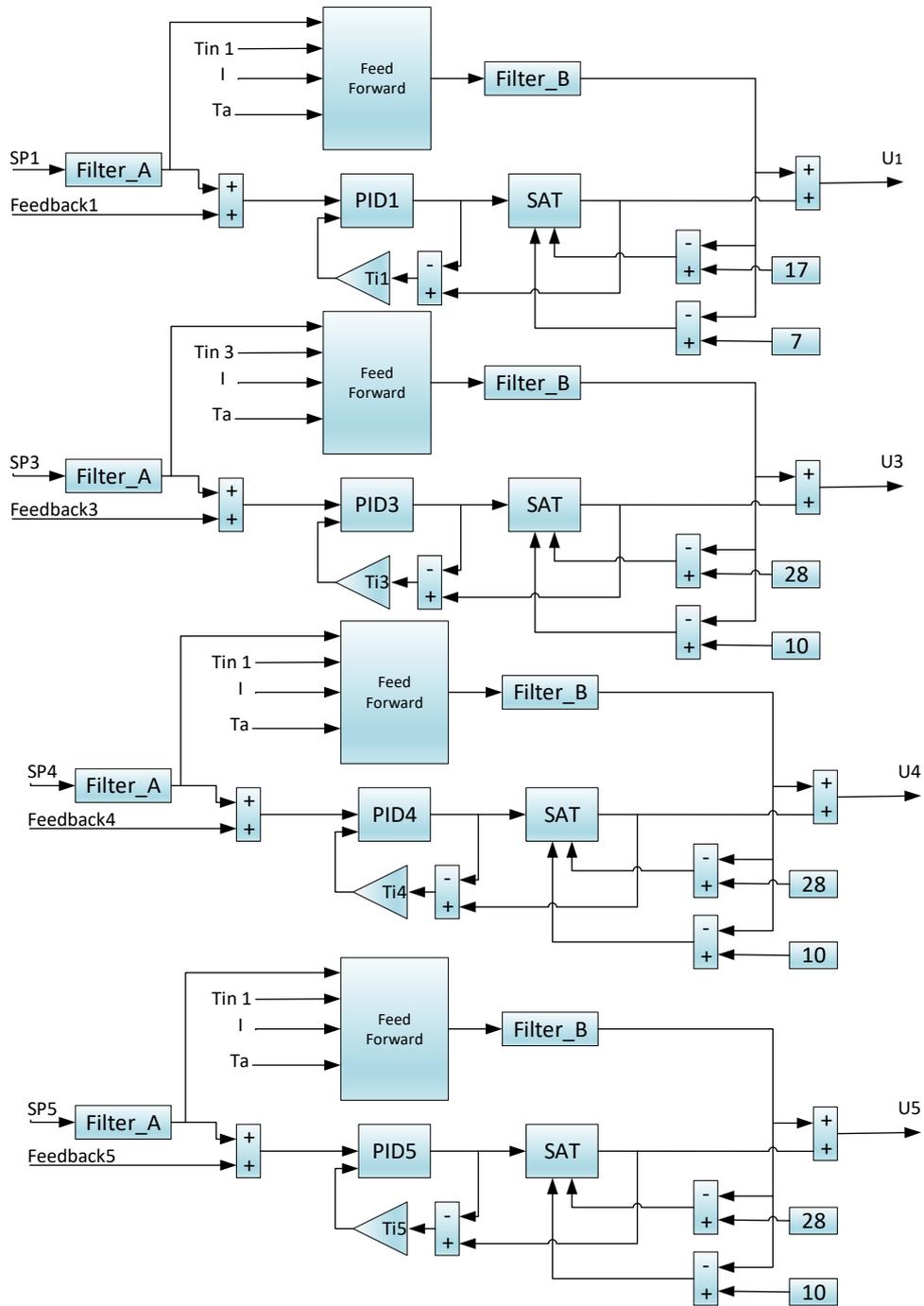


Figure 4.4: Temperature controller scheme: PIDs action, FF action and anti windup

equation and it describes how the temperature changes in the solar field.

$$A_{sf}\rho c_p \frac{\partial TT2(t)}{\partial t} = \beta I(t) - \frac{H}{L_{eq}}(\bar{T}(t) - T_a(t)) - c_p \dot{m}_{eq} \frac{TT2(t) - TT1(t)}{L_{eq}} \quad (4.4)$$

$$\dot{m}_{eq} = \frac{FT1\rho}{C_f} \quad (4.5)$$

$$\bar{T}(t) = \frac{TT1(t) + TT2(t)}{2} \quad (4.6)$$

where:

- A_{sf} , Collector absorber cross-section area
- ρ , Water density
- c_p , Specific heat capacity
- $TT2$, Outlet temperature of the solar field
- β , Irradiance model parameter
- I , Global irradiance
- H , Global thermal losses coefficient
- L_{eq} , Equivalent absorber tube length
- T_a , Ambient temperature
- $TT1$, Inlet temperature of the solar field
- c_f , Conversion factor to account for connections, number of modules and L/min conversion

The FF action is calculated at the steady state and its formula is shown in Eqn. 4.7. This formula is obtained by placing $TT2 = TT2_{SP}$ in Eqn. 4.4

$$FT1_{ff} = \left[\frac{\beta L_{eq}}{c_p(TT2_{sp}(t) - TT1(t))} I(t) - \frac{H}{c_p} \frac{(\bar{T}(t) - T_a(t))}{(TT2_{sp}(t) - TT1(t))} \right] \frac{c_f}{\rho} \quad (4.7)$$

$$\bar{T}(t) = \frac{TT1(1) + TT2_{sp}(t)}{2} \quad (4.8)$$

where $FT1_{ff}$ is the feed forward action. If the real system was equal to the model, the PIDs controllers action would have zero value at the steady state.

Filter A The filter A is placed on the set-point in order to smooth the changes. This filter reduces the dynamics with which temperature can be changed. This makes sense because the outlet temperature in the solar field has got a slow dynamics (in the order of minutes) and a sharp change in the set point will stress the actuators without huge time advantages. Its equation is shown in Eqn. 4.9

$$F_a(s) = \frac{1}{60s + 1} \quad (4.9)$$

Filter B The filter B is placed on the FF output. As it is well known, the FF action is usually intence, so is preferable to put a filter on it to smooth it behaviour. Without it, the actuators will be stressed accordingly to a rapid change in the variables. The Eqn 4.10 describes it.

$$F_b(s) = \frac{1}{75s + 1} \quad (4.10)$$

Saturation As it was done for the inner loop, the saturation blocks and the anti windup system are implemented with no constant saturation limits. This methodology is necessary in this case too because the PIDs limits are conditioned by the action of FF. The PID action must respect the limits described in Eqn. 4.11 and Eqn. 4.12

$$U_{PID-MAX} = Flow_{max} - FT1_{ff} \quad (4.11)$$

$$U_{PID-MIN} = Flow_{min} - FT1_{ff} \quad (4.12)$$

In this way the signal produced by this part of the controller will not get out from the limits. The flow limits are set equal to:

- $Flow_{max-loop1} = 17 \text{ l/min}$
- $Flow_{max-loopj} = 28 \text{ l/min } j=3,4,5$
- $Flow_{min-loop1} = 7 \text{ l/min}$
- $Flow_{min-loopj} = 10 \text{ l/min } j=3,4,5$

These values can be reached without keeping saturated actuators. This point is important for a closed loop control.

Chapter 5

Control - simulations and experiments

5.1 Introduction

In this chapter the results obtained from the simulations will be shown together with the results obtained when it was possible to test the controllers in the real plant. Not all the possible cases have been tested in the real field due to technical problems and overlapping experiments. Also the weather conditions played a fundamental role because with too high sun radiation it was not possible to operate in accordance with the safety conditions required by the plant. Since the gain of the system is variable, it would be advisable to develop an auto-tuning system, in order to have the controller well calibrated for each condition. In this thesis we will not deal with this but we will use simple tuning rules that can be used in the auto-tuning algorithms, to optimize start-up times. A LabView interface is required to configure the controllers, implemented in Simulink, with the solar field. Inside of this LabView interface also the feedback signal filters are implemented. These filters have been implemented directly here because they are required in all the controllers and because in this way the initial condition problems can be simplified. The LabView interface is presented in the Cap. A.

Significant are the different step amplitudes in the following experiments. It was necessary to test the different controllers with different step sizes because the experiments were done in different days with different weather conditions and in different hours of the day. As already mentioned, due to the temperature it was not always possible to lower the flow for a prolonged time.

5.2 Flow control - gain scheduling control

The controller illustrated in Chap.3.2 was tested in the simulation and in the real plant. The gain scheduling model (Chap. 2.2) was used to test this controller in the simulation. In the following tests the behavior of the controller will be shown with different types of tunings. One of the main problems was the choice of sampling time. A too short sampling time would not give the time to the computer to perform the necessary calculations. A too long sampling time would make instead the control inaccurate. From the tests carried out with this controller it has emerged that the computational times never exceed three seconds (sometimes it takes just a little more than two seconds). To be consistent with what it was said, a sampling time of three seconds was selected.

$$T_s = 3s \tag{5.1}$$

5.2.1 Test 1 - tuning Kappa Tau $M_s = 2.0$

In this calibration method it is necessary:

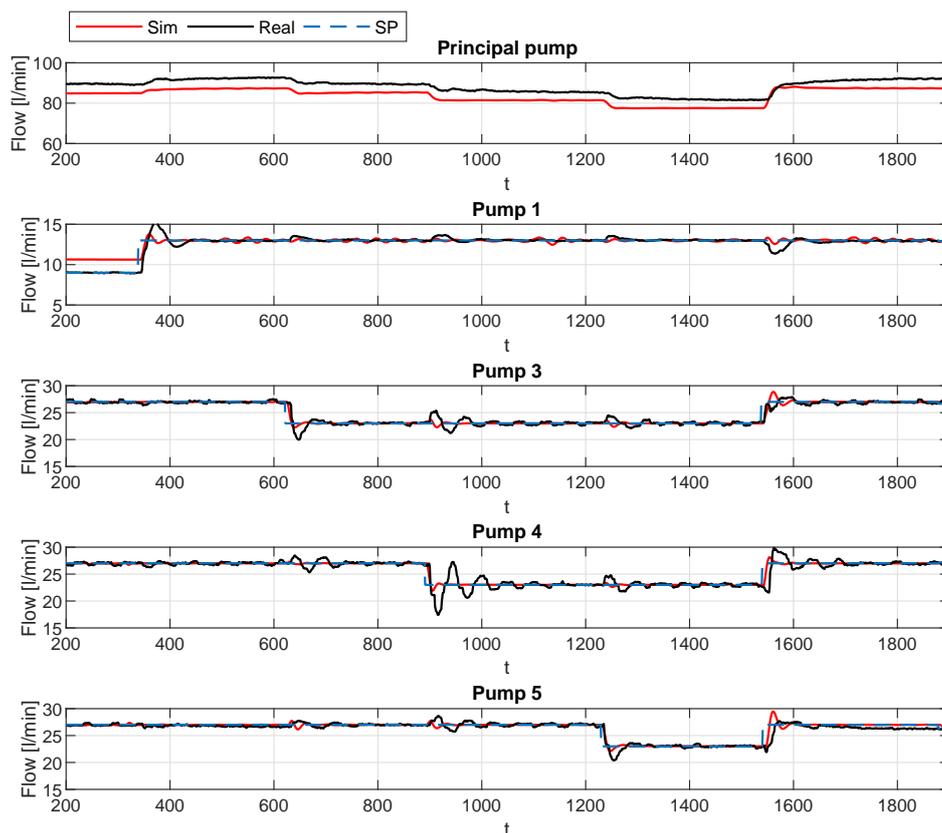
1. to calculate the result of a specific function,
2. with the result obtained at point one, to enter in a table for the identification of the parameters of the PIDs.

The interesting thing with this method is the possibility to choose the maximum sensitivity M_S of the system. In this chapter the results for a phase margin of 2 will be illustrated.

Table 5.1: Tuning gain scheduling controller - Kappa Tau Methods $M_s = 2.0$, PIDs parameters value

Controller $u_0 < 50\%$	K_p	T_i	Controlloer $u_0 > 50\%$	K_p	T_i
Loop 1	1.7555	3.1295	Loop 1	2.1999	3.1295
Loop 3	1.1600	3.4149	Loop 3	1.3128	3.4149
Loop 4	1.3704	2.2809	Loop 4	1.5118	2.2809
Loop 5	1.1409	3.2109	Loop 5	1.2943	3.2109

In Fig. 5.1 and Fig. 5.2, the results obtained from the simulation are compared with the results obtained from the real plant. The loop 1 was tested with steps from 9 l/min to 13 l/min (31% of step), and the other loops were tested with steps from 27 l/min to 23 l/min (17% of step). The controller in the real plant was tested on the 28/06/2018 at 10:29

Figure 5.1: Gain scheduling control - Tuning Kappa Tau $M_s = 2.0\%$ - Flow, comparison between simulated and real signal

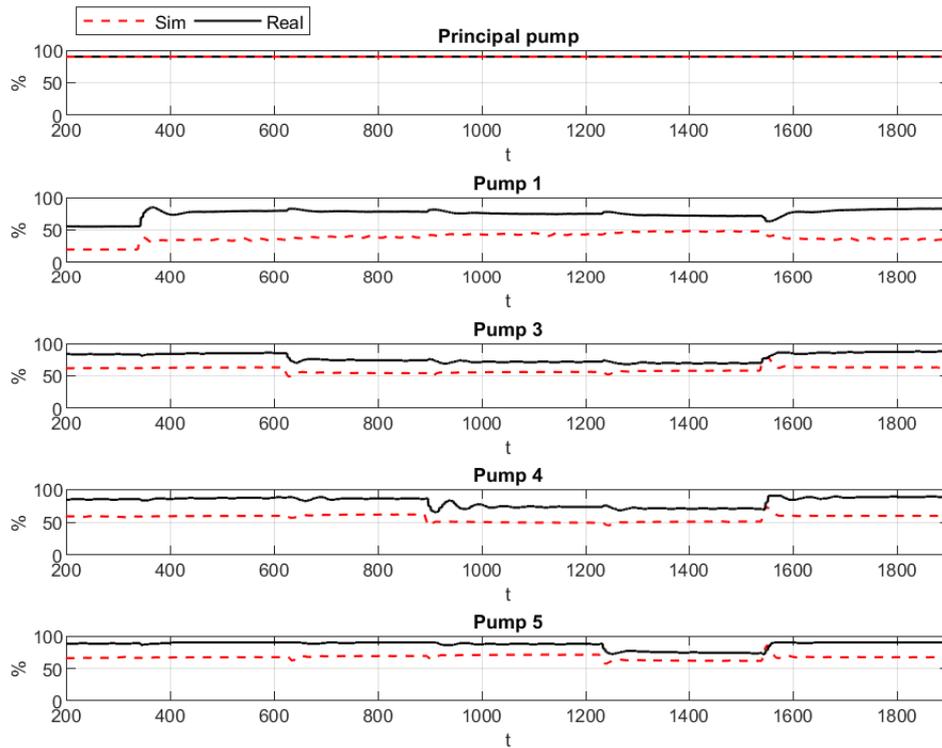


Figure 5.2: Gain scheduling control - Tuning Kappa Tau $M_s = 2.0\%$ - Control signal, comparison between simulated and real signal

The following points can be noted:

- In the simulation the principal pump flow doesn't follow the real behavior. The value, in this case, should be equal to the sum of the flows 1 3 4 5. In the real field this happens due to the feature of the plant itself. In the simulation the errors introduced by the non-linearity and the open loop control used caused the appearance of an error.
- The oscillations introduced into the flow are greater in the real plant. In particular, loop 4 presents the worst situation. These fluctuations are not too high in the control signals.
- note how in the real system the control signals are always higher than those obtained

in simulation.

- in pump 1 (at the start) and in pump 5 (at the end) the simulation and the reality are different due to the actuator saturation.

In detail the loops are characterized by the following performance parameters:

- Loop 1: $Overshoot_{sim} = 6.2\%$ $Overshoot_{real} = 15.4\%$
- Loop 3: $Overshoot_{sim} = 3.0\%$ $Overshoot_{real} = 13.1\%$
- Loop 4: $Overshoot_{sim} = 5.2\%$ $Overshoot_{real} = 24.2\%$
- Loop 5: $Overshoot_{sim} = 4.0\%$ $Overshoot_{real} = 11.3\%$
- Loop 1: $Ta_{sim} = 26s$ $Ta_{real} = 83s$
- Loop 3: $Ta_{sim} = 10s$ $Ta_{real} = 39s$
- Loop 4: $Ta_{sim} = 9s$ $Ta_{real} = 120s$
- Loop 5: $Ta_{sim} = 10s$ $Ta_{real} = 36s$

5.2.2 Test 2 - tuning Kappa Tau $M_s = 1.4$

The Kappa Tau tuning method was used also with a phase margin equal to 1.4 .

In Tab. 5.2 the tuning parameters are presented.

Table 5.2: Tuning gain scheduling controller - Kappa Tau Methods $M_s = 1.4$, PIDs parameters value

Controller $u_0 < 50\%$	K_p	T_i	Controlloer $u_0 > 50\%$	K_p	T_i
Loop 1	0.7461	3.1295	Loop 1	0.9350	3.1295
Loop 3	0.5252	3.4149	Loop 3	0.5944	3.4149
Loop 4	0.5972	2.2809	Loop 4	0.6588	2.2809
Loop 5	0.5107	3.2109	Loop 5	0.5794	3.2109

The results are presented in Fig. 5.3 and Fig. 5.4. The loop 1 was tested with steps from 8 l/min to 12 l/min (33% of step), and the other loops were tested with steps from

25 l/min to 20 l/min (25% of step). The controller in the real plant was tested on the 26/06/2018 at 14:56

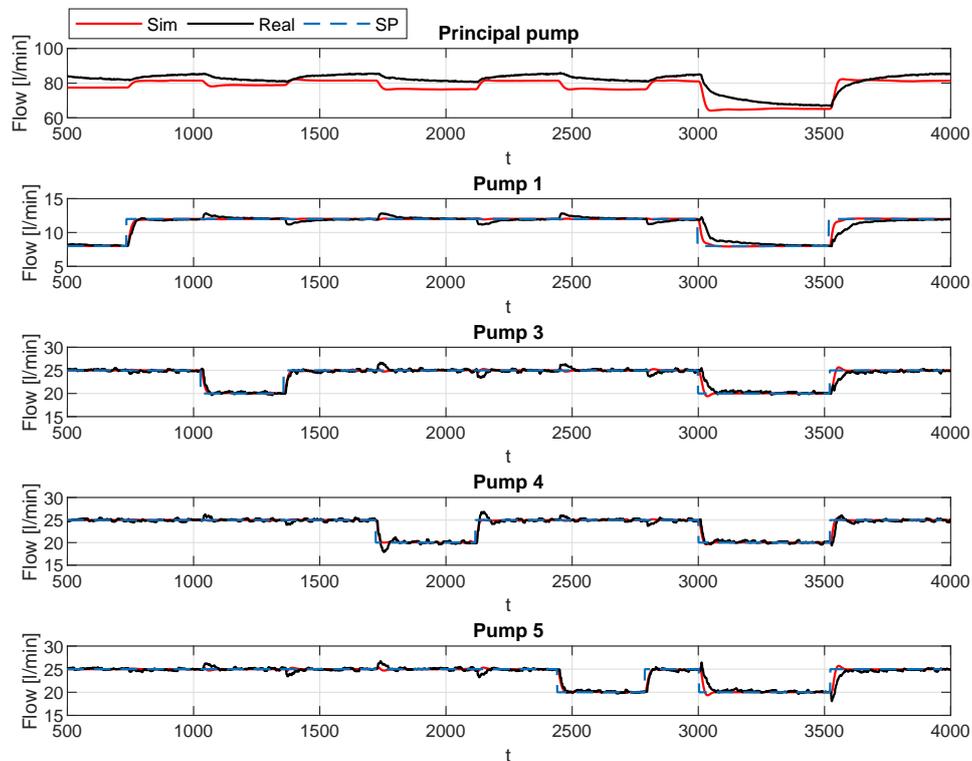


Figure 5.3: Gain scheduling control - Tuning Kappa Tau $M_s = 1.4\%$ - Flow, comparison between simulated and real signal

The following points can be noted:

- also in this case, the flow simulation on the main pump does not reflect reality.
- in all the loops, the real and simulated flow values reach the set point.
- For the loop 1, the simulation has a faster dynamics. This influences the rejection of load disturbances. In fact, it is possible to see in the real plant how the steps on the loops 3 4 5 are considerably reflected on loop 1.
- the system, with this tuning, does not present high oscillations in the flow and in

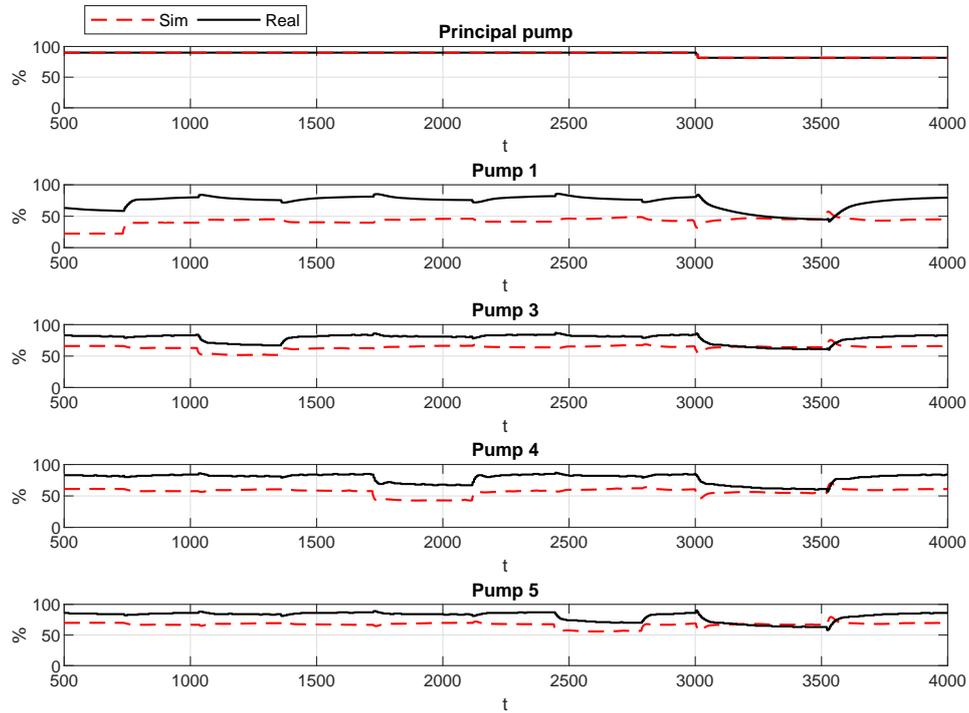


Figure 5.4: Gain scheduling control - Tuning Kappa Tau $M_s = 1.4\%$ - Control signal, comparison between simulated and real signal

the control signal. In order to test it in its entirety, at the end of the simulation all the set points were then changed together. The result can be considered satisfactory, even if the loop 1 has a particularly slow dynamic.

More specifically, the loops are characterized by the following performance parameters:

- Loop 1: $Overshoot_{sim} \simeq 0\%$ $Overshoot_{real} \simeq 0\%$
- Loop 3: $Overshoot_{sim} \simeq 0\%$ $Overshoot_{real} \simeq 0\%$
- Loop 4: $Overshoot_{sim} \simeq 0\%$ $Overshoot_{real} = 10.5\%$
- Loop 5: $Overshoot_{sim} \simeq 0\%$ $Overshoot_{real} \simeq 0\%$
- Loop 1: $Ta_{sim} = 27s$ $Ta_{real} = 36s$

- Loop 3: $Ta_{sim} = 21s$ $Ta_{real} = 28s$
- Loop 4: $Ta_{sim} = 16s$ $Ta_{real} = 21s$
- Loop 5: $Ta_{sim} = 21s$ $Ta_{real} = 30s$

5.2.3 Test 3 - C.H.R. tuning set-point following

The Chien Hrones Reswch tuning rules allow to choose among different tasks. In this test, for the set point following it was chosen the table with an overshoot of 20%. In Tab. 5.3 there are shown the PIDs parameters.

Table 5.3: Tuning gain scheduling controller - CHR tuning Set-Point following, PIDs parameters value

Controller u0<50%	K_p	T_i	Controlloer u0>50%	K_p	T_i
Loop 1	2.1086	4.2416	Loop 1	2.6424	4.2416
Loop 3	1.6540	4.9042	Loop 3	1.8718	4.9042
Loop 4	1.7620	3.1721	Loop 4	1.9438	3.1721
Loop 5	1.5775	4.5723	Loop 5	1.7897	4.5723

In Fig. 5.5 and Fig. 5.6 the results are presented. The loop 1 was tested with steps from 10 l/min to 15 l/min (33% of step), and the other loops were tested with steps from 27 l/min to 24 l/min (13% of step). The controller in the real plant was test on the 26/06/2018 at 10:02

The following points can be noted:

- as in the previous test, the simulation and the real flow are different for the principal pump
- for the pump 1, this tuning is appropriate. The oscillations are not high and lasting, both in the simulation and in reality. In particular, also the load disturbance rejection is correctly performed.
- for loop 3 and 5 the flows follow the set-point in a proper way but in the real plant the load disturbance rejection are shoddy.

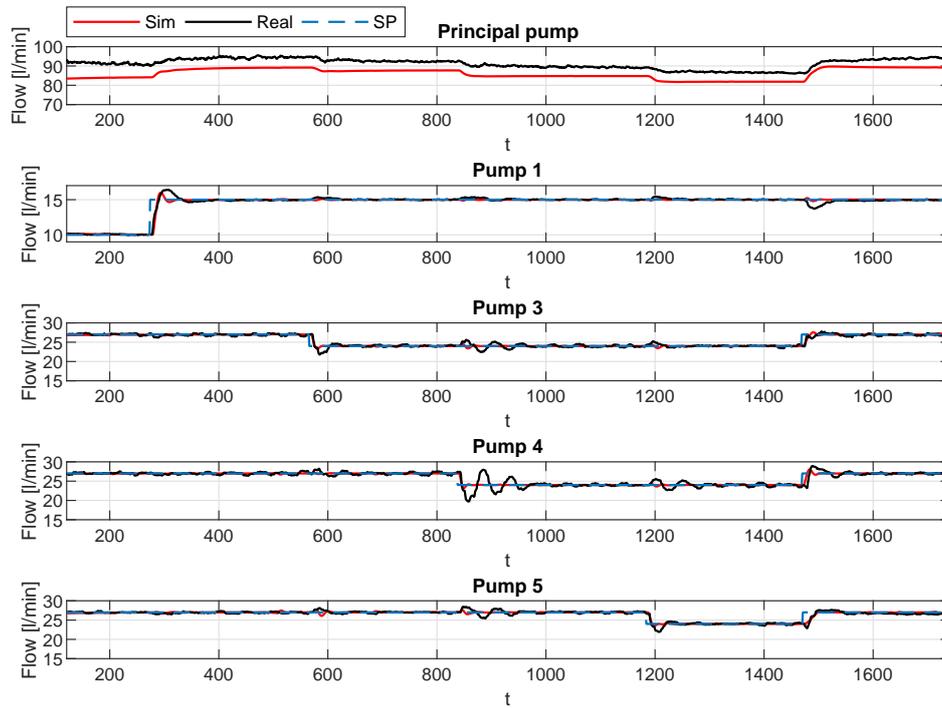


Figure 5.5: Gain scheduling control - Tuning CHR Set-Point following - Flow, comparison between simulated and real signal

- loop 4 presents the opposite situation. The load disturbance rejection performed good but the set point following task is not applicable, in particular it presents high and lasting oscillation.

In detail the loops are characterized by the following performance parameters:

- Loop 1: $Overhoot_{sim} = 6.7\%$ $Overhoot_{real} = 9.2\%$
- Loop 3: $Overhoot_{sim} = 2.0\%$ $Overhoot_{real} = 9.3\%$
- Loop 4: $Overhoot_{sim} = 3.3\%$ $Overhoot_{real} = 17.7\%$
- Loop 5: $Overhoot_{sim} = 2.5\%$ $Overhoot_{real} = 8.54\%$
- Loop 1: $Ta_{sim} = 23s$ $Ta_{real} = 45s$

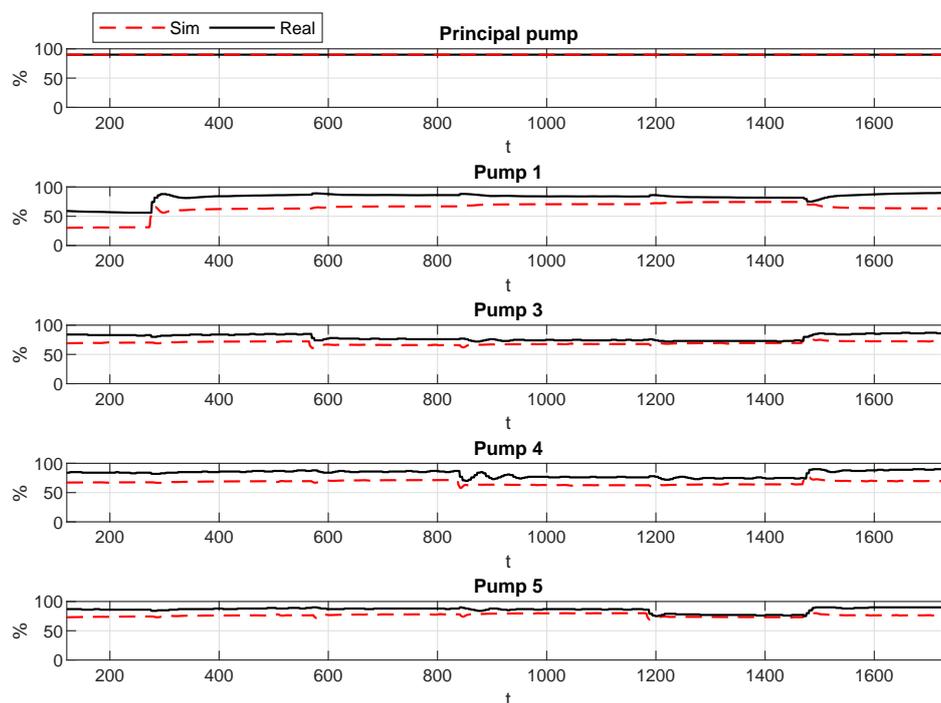


Figure 5.6: Gain scheduling control - Tuning CHR Set-Point following - Control signal, comparison between simulated and real signal

- Loop 3: $T_{a_{sim}} = 9s$ $T_{a_{real}} = 34s$
- Loop 4: $T_{a_{sim}} = 7s$ $T_{a_{real}} = 122s$
- Loop 5: $T_{a_{sim}} = 9s$ $T_{a_{real}} = 31s$

5.2.4 Test 4 - C.H.R. tuning load disturbance rejection

In this test, for the load disturbance rejection it was chosen the table with an overshoot of 20%. The PIDs parameters are shown in Tab. 5.4

In Fig. 5.7 and Fig. 5.8 the results are presented. The loop 1 was tested with steps from 10 l/min to 15 l/min (33% of step), and the other loops were tested with steps from 23 l/min to 27 l/min (15% of step). The controller in the real plant was tested on the 26/06/2018 at 10:47

Table 5.4: Tuning gain scheduling controller - CHR tuning load disturbance rejection, PIDs parameters value

Controller $u_0 < 50\%$	K_p	T_i	Controller $u_0 > 50\%$	K_p	T_i
Loop 1	2.4600	13.800	Loop 1	3.0828	13.800
Loop 3	1.9296	11.500	Loop 3	2.1838	11.500
Loop 4	2.0556	9.2000	Loop 4	2.2677	9.2000
Loop 5	1.8404	11.500	Loop 5	2.0879	11.500

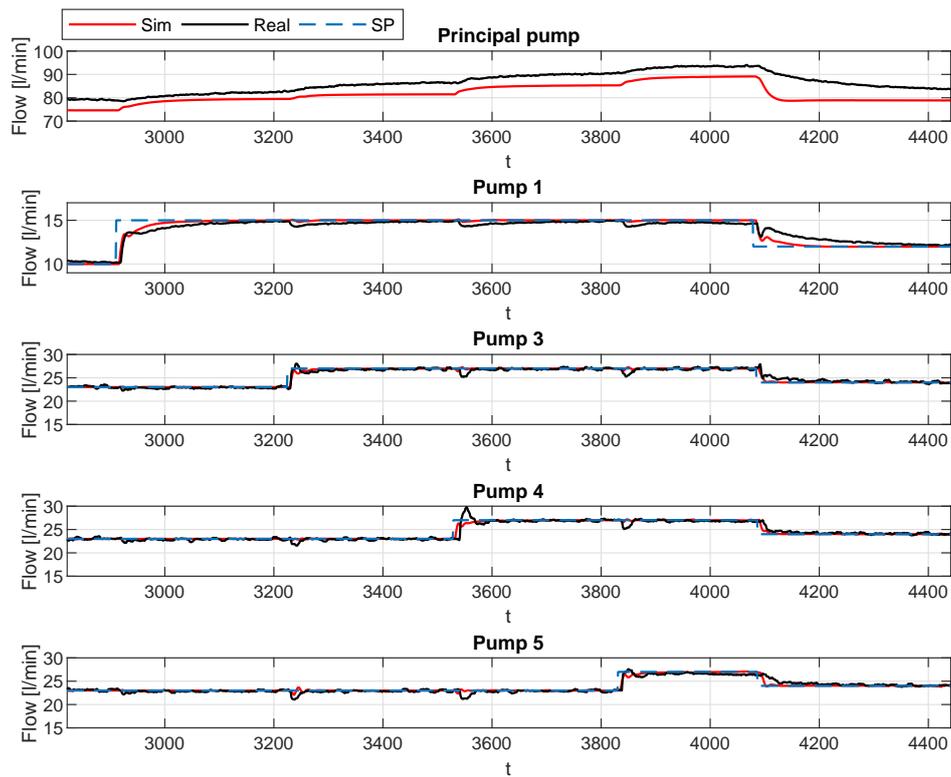


Figure 5.7: Gain scheduling control - Tuning CHR load disturbance rejection - Flow, comparison between simulated and real signal

The following points can be noted:

- for loop 0, the imprecision of the model is not compensated by a closed loop control
- loop 1 closed loop dynamics is too slow. The set point following and the load disturbance rejection need a long time to reach the set point

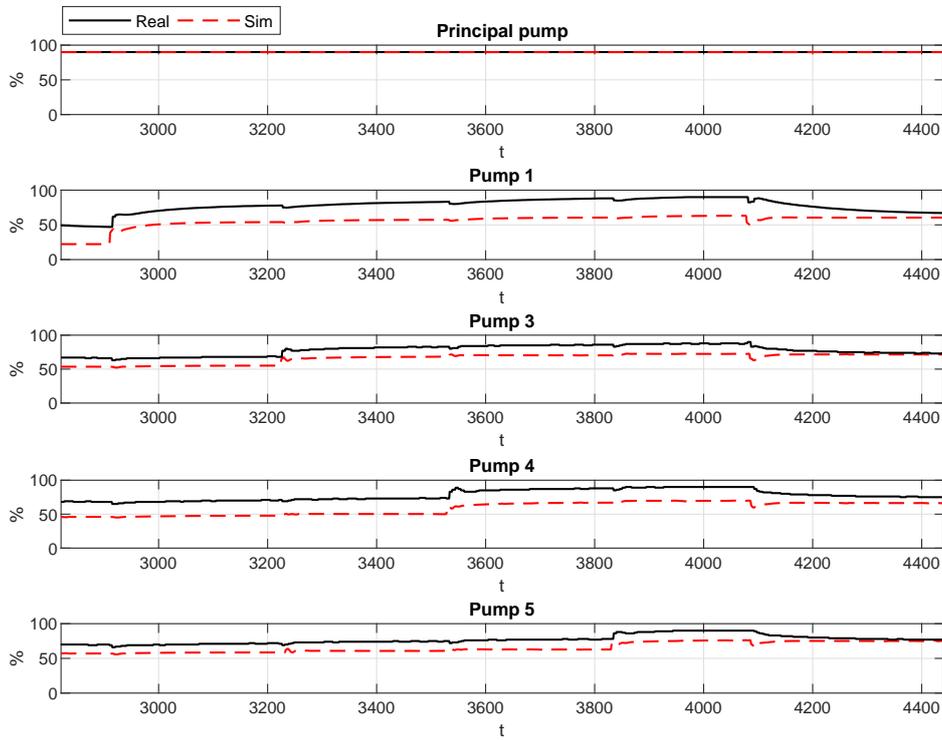


Figure 5.8: Gain scheduling control - Tuning CHR load disturbance rejection - Control signal, comparison between simulated and real signal

- for the loops 3, 4, and 5 this PID's tuning provides good performance, both in the simulation and in the real plant. The set points are reached with a little overshoot and not a lot of oscillation. Moreover, the load disturbance rejection is carried out in some seconds and without strong oscillation.

In detail the loops are characterized by the following performance parameters:

- Loop 1: $Overshoot_{sim} \simeq 0\%$ $Overshoot_{real} \simeq 0\%$
- Loop 3: $Overshoot_{sim} \simeq 0\%$ $Overshoot_{real} = 4.8\%$
- Loop 4: $Overshoot_{sim} \simeq 0\%$ $Overshoot_{real} = 10.3\%$
- Loop 5: $Overshoot_{sim} \simeq 0\%$ $Overshoot_{real} = 2.1\%$

- Loop 1: $Ta_{sim} = 53s$ $Ta_{real} = 102s$
- Loop 3: $Ta_{sim} = 10s$ $Ta_{real} = 8s$
- Loop 4: $Ta_{sim} = 8s$ $Ta_{real} = 31s$
- Loop 5: $Ta_{sim} = 10s$ $Ta_{real} = 9s$

5.3 Flow control - non-linear control

In this section are presented the simulation results and the real plant experiments for the 'non-linear controller' presented in Cap. 3.3. It was tuned with different rules and the results are shown below. In order to be consistent, this controller was simulated with the model presented in Cap. 2.3. The sampling time must be greater than the computational time. For this controller the computational time was always under 2s so the sampling time was set equal to 2s. It is noticeable that with this controller, the sampling time was reduced compared with the previous.

$$T_s = 2s \quad (5.2)$$

5.3.1 Test 1 - tuning Kappa Tau $M_s = 2.0$

In this first test the controller was tested with Kappa Tau tuning rules. A phase margin of 2.0 was chosen. In Tab. 5.6 the PID's parameters values are shown.

Table 5.5: Tuning non-linear controller - Kappa Tau Methods $M_s = 2.0$, PID's parameters value

Controller	Kp	Ti
Loop 1	0.3531	3.1295
Loop 3	0.4127	3.4149
Loop 4	0.3701	2.2809
Loop 5	0.3968	3.2109

In Fig. 5.9 and Fig. 5.10 it is possible to observe the results obtained from the simulation and the real plant. The controller was tested with steps from 12 l/min to 16 l/min (25% of step) for the loop 1 and with steps from 24 l/min to 28 l/min (14% of step) for the other loops. The test on the real plant was done on the 16/07/2018 at 15:18

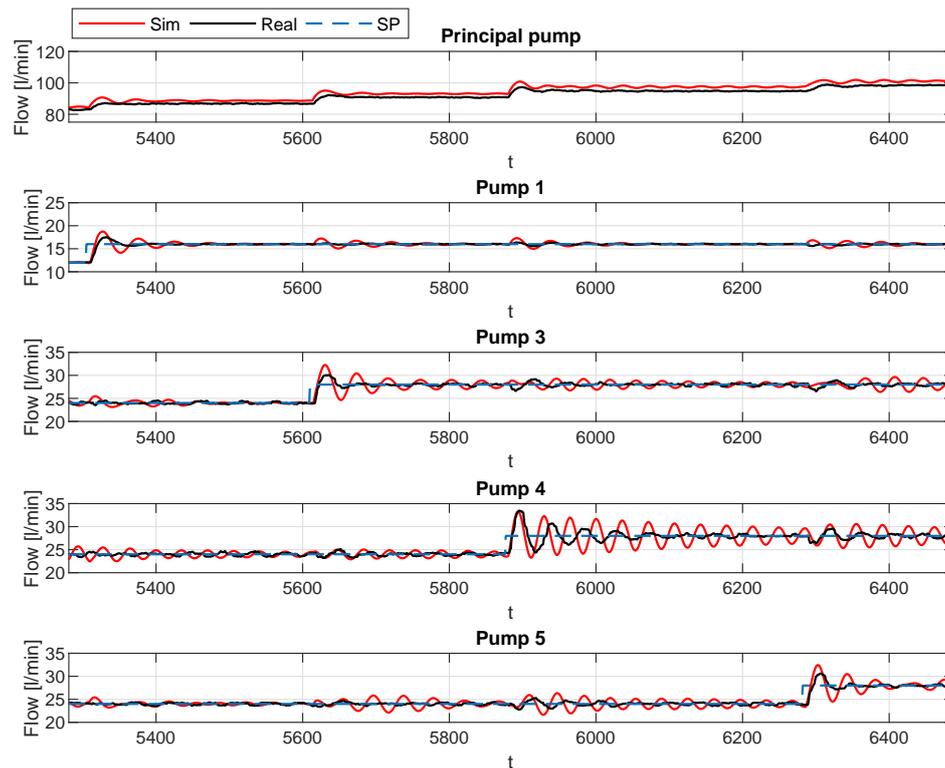


Figure 5.9: Non-linear control - Tuning Kappa Tau $M_s = 2.0\%$ - Flow, comparison between simulated and real signal

The following points can be noted:

- Loop 1 achieves the set point value but slowly and with strong oscillations. Moreover the interaction with the other loops is not well compensated and it needs a lot of time to eliminate the oscillation.
- Loop 3, 4, and 5 do not reach the set point in a stable way. The oscillations are maintained over time. These oscillations are not divergent but they cancel in an

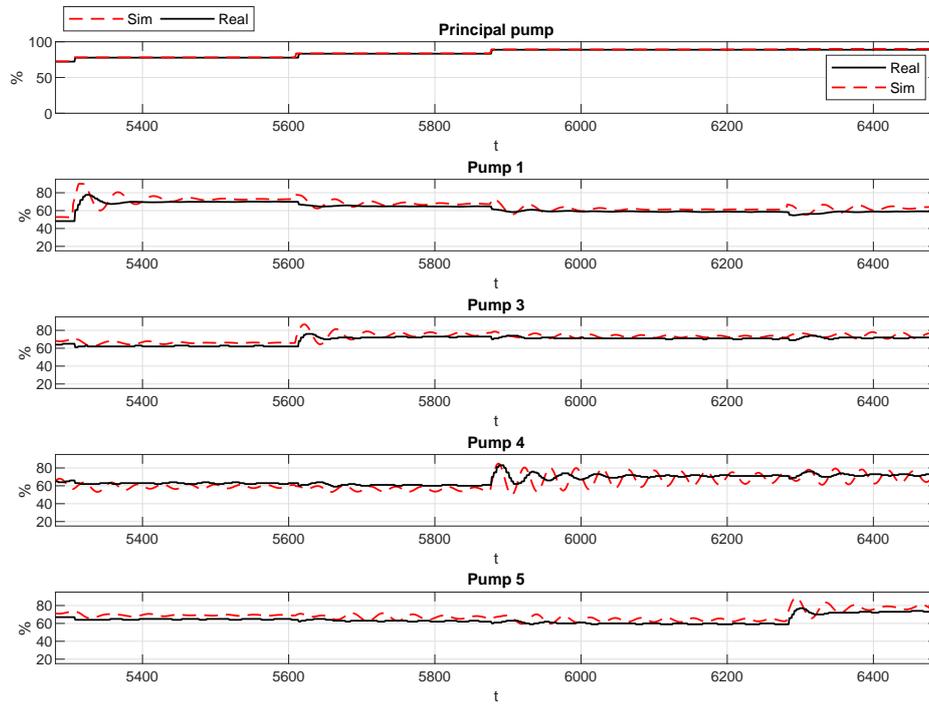


Figure 5.10: Non-linear control - Tuning Kappa Tau $M_s = 2.0\%$ - Control signal, comparison between simulated and real signal

excessively long time.

- from the observations made in the previous points we can say that this tuning rule is not suitable for this system.

In detail the loops are characterized by the following performance parameters: overshoot

- Loop 1: $Overshoot_{sim} = 17.1\%$ $Overshoot_{real} = 9.4\%$
- Loop 3: $Overshoot_{sim} = 15.1\%$ $Overshoot_{real} = 7.4\%$
- Loop 4: $Overshoot_{sim} = 18.5\%$ $Overshoot_{real} = 19.3\%$
- Loop 5: $Overshoot_{sim} = 15.5\%$ $Overshoot_{real} = 9.1\%$

- Loop 1: $Ta_{sim} = 98s$ $Ta_{real} = 38s$
- Loop 3: $Ta_{sim} = 107s$ $Ta_{real} = 31s$
- Loop 4: $Ta_{sim} = 392s$ $Ta_{real} = 132s$
- Loop 5: $Ta_{sim} = 84s$ $Ta_{real} = 31s$

5.3.2 Test 2 - tuning Kappa Tau $M_s = 1.4$

In this test the controller was tested with Kappa Tau tuning rules. A phase margin of 1.4 was chosen. In Tab. 5.6 the PID's parameters values are shown.

Table 5.6: Tuning non-linear controller - Kappa Tau Methods $M_s = 1.4$, PID's parameters value

Controller	Kp	Ti
Loop 1	0.1501	3.1295
Loop 3	0.1869	3.4149
Loop 4	0.1613	2.2809
Loop 5	0.1776	3.2109

In Fig. 5.11 and Fig. 5.12 it is possible to observe the results obtained from the simulation and the real plant. The controller was tested with steps from 12 l/min to 16 l/min (25% of step) for the loop 1 and with steps from 24 l/min to 28 l/min (14% of step) for the other loops. The test on the real plant was done the 16/07/2018 at 14:51

The following points can be noted:

- In all the loops the oscillation are reduced, compared with the previous tuning.
- Loop 1 in particular doesn't present an overshoot in the real plant. Also the load disturbance rejection is greater in the real plant than in the simulation
- Loop 3 and 5 present an insignificant overshoot in both the real plant and in the simulation.
- Only the loop 4 presets a relevant overshoot but also in this case the step response presents a good shape.

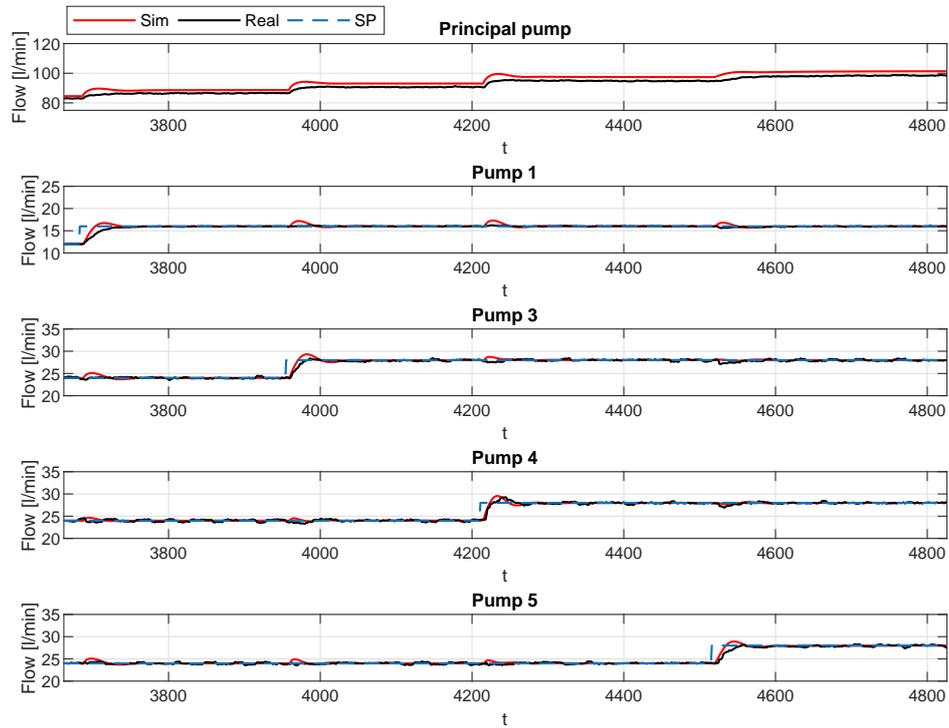


Figure 5.11: Non-linear control - Tuning Kappa Tau $M_s = 1.4\%$ - Flow, comparison between simulated and real signal

- All the loops present a good load disturbance rejection.
- About what it was said at the previous point, this tuning rules can be considered appropriate

In detail the loops are characterized by the following performance parameters:

- Loop 1: $Overhoot_{sim} = 4.6\%$ $Overhoot_{real} = 0\%$
- Loop 3: $Overhoot_{sim} = 4.7\%$ $Overhoot_{real} = 1.3\%$
- Loop 4: $Overhoot_{sim} = 5.4\%$ $Overhoot_{real} = 4.5\%$
- Loop 5: $Overhoot_{sim} = 3.25\%$ $Overhoot_{real} = 1\%$
- Loop 1: $Ta_{sim} = 16s$ $Ta_{real} = 31s$

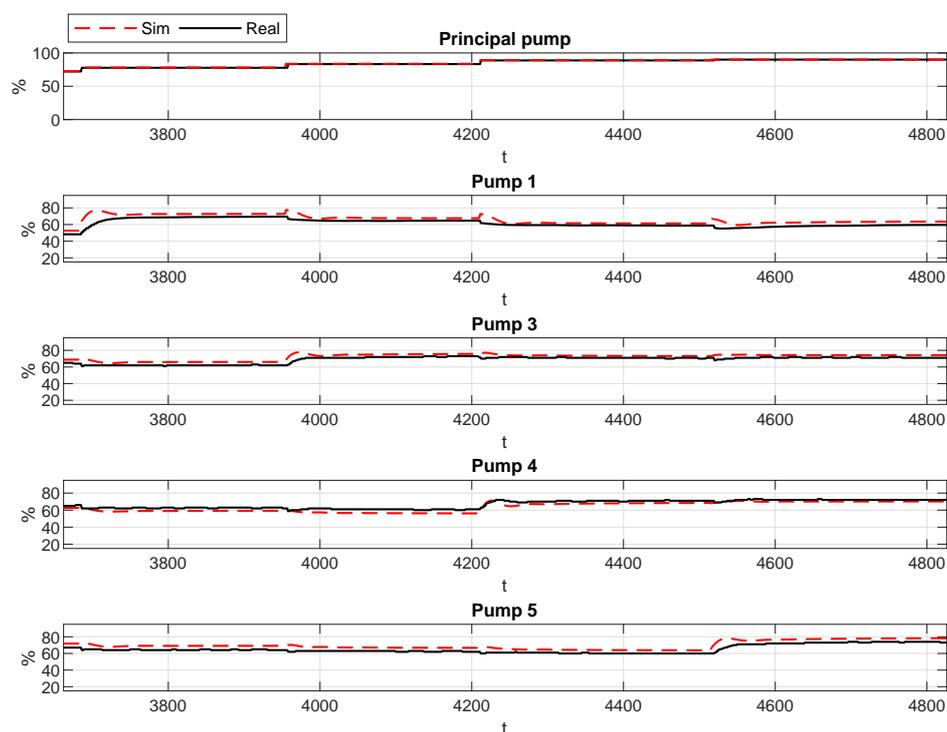


Figure 5.12: Non-linear control - Tuning Kappa Tau $M_s = 1.4\%$ - Control signal, comparison between simulated and real signal

- Loop 3: $Ta_{sim} = 11s$ $Ta_{real} = 17s$
- Loop 4: $Ta_{sim} = 26s$ $Ta_{real} = 10s$
- Loop 5: $Ta_{sim} = 13s$ $Ta_{real} = 21s$

5.3.3 Test 3 - C.H.R. tuning set-point following

In this test it was chose the tuning table for the set point following with an overshoot of 20%. In Tab. 5.7 the PIDs parameters are shown.

In Fig. 5.13 and Fig. 5.14 it is possible to observe the results obtained from the simulation and the real plant. The controller was tested with steps from 12 l/min to 16 l/min (25% of step) for the loop 1 and with steps from 24 l/min to 28 i/min (14% of step) for

Table 5.7: Tuning non-linear controller - CHR tuning Set-Point following, PIDs parameters value

Controller	K_p	T_i
Loop 1	0.4242	4.2416
Loop 3	0.5885	4.9042
Loop 4	0.4758	3.1721
Loop 5	0.5487	4.5723

the other loops. The test on the real plant was done on the 16/07/2018 at 13:50

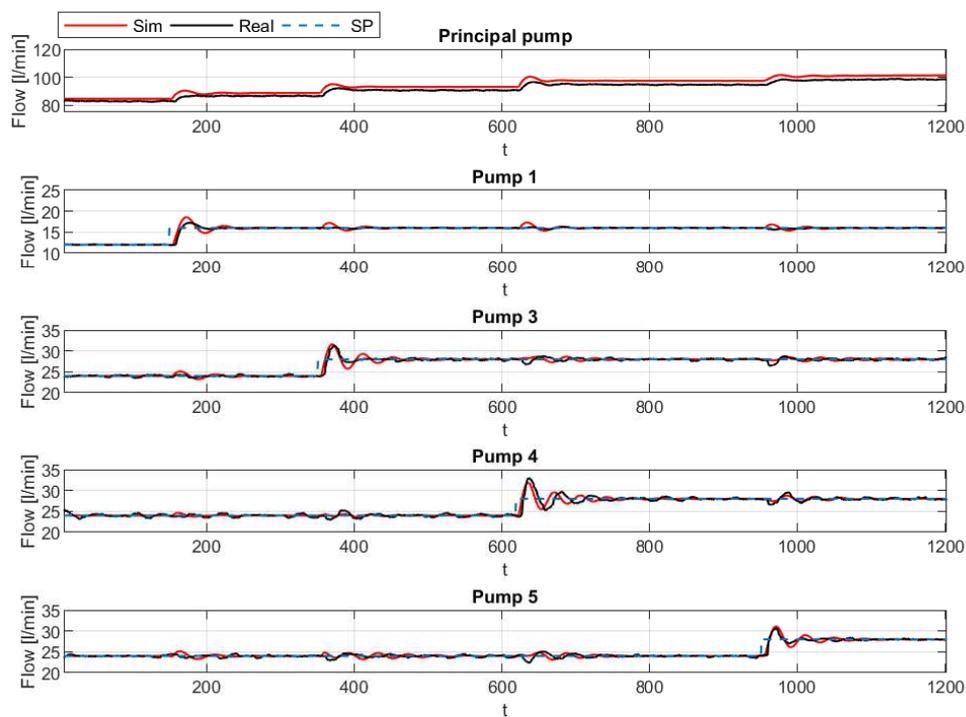


Figure 5.13: Non-linear control - Tuning CHR Set-Point following - Flow, comparison between simulated and real signal

The following points can be noted:

- All the loops reach the set point in a proper way.
- Loop 1, 3 and 5 present a negligible oscillation in the simulation and in the real plant. Moreover, the load disturbance rejection takes place in a appropriate way.

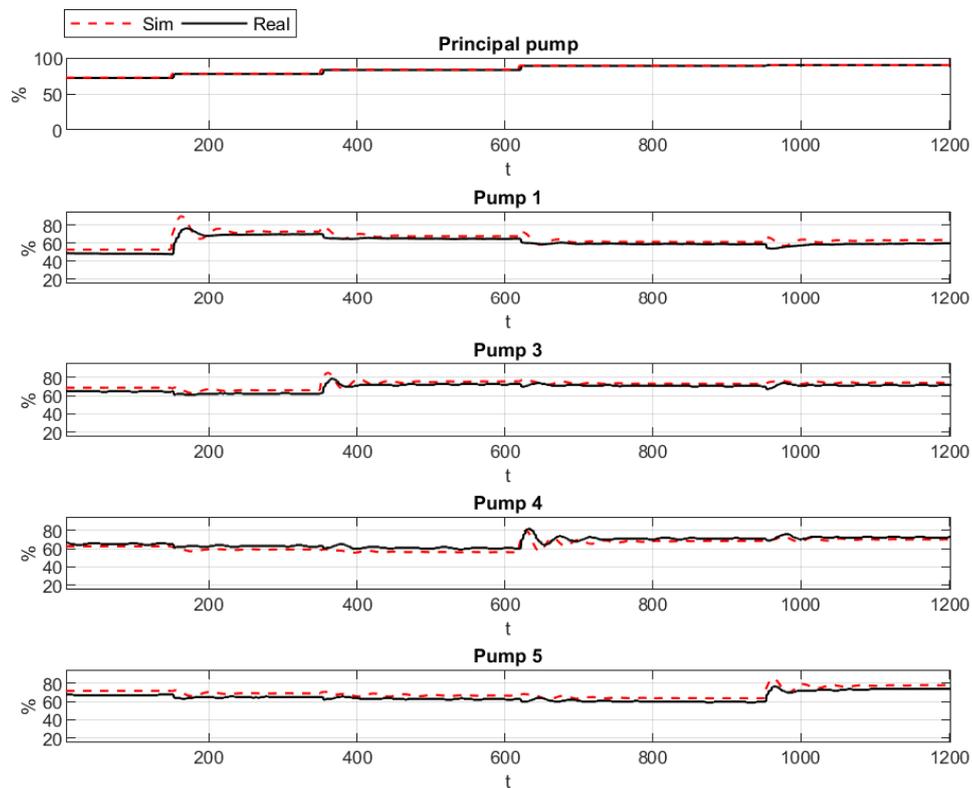


Figure 5.14: Non-linear control - Tuning CHR Set-Point following - Control signal, comparison between simulated and real signal

- Loop 4 presents some more problems. The oscillations are significant and they need a long time to be near zero. The load disturbance continues to be relevant but we can consider it acceptable

In detail the loops are characterized by the following performance parameters:

- Loop 1: $Overhoot_{sim} = 16.6\%$ $Overhoot_{real} = 7.6\%$
- Loop 3: $Overhoot_{sim} = 15\%$ $Overhoot_{real} = 11.7\%$
- Loop 4: $Overhoot_{sim} = 15.8\%$ $Overhoot_{real} = 17.6\%$
- Loop 5: $Overhoot_{sim} = 12.8\%$ $Overhoot_{real} = 9.6\%$

- Loop 1: $Ta_{sim} = 53s$ $Ta_{real} = 36s$
- Loop 3: $Ta_{sim} = 80s$ $Ta_{real} = 28s$
- Loop 4: $Ta_{sim} = 86s$ $Ta_{real} = 64s$
- Loop 5: $Ta_{sim} = 60s$ $Ta_{real} = 23s$

5.3.4 Test 4 - C.H.R. tuning load disturbance rejection

In this test it was chose the tuning table for the load disturbance rejection task with an overshoot of 20%. In Tab. 5.8 the PIDs parameters are shown.

Table 5.8: Tuning non-linear controller - CHR tuning load disturbance rejection, PIDs parameters value

Controller	Kp	Ti
Loop 1	0.4948	13.8
Loop 3	0.6866	11.5
Loop 4	0.5551	9.2
Loop 5	0.6401	11.5

In Fig. 5.15 and Fig. 5.16 it is possible to observe the result obtained from the simulation and the real plant. The controller was tested with steps from 12 l/min to 16 l/min (25% of step) for the loop 1 and with steps from 24 l/min to 28 i/min (14% of step) for the other loops. The test on the real plant was done on the 16/07/2018 at 14:22

The following points can be noted:

- Loop 1 reaches the set point in a long time without overshoot. The load disturbance rejection is performed properly in the real plant even if in the simulation it doesn't happen.
- Loop 3, 4, and 5 are well tuned. The set point are achieved quickly and with no strong oscillation. The load disturbance rejections are also performed in a proper way.

In detail the loops are characterized by the following performance parameters:

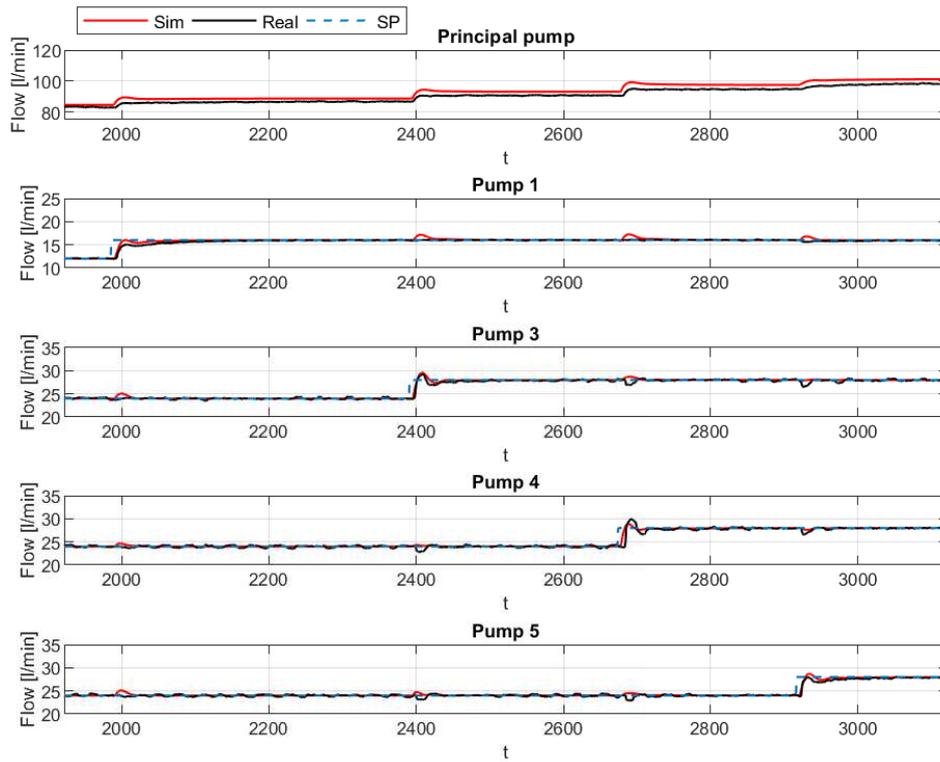


Figure 5.15: Non-linear control - Tuning CHR load disturbance rejection - Flow, comparison between simulated and real signal

- Loop 1: $Overshoot_{sim} = 1.75\%$ $Overshoot_{real} = 0\%$
- Loop 3: $Overshoot_{sim} = 7\%$ $Overshoot_{real} = 4.46\%$
- Loop 4: $Overshoot_{sim} = 4.4\%$ $Overshoot_{real} = 6.9\%$
- Loop 5: $Overshoot_{sim} = 3.9\%$ $Overshoot_{real} = 0\%$
- Loop 1: $Ta_{sim} = 11s$ $Ta_{real} = 60s$
- Loop 3: $Ta_{sim} = 21s$ $Ta_{real} = 9s$
- Loop 4: $Ta_{sim} = 6s$ $Ta_{real} = 23s$
- Loop 5: $Ta_{sim} = 9s$ $Ta_{real} = 8s$

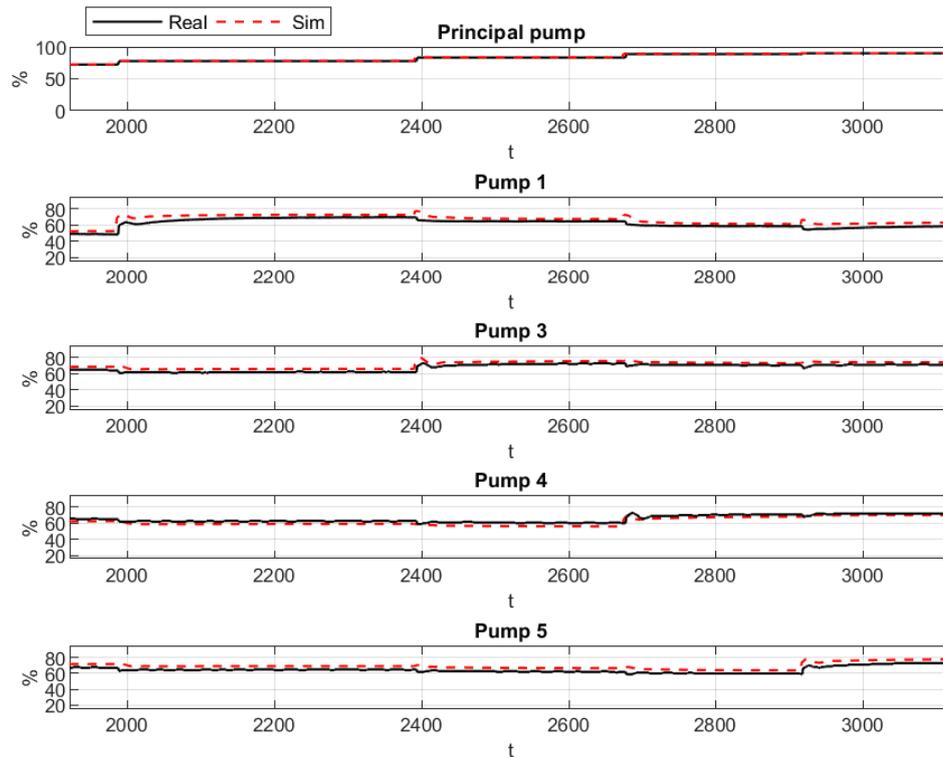


Figure 5.16: Non-linear control - Tuning CHR load disturbance rejection - Control signal, comparison between simulated and real signal

5.4 Temperature control

The cascade controller treated in Cap. 4 was tested only in a simulation. Due to technical problems it was not possible to test it in the real plant. To solve this problem the simulations are done with real data for the load disturbances. To make simulations closer to the truth, the data were measured on two different days, a sunny day and a cloudy one. This is useful because the solar irradiation is the most important element in this temperature control, and the presence of the clouds makes it erratic. Having said that, the simulation was made twice for this two different meteorological conditions. The sunny day data were collected on 20/07/2018 and the cloudy ones on 14/02/2018. In this way it is possible to test the controller with summer and winter data. This is very useful because

the climate changes a lot with the seasons.

Nomenclature:

- Test 1: sunny day named 'Day 1' - 20/07/2018
- Test 2: cloudy day named 'Day 2' - 14/02/2018

For this simulations it was used the same PID tuning. In a first phase, various tuning rules have been used but only the best ones are shown in this thesis. The tuning rule used for the inner loop (flow control) is C.H.R. for the set point following task. The external loop (temperature control) is tuned with Kappa Tau $M_s = 1.4$. The PIDs coefficients value are shown in Tab. 5.10

Table 5.10: Cascade control tuning

	External loop		Inner loop	
	Kp	Ti	Kp	Ti
Loop 1	-2.4781	94.5136	0.4242	4.2416
Loop 3	-5.6166	118.6993	0.5885	4.9042
Loop 4	-5.6094	118.6993	0.4758	3.1721
Loop 5	-5.6094	118.6993	0.5487	4.5723

To make this simulation more similar to the reality, the set point values are chosen in accordance with the downstream system required for the MED system. It makes sense, put the set-point on the loop 1 different to the other because this loop can feed a different circuit. The loop 1 output water can not be therefore mixed with the other loops' water.

5.4.1 Load disturbances

Irradiation The irradiation values are collected on a sunny and a cloudy day. The shape of the solar radiation is shown in Fig. 5.17(a) Fig. 5.17(b).

In Fig. 5.17(a) the data collected during 'Day 1' are shown. In this case the graph shape does not present strong variation and this data will be used to test the controller in more relaxed conditions.

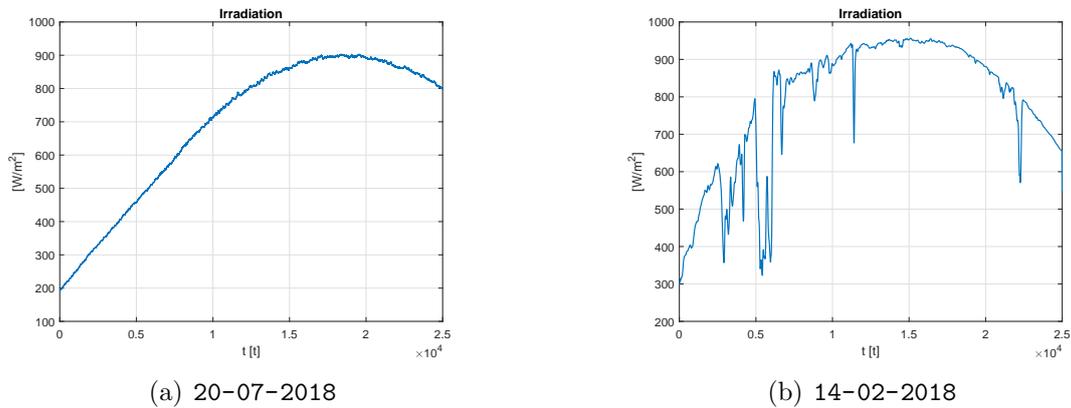


Figure 5.17: Irradiation data for the simulation

In Fig. 5.17(b) the presence of clouds is clear. In particular, from the trend of the graph can be seen strong changes in the radiation during the morning. This data will be used to test the controller in case of strong and unpredictable changes in the solar radiation.

Ambient temperature In Fig. 5.18(a) and Fig. 5.18(b) the data of the environmental temperature are shown.

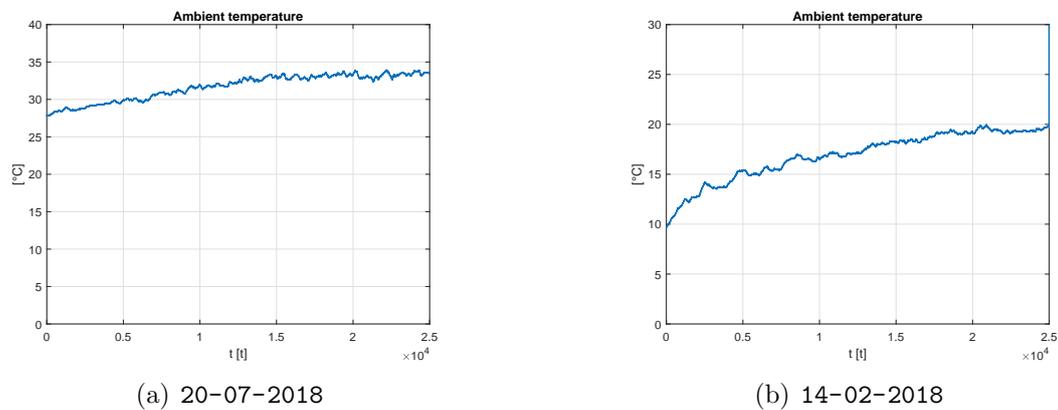
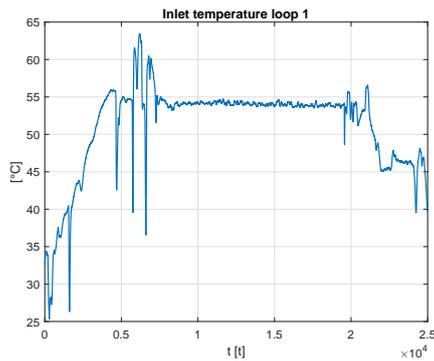


Figure 5.18: Ambient temperature data for the simulation

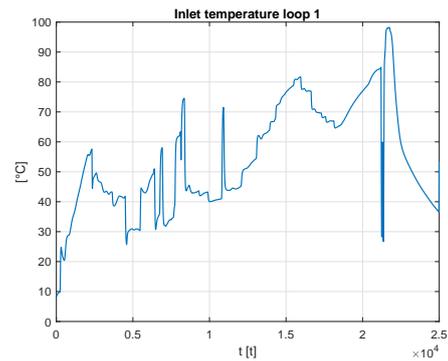
It stands to reason that the higher temperature was collected during the summer. The temperature shape is consistent with the irradiation data. It increases when comparing the morning with the afternoon. Besides, as it can be seen in Fig. 5.18(b), the temperature measured during the winter is not stable and presents some sharp changes in the morning

according with the irradiation values.

Inlet temperature It is the water inlet temperature in the various loops of the solar field. The values are a little different for each loop despite the fact that the water comes from the same tank.

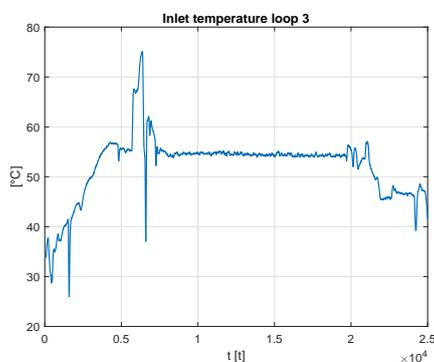


(a) 20-07-2018

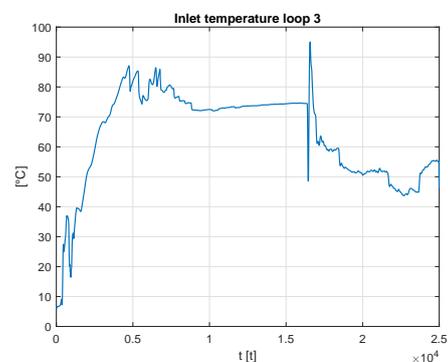


(b) 14-02-2018

Figure 5.19: Inlet temperature loop 1 data for the simulation



(a) 20-07-2018



(b) 14-02-2018

Figure 5.20: Inlet temperature loop 3 data for the simulation

The inlet temperature depends on many operating conditions of the plant. In particular the main influencers are the air cooler and the tank's heat exchanger. In fact, the water in the field can circulate without being used to heat the tanks. In this case the temperature in the field will increase. The loop 1 can be used to heat the water in a different circuit.

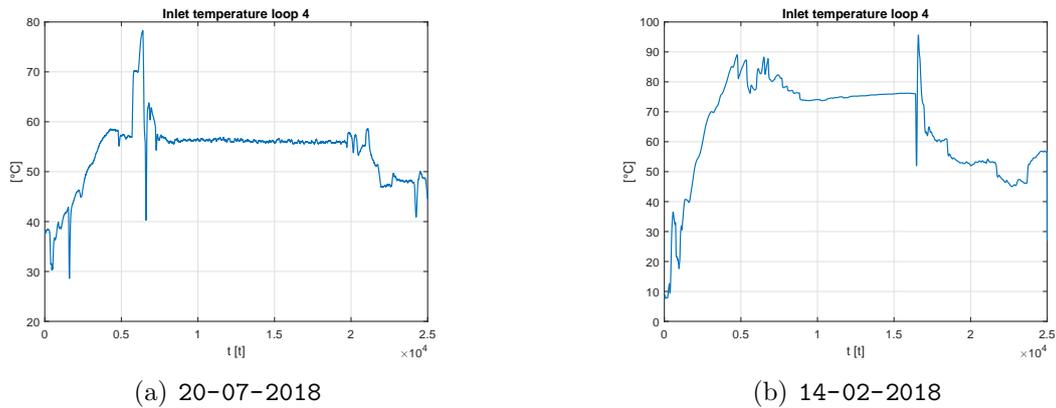


Figure 5.21: Inlet temperature loop 4 data for the simulation

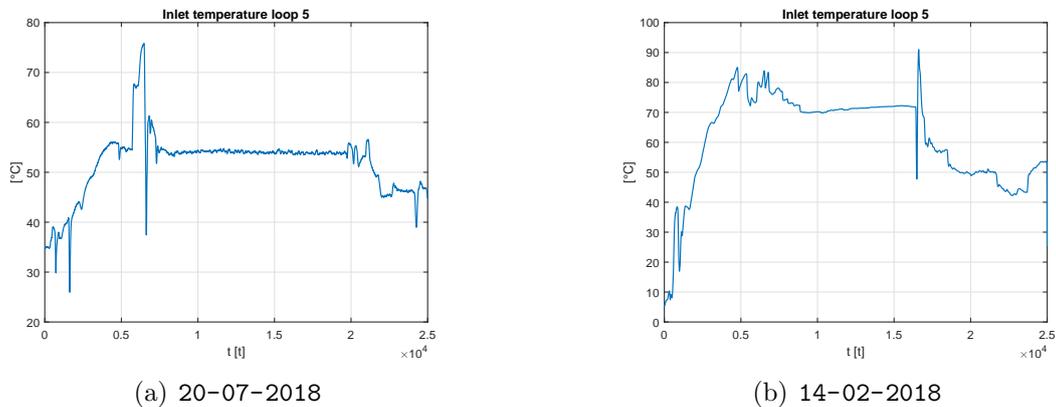


Figure 5.22: Inlet temperature loop 5 data for the simulation

This is the reason because in Fig. 5.19(b) The loop 1 inlet temperature is different from the other loops inlet temperature. Some times the inlet temperature is lowered to a constant value. The air cooler activation is the reason of it. we can say that we are not interested in all of these operating condition because they are incorporated in the inlet temperature.

5.4.2 Test 1

Test 1 is done with the data collected on 20-07-2018. The variable conditions during this day are advantageous because they present less sharp changes.

In Fig. 5.23, Fig. 5.24, Fig. 5.25 and Fig 5.26 the loops simulations are shown. During

the early hours of the day the irradiation is not high enough to raise the temperature up to the set point. We can see, however, that the control takes place in the correct way. All the pumps work in order to maintain the minimum flow values. In this initial phase the loop 1 has a flow equal to 7 and the other loops have a flow equal to 10. This corresponds to what it was expected since they are the minimum flow values imposed and the pump must allow these values to fall below this level. In Fig. 5.23, at seconds 4397 the temperature exceeds the set-point for a short time. The controller works in a proper way given the fact that the pump speed increase (and the flow increase too). In the previous period the flow set point was in saturation and, at this point, the correct operation of the anti windup also occurs. The same situation can be seen in the loop 3 at seconds 3500, loop 4 at seconds 4125 and in the loop 5 at seconds 3452. During the second part of the day the temperature achieved the set-point due to the high irradiation. In this case the pumps increase the flow in order to keep the temperature constant. It should be noted the behavior of the pump 1. Its speed decreases and the flow increases. This is caused by the interaction with the other pumps. The control signal can be considered satisfactory since it does not present many oscillations and where it is possible the set point value is maintained.

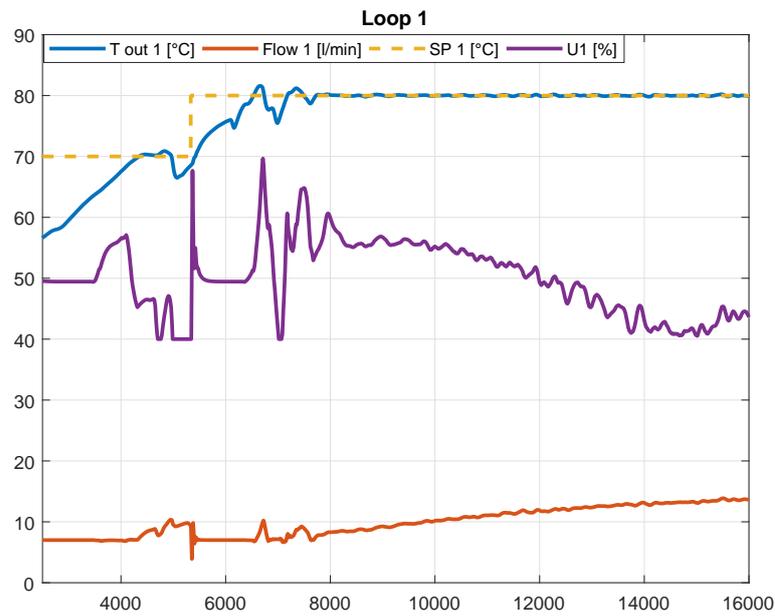


Figure 5.23: Simulation 1 - temperature control loop 1

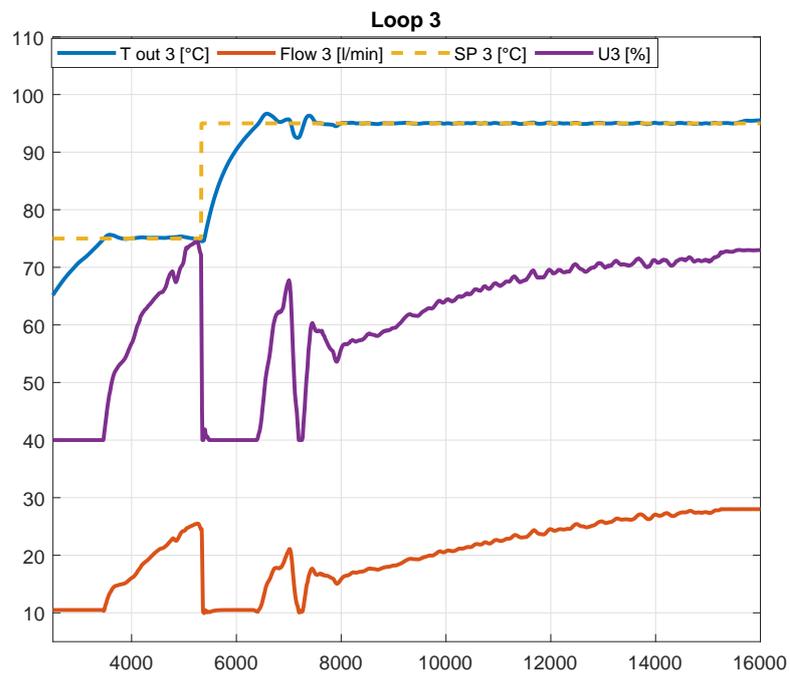


Figure 5.24: Simulation 1 - temperature control loop 3

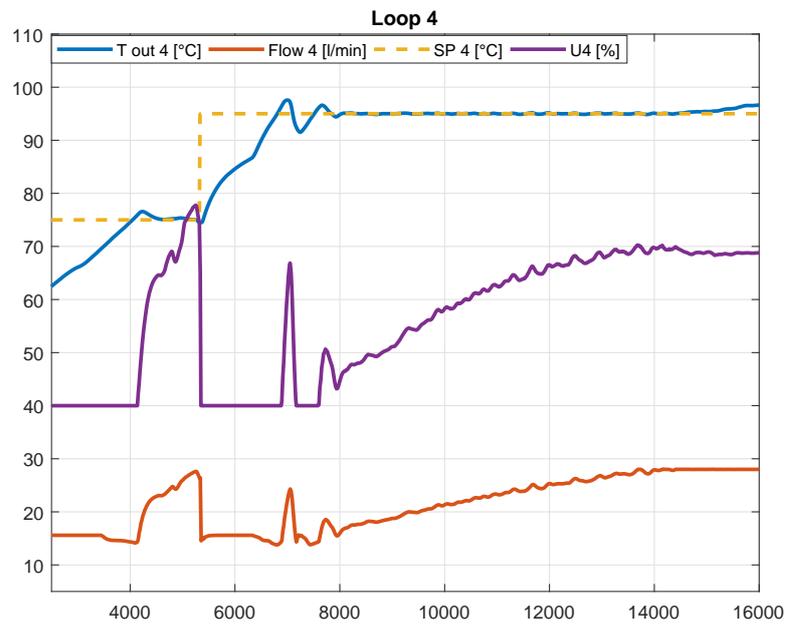


Figure 5.25: Simulation 1 - temperature control loop 4

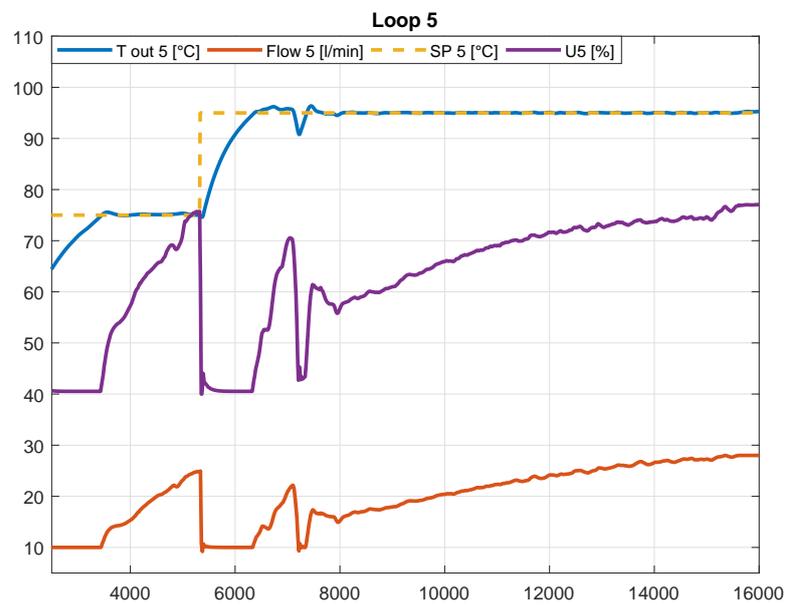


Figure 5.26: Simulation 1 - temperature control loop 5

5.4.3 Test 2

Test 2 is done with the data collected on 14-02-2018. This day was a cloudy day and the irradiation suffered rapid changes so it was the perfect opportunity to test the controller in non-ideal conditions.

In Fig. 5.27, Fig. 5.28, Fig. 5.29 and Fig. 5.30 the simulation results are shown. The set-point values are the same used in Test 1. It is clear from the very beginning that in this situation the loop 1 doesn't reach the set-point. Within the flow and the pump limits, it tries to get the set point value in a proper way. After the early stage of the simulation, loop 3,4 and 5 reach the set point value. In the first part of the simulation, it can be seen that the controller is not able to handle strong variations in the irradiation. At the same time, in the second part of the simulation the controller manages the disturbances correctly.

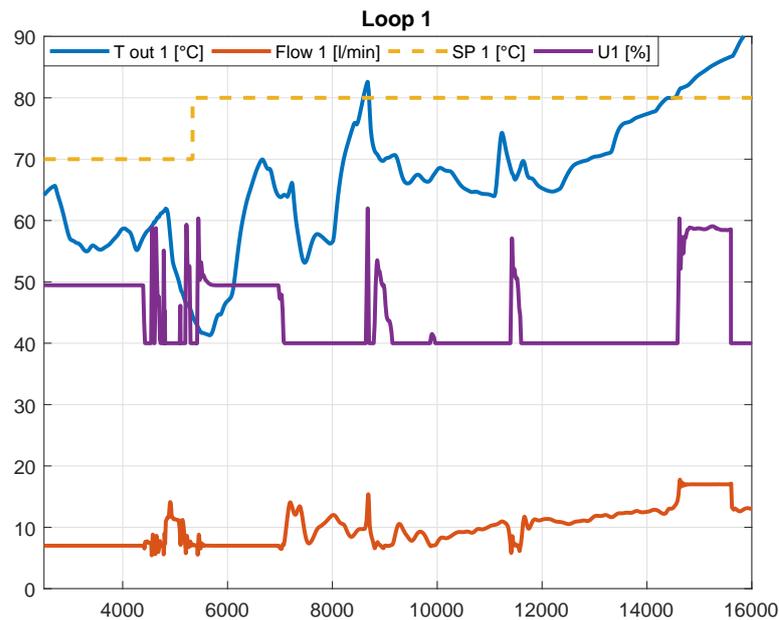


Figure 5.27: Simulation 2 - temperature control loop 1

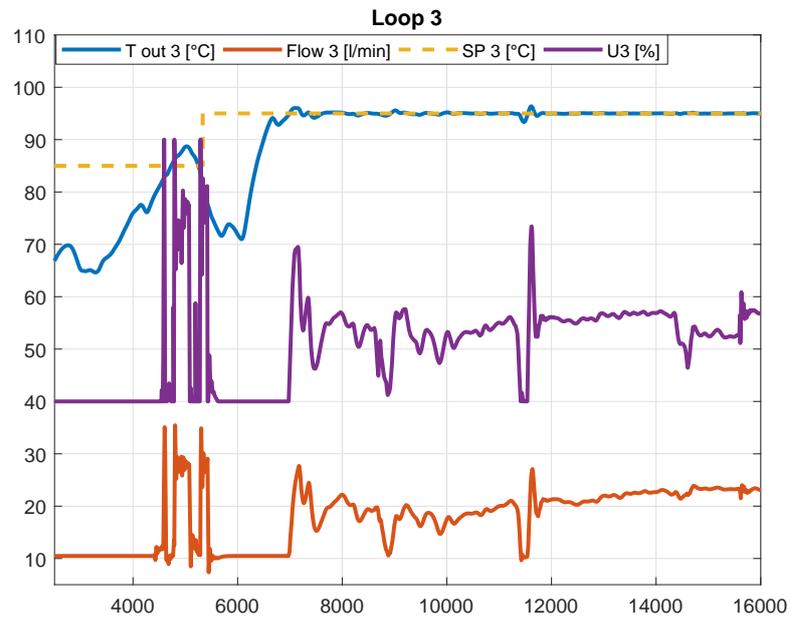


Figure 5.28: Simulation 2 - temperature control loop 3

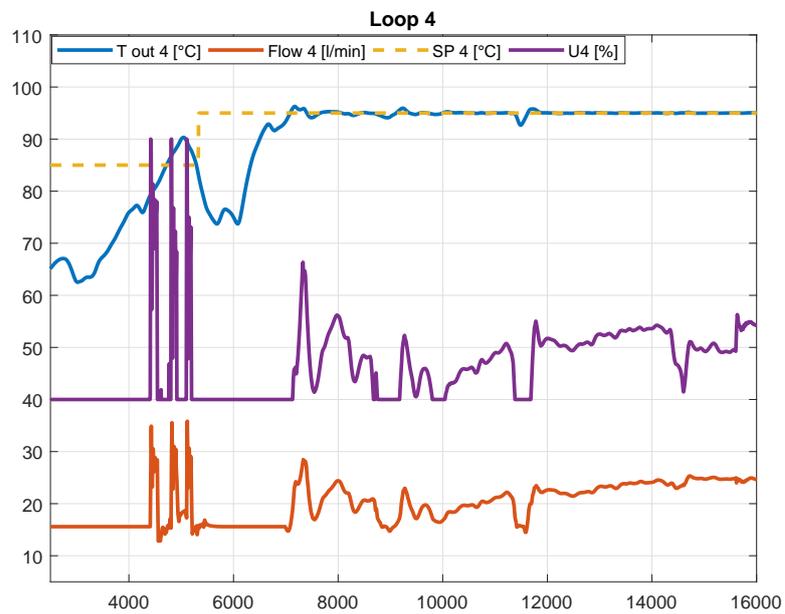


Figure 5.29: Simulation 2 - temperature control loop 4

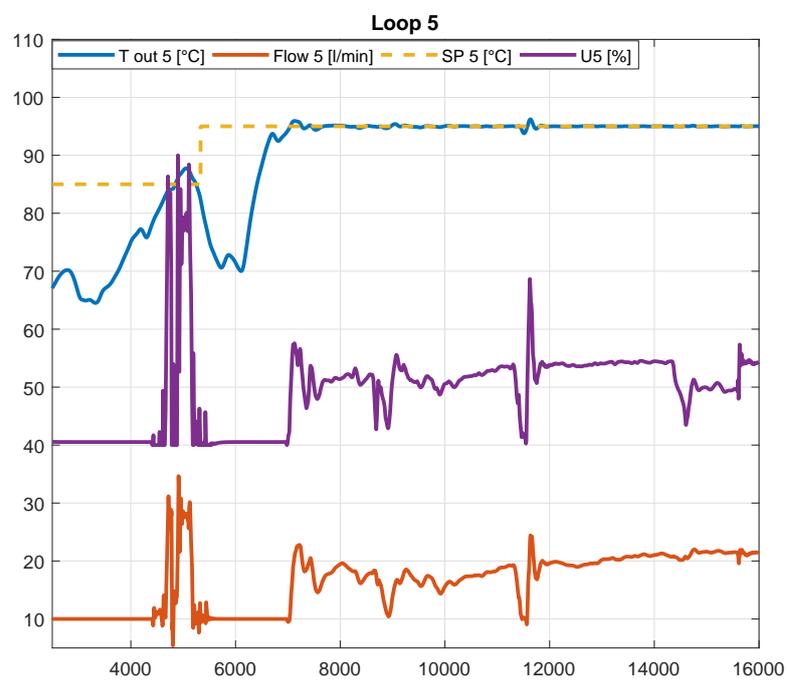


Figure 5.30: Simulation 2 - temperature control loop 5

Conclusions and future works

The controllers developed in order to control the flow and the temperature in the solar field loop have provided satisfactory results. The purpose of this thesis was to control a non-linear system (and no-minimum phase) with PIDs. As it is well known, PIDs are linear controllers and for this reason they are not suitable for controlling systems such as the one examined here.

First of all, two models (gain scheduling and non-linear approach) were designed to test in simulation the controller. Their validity was test with a comparison with the real hydraulic system data. In the first phase of the project, the flow control was developed. This part was the most laborious. It was necessary to decoupling the loops and solve the non-linearity and the non-minimum phase problems. Two channels were designed, one with the gain scheduling approach and one with a non linear PIDs's signal conditioning. With these two approaches, both the model and the associated controller were planned. For both of them the various controller tuning rules were tested and the results were shown in Cap. 5.2 and Cap. 5.3. However, the non-linear controller has a big advantage: it have not a switch system and the control action is therefore calculated continuously without jumps. For this reason it was used into in the internal loop of the temperature cascade control.

With the aim of controlling the temperature, a cascade control was designed. The external part takes the temperature as set-point, and it provides the flow set point to the internal part. In order to control the disturbance in the temperature in a proper way the disturbance in the temperature part, a feed forward controller was included. The FF action was calculated starting from the differential equations that describe the dynamics of the system.

Finally, the controller was tested in a simulation with the real disturbance data.

An auto tuning system should be developed in the future. In fact, the system changes greatly its parameters according to the water temperature. The implementation of a calibration every time the temperature changes could increase the performance of the system.

Appendix A

LabVIEW interface

At PSA, the installations communicate with staff through the LabVIEW interface. The data and the control signals travel on an internal network and to test the controllers it is therefore necessary to configure the LabVIEW interface with that networks.

All the control systems developed in this thesis were done in MatLab. In order to be able to test the controllers in the field, it was necessary to develop a LabVIEW interface. The controller could be developed directly in LabVIEW but in this way it is more likely to make some error in the translation.

The interface developed for the flow controller in Cap. 5.3 will be presented below. It is however possible to use it to configure any controller developed in MatLab.

It is divided in three blocks, where:

1. the old variables and the data are deleted from the MatLab Workspace
2. all the constants and variables required are initialized
3. the control action is performed.

A.1 Front Panel

As it can be seen in Fig. A.1, in the Front Panel there are all the user controls on the left and the variable graphs on the right. These graphs are important for checking the

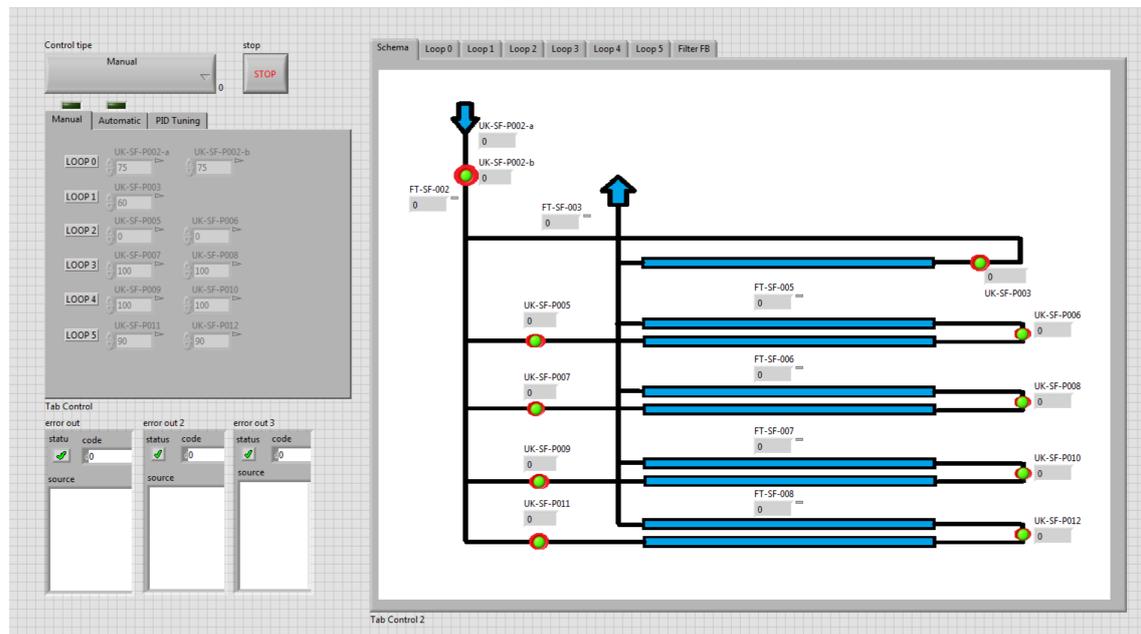


Figure A.1: LabVIEW - Front Panel

controller run time. We can see the temporal evolution of the variables and understand if the system is diverging. This is important for security issues.

In the right side the commands are managed. We can find:

- **Manual/Automatic switch** : This is one of the most important interface control tasks. For safety reasons, the test must always start in manual mode. In a second stage, it is possible to pass to the automatic mode. Moreover, it must always be possible to go back to the manual mode.
- **Stop** : this button is obviously used to stop the system. It should be noted that this button has no immediate effect. Its status is checked every 'while' loop cycle. The reaction time of the stop button therefore depends on the duration of the 'while' cycle (sampling time).
- **Control commands** : under the Manual/Automatic switch there is the control command window. It consists of three sections.

The first section, 'Manual', it is activated in manual mode. From here, it is possible

to manually set the operating point of the various pumps.

The second section, 'Automatic', it is activate in the automatic mode. In this case it is possible to set the flow set point, the PIDs tuning and the filter on the feed back action.

The third section, called 'PID Tuning', gives us the possibility to change the PID tuning run time.

- **Graph** : the graph can be found in the right side of the panel. It is present the general view of the plant and the variable evolution of the variables of all the loops.

A.2 Block diagram

All the interface program is developed in the block diagram. It is divided in three part and each of them is a succession of tasks.

1. In the first one it is necessary to clean the MatLab work space. The old variables and constants must be removed from there.
2. The second part corresponds to the initialization. Here the PIDs parameters are calculated and the use of the decoupling structure is established. Another very important aspect of this part is the Simulink initialization. The controller is developed in Simulink and, in order to speed up the future calls to the program, it is launched a first times.
3. In the third part (Fig. A.2, Fig. A.3), the main part of the code is included. here they are managed:
 - the change from the manual mode to the automatic mode. The change must take place without jumps in the actuators. In particular, when the change from the manual to the automatic mode occurs, the status vector of the Simulink controller must be initialized appropriately.

- the control in manual mode. The values set in the front panel must be sent to the actuators
- the control in automatic mode. The simulink controller must be put in run and the status vector must be updated in every sampling instant.
- The front panel controllers are disabled when they are not required in the current controller mode (manual/automatic)
- in this part are managed also all the front panel graphic signals.

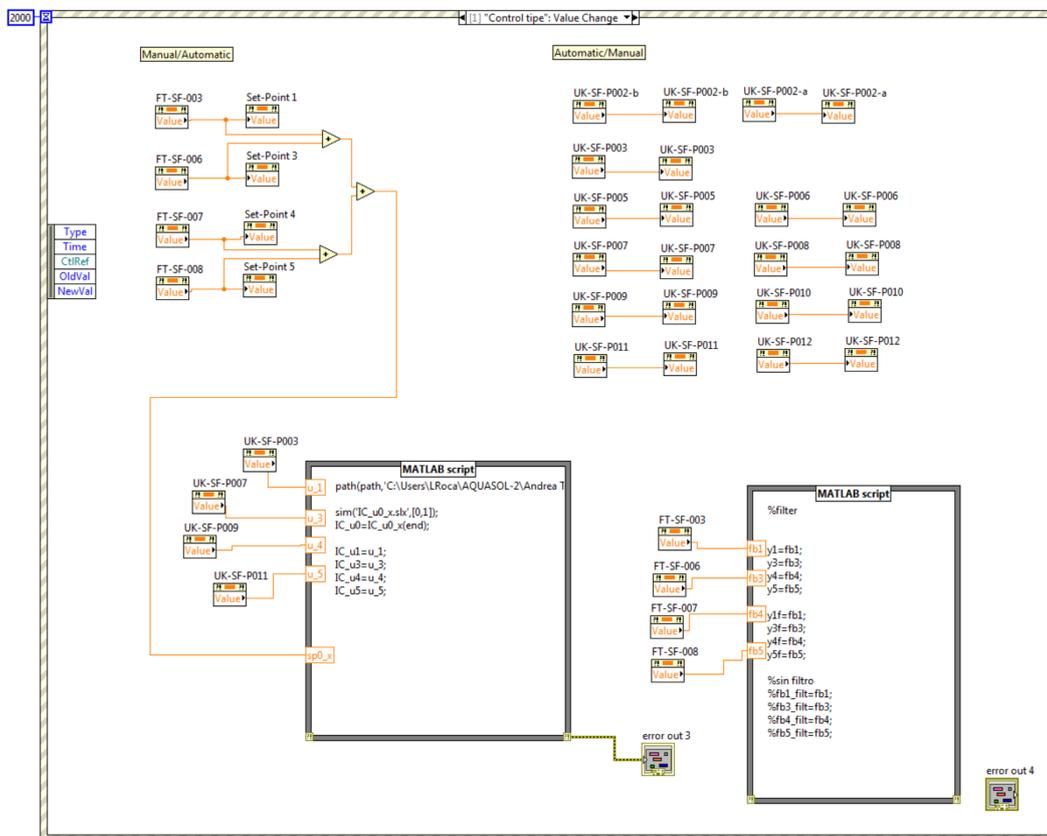


Figure A.2: LabVIEW - Block diagram - pt.3.1

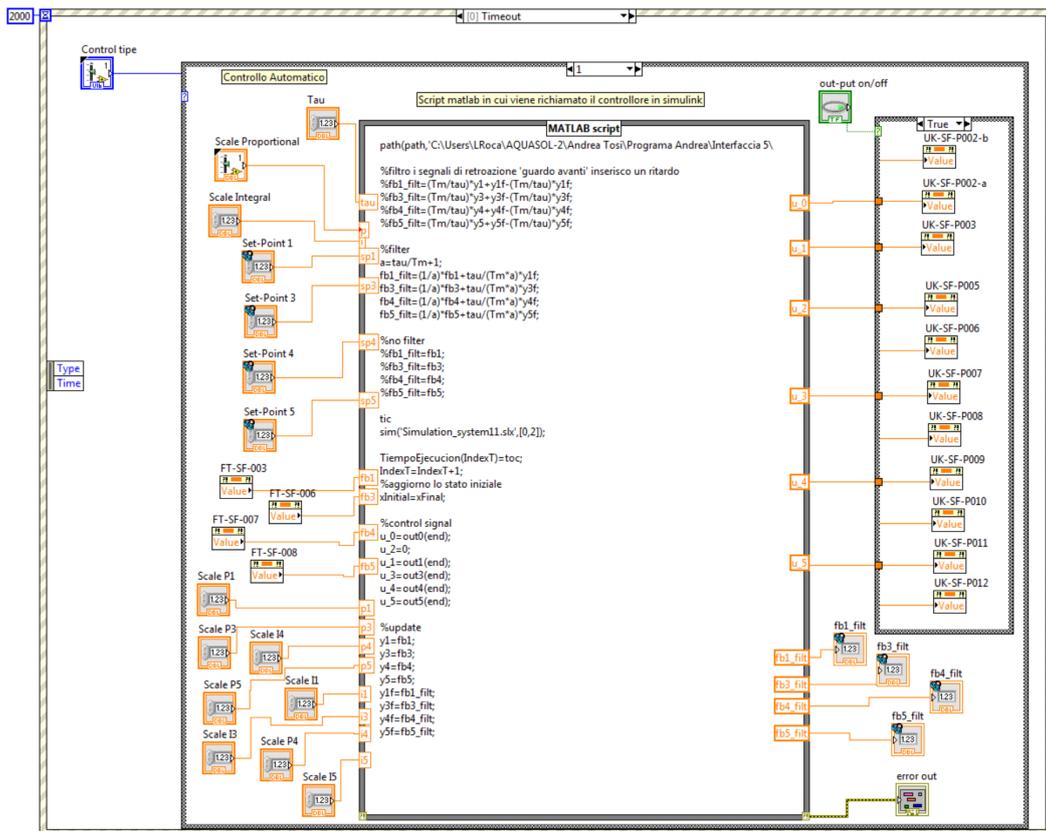


Figure A.3: LabVIEW - Block diagram - pt.3.2

Bibliography

- [1] J. D. Gil, L. Roca, G. Zaragoza, and M. Berenguel. A feedback control system with reference governor for a solar membrane distillation pilot facility. *Renewable Energy*, 120:536–549, 2018.
- [2] S. Skogestad and I. Postlethwaite. *Multivariable feedback control: analysis and design*. Chichester: John Wiley & Sons, 2001.
- [3] F. Morilla, J. Garrido, and F. Vázquez. Multivariable control by decoupling. *Revista Iberoamericana de Automática e Informática industrial*, 10:317, 2013.
- [4] S. J. Navas, F. R. Rubio, P. Ollero, and J. M. Lemos. Optimal control applied to distributed solar collector fields with partial radiation. *Solar Energy*, 159:811–819, 2018.
- [5] J. Vergara-Dietrich, J. Normey-Rico, L. Roca, and M. Berenguel. Controle de temperatura em campos solares de grande porte utilizando a abordagem do pnmpc - practical nonlinear model predictive control. In *Congresso Brasileiro de Automática CBA-2016*, 2016.
- [6] G. Ampuño, L. Roca, M. Berenguel, J. D. Gil, M. Pèrez, and J. E. Normey-Rico. Modeling and simulation of a solar field based on flat-plate collectors. *Solar Energy*, 170:369–378, 2018.
- [7] K. J. Åström and B. Wittenmark. *Adaptive control*. New York: Dover Publications, 2008.
- [8] F. R. Rubio E. F. Camacho, M. Berenguel and D. Martínez. *Control of solar energy systems*. Springer Science & Business Media, 2012.
- [9] J. D. Gil, L. Roca, G. Zaragoza, and M. Berenguel. A feedback control system with reference governor for a solar membrane distillation pilot facility. *Renewable Energy*, 120:536–549, 2018.



UNIVERSITÀ
DEGLI STUDI
DI BRESCIA

UNIVERSITÀ DEGLI STUDI DI BRESCIA
UNIVERSIDAD DE ALMERÍA



Resumen/Abstract

Located in the Taberna desert, thirty kilometers north of Almería, there is the PSA. The Plataforma Solar de Almería - PSA is the biggest research center on concentrated solar energy in Europe. Here a MED water desalination unit is located. This thesis is focused on the control of the hydraulic circuit in the stationary flat plate solar collectors. This part of the system is called AQUASOL-II. With the aim of tuning a controller to regulate the water flow rates inside the solar loops, two hydraulic models of the system have been experimentally obtained. During the experimental campaign performed to obtain the transfer functions models, it was observed that the static gain of the process change with the operating conditions due to the non-linearity of the actuators (pump). In order to deal with this non-linearity two system models were designed. The first using the gain scheduling principle (piecewise linear model) and the second composed by a linear and a non-linear part. Two different control systems are planned for the two different models. The main purpose of the control is decoupling the loops. In order to impose the required flow in each pipe. It is treated the flow control with the aim of obtaining the internal loop of a temperature cascade control.

Doble Título UNIBS-UAL

Mechatronics for Industrial Automation

Ingegneria dell'Automazione Industriale

Grado en Ingeniería Electrónica Industrial

Curso 2018/2019