

## *Resum*

En aquest Treball de Fi de Grau es realitza una introducció al llenguatge Modelica amb l'estudi de dos casos: un motor de contínua i el *continuous stirred tank reactor* (CSTR).

El primer cas es realitza com a introducció al llenguatge de modelització i estarà regulat mitjançant un controlador proporcional-integral (PI).

Amb el segon cas es realitza l'estudi de quatre tancs CSTR els quals es regulen mitjançant un controlador òptim per mantenir les respectives concentracions a unes referències desitjades. Aquest segon cas d'estudi està dividit en tres problemes: un problema d'una entrada i una sortida, un problema múltiple entrades única sortida i una distribució dinàmica de recursos.

Aquests dos problemes s'implementen mitjançant dues eines que utilitzen Modelica: OpenModelica i JModelica.org per al primer i segon cas d'estudi respectivament.

El treball té un objectiu netament introductori en quant es vol demostrar, de manera pràctica, la manera com dos sistemes reals es modelen a partir de les lleis físiques que regeixen el seu comportament el qual implica certa complexitat en l'adequació i tractament del model dinàmic. Adicionalment, s'implementen controladors òptims senzills que regulen els sistemes per complir els objectius de control desitjats.

# *Agraïments*

Primer i com a més important, m'agradaria agrair sincerament al meu director del treball, el Dr. Carlos Ocampo-Martínez, pel seu esforç i dedicació. El seus coneixements, orientacions, persistència i motivació han estat fonamentals per a la consecució d'aquest treball.

També agrair al Julián Barreiro-Gómez que sense estar obligat a participar, ha estat en tot moment disponible i gràcies al seu treball desconsiderat aquest treball ha pogut ser realitzat.

Finalment a la meva família que encara no estar a prop han estat un suport fonamental durant aquests anys d'estudis.

# Índex

<b>Resum</b>	<b>i</b>
<b>Agraïments</b>	<b>ii</b>
<b>Llista de Figures</b>	<b>v</b>
<b>Llista de Taules</b>	<b>vii</b>
<b>Símbols</b>	<b>ix</b>
<b>1 Introducció</b>	<b>1</b>
1.1 Motivació . . . . .	1
1.2 Objectius del projecte . . . . .	2
1.2.1 Objectiu principal . . . . .	2
1.2.2 Objectius específics . . . . .	2
1.3 Abast del projecte . . . . .	2
<b>2 Estat de l'Art</b>	<b>3</b>
2.1 Llenguatges de modelització . . . . .	3
2.1.1 Basats en gràfics versus basats en equacions . . . . .	3
2.1.2 Temps continu versus temps discret . . . . .	4
2.1.3 Multi-domini versus únic-domini . . . . .	5
2.1.4 Funcionals versus orientat a objectes . . . . .	5
2.2 Control . . . . .	5
2.2.1 Controladors PID . . . . .	6
2.2.2 Control òptim . . . . .	6
<b>3 Modelica i Entorns de Simulació</b>	<b>9</b>
3.1 Antecedents . . . . .	9
3.2 Introducció a Modelica . . . . .	9
3.3 OpenModelica . . . . .	10
3.4 JModelica.org . . . . .	11
3.5 Estructura d'un model en Modelica . . . . .	11
<b>4 Cas d'Estudi 1: Motor de Contínua</b>	<b>13</b>
4.1 Model matemàtic . . . . .	14
4.2 Implementació amb Modelica . . . . .	15

4.3	Disseny i implementació en simulació del controlador PI . . . . .	17
4.4	Simulacions i resultats . . . . .	19
4.4.1	$K_p = 0.5, T_i = 5$ . . . . .	21
4.4.2	$K_p = 0.5, T_i = 10$ . . . . .	23
4.5	Conclusions . . . . .	23
<b>5</b>	<b>Cas d'Estudi 2: Reactor Contínuament Agitat (CSTR)</b>	<b>25</b>
5.1	Descripció del sistema . . . . .	25
5.2	Model matemàtic en Modelica . . . . .	27
5.3	Problema de control . . . . .	28
5.4	Disseny del llaç de control . . . . .	29
5.5	Tipus de models . . . . .	30
5.5.1	Model de simulació . . . . .	30
5.5.2	Model d'optimització . . . . .	31
5.6	Models en Modelica/Optimica . . . . .	32
5.6.1	Punt inicial . . . . .	32
5.6.2	Model de simulació . . . . .	33
5.6.3	Model de control . . . . .	34
5.7	Escenaris de control . . . . .	35
5.7.1	Escenari 1: Cabal d'alimentació com a variable de control . . . . .	35
5.7.2	Escenari 2: Cabal d'alimentació i refrigeració com a variables de control .	37
5.7.3	Distribució dinàmica de recursos . . . . .	40
5.8	Conclusions . . . . .	42
<b>6</b>	<b>Anàlisi Econòmic</b>	<b>45</b>
<b>7</b>	<b>Impacte Ambiental</b>	<b>47</b>
<b>8</b>	<b>Conclusions i Treball Futur</b>	<b>49</b>
8.1	Conclusions . . . . .	49
8.2	Treball futur . . . . .	50
<b>A</b>	<b>Planta Laboratori</b>	<b>51</b>
<b>B</b>	<b>CSTR</b>	<b>53</b>
	<b>Bibliografia</b>	<b>57</b>

# Llista de Figures

2.1	Modelat Acausal . . . . .	4
2.2	Modelat Causal . . . . .	4
4.1	Esquemàtic del motor de contínua . . . . .	14
4.2	Resposta transitòria de la velocitat angular del motor de contínua . . . . .	16
4.3	Resposta transitòria de la intensitat del motor de contínua . . . . .	16
4.4	Modelització física del motor de contínua utilitzant la <i>Modelica Standard Library</i> . . . . .	16
4.5	Modelització Causal Motor de Contínua . . . . .	17
4.6	Llaç de control . . . . .	17
4.7	Resposta transitòria per $T_i = 999$ i per a diferents valors de $K_p$ . . . . .	19
4.8	Resposta transitòria per $K_p = 0.5$ i per a diferents valors de $T_i$ . . . . .	20
4.9	Resposta transitòria del model per al motor de contínua reduït . . . . .	22
5.1	Esquema del CSTR . . . . .	26
5.2	Esquema del problema de control proposat . . . . .	28
5.3	Llaç de Control de la Optimització . . . . .	29
5.4	Pla de fase del CSTR i punt d'equilibri inicial . . . . .	32
5.5	Concentracions per a l'Escenari 1 . . . . .	35
5.6	Variable de Control, $q_i$ , per a l'Escenari 1 . . . . .	36
5.7	Pla de fase CSTR <sub>4</sub> . . . . .	37
5.8	Concentracions Escenari 2 . . . . .	38
5.9	Cabal d'alimentació Escenari 2 . . . . .	39
5.10	Cabal de refrigeració Escenari 2 . . . . .	39
5.11	Pla de fase CSTR <sub>4</sub> amb l'Escenari 2 . . . . .	40
5.12	Concentracions Problema de Distribució Dinàmica . . . . .	41
5.13	Cabals Problema de Distribució Dinàmica . . . . .	42
A.1	Funció de transferència del motor . . . . .	51
A.2	Contol Planta Laboratori . . . . .	52
B.1	CSTR mitjançant blocs MatLab/SimuLink . . . . .	53



# Llista de Taules

4.1	Paràmetres del model del motor de contínua . . . . .	15
4.2	Comparació de Resultats . . . . .	24
5.1	Paràmetres del CSTR i els seus corresponents valors . . . . .	26
5.2	Resultats de l'Escenari 1 després de 200 minuts de simulació . . . . .	36
5.3	Resultats de l'Escenari 2 després de 200 minuts . . . . .	40





# Símbols

$V_a$	Voltatge a l'induït	V
$\Gamma_e$	Parell del motor	Nm
$i_a$	Intensitat a l'induït	A
$E_b$	Força electromotriu	V
$\omega$	Velocitat angular	rad/s
$R_a$	Resistència a l'induït	$\Omega$
$L_a$	Inductància a l'induït	H
$K_T$	Constant del parell	Nm/A
$K_E$	Constant electromotriu	V s/rad
$J$	Inèrcia	Kg/m <sup>2</sup>
$u$	Acció de control	
$K_p$	Acció proporcional controlador PI	
$T_i$	Temps integral	s
$e$	Error	
$V$	Volum del CSTR	l
$q$	Cabal d'alimentació	l/min
$c_{Af}$	Concentració a l'alimentació	mol/l
$T_f$	Temperatura del fluid d'entrada	K
$q_c$	Cabal de refrigerant	l/min
$\rho, \rho_c$	Densitat del fluid d'entrada i refrigerant	g/l
$C_p, C_{pc}$	Calor específic del fluid d'entrada i refrigerant	cal/(gK)
$k_0$	Factor pre-exponencial	1/min
$E/R$	Factor exponencial	K
$\Delta H$	Calor de la reacció	cal/mol
$hA$	Característica de transferència de calor	1/(minK)

$c_{A,i}$	Concentració del producte de l' $i$ -èssim CSTR	mol/l
$\Delta t$	Període de mostreig	min
$MOC_i$	Model orientat a control de l' $i$ -èssim CSTR	

# Capítol 1

## Introducció

### 1.1 Motivació

L'elecció del tema del treball de fi de grau ve lligada a dos components: l'elecció del llenguatge utilitzat i el perquè de la modelització i simulació de sistemes dinàmics.

El llenguatge utilitzat és Modelica, el qual es tracta d'un llenguatge de simulació orientat a objectes i basat en equacions que no està sotmès a cap patent per la qual cosa és de domini públic. A més, darrerament han sortit diverses eines i entorns, també de caràcter públic, per a la simulació de models escrits en Modelica. Tot i estar, aquests entorns, sota desenvolupament, és interessant poder comprovar les seves capacitats ja que cada vegada es demana més software lliure i de qualitat.

El segon component ve lligat a la importància de la modelització i la simulació en l'entorn de la indústria. Aquestes eines permeten als enginyers provar els seus dissenys d'una manera virtual enfront de l'experimentació sense utilitzar eines d'alt cost i de domini exclusivament acadèmic.

Projectes de gran escala comencen a ser la regla i no la excepció. Un exemple d'aquests projectes és el projecte ITER (*International Thermonuclear Experimental Reactor*), l'objectiu del qual és la construcció d'un reactor de fusió nuclear a gran escala que permeti obtenir energia de forma eficient substituint així els reactors nuclear que actualment operen mitjançant la fissió.

Aquests projectes requereixen d'estudis previs a la seva construcció els quals es realitzen mitjançant tècniques de modelització, és a dir, la construcció de models matemàtics sobre els quals es realitza l'estudi amb el fi d'obtenir dades per entendre com es comporta el sistema real.

## 1.2 Objectius del projecte

### 1.2.1 Objectiu principal

La realització d'aquest projecte ha tingut com a objectiu principal utilitzar el llenguatge de modelització Modelica en l'estudi de dos problemes concrets els quals estan implementats amb dos controls distints, un controlador clàssic com és el proporcional-integral (PI) i un control òptim.

A més s'espera arribar a unes conclusions sobre les avantatges que suposa el llenguatge estudiat en la memòria en front a un dels entorns de modelització utilitzats a l'escola durant la realització del grau, e.g., MatLab.

### 1.2.2 Objectius específics

L'objectiu principal es pot desglossar en una sèrie d'objectius específics:

1. Fer una introducció en l'estudi de sistemes no lineals (segon cas d'estudi).
2. Estudiar els models matemàtics per a la modelització de sistemes reals.
3. Estudiar teoria de control no vista en la realització del grau per a la seva implementació (control òptim).
4. Realitzar el control dels models estudiats en l'objectiu específic 2, mitjançant entorns de Modelica especialitzats per a la seva realització.
5. Amb l'estudi realitzat, finalitzar amb una conclusió que reflecteixi la idoneïtat o no del llenguatge Modelica.

## 1.3 Abast del projecte

El present projecte és un estudi introductori del llenguatge Modelica aplicat a dos casos particulars:

- Motor de contínua implementat amb un controlador proporcional-integral (PI).
- Quatre *continuous stirred-tank reactor* (CSTR) els quals es controlaran aplicant un control òptim.

Els controls presentats a la memòria no venen discutits front a altres possibles sinó que únicament es justificarà la seva qualitat en front de si aconsegueixen acomplir les especificacions que es demanen.

## Capítol 2

# Estat de l'Art

En aquest capítol s'introdueix l'estat de l'art sobre dos temes: l'estat de l'art sobre els llenguatges de simulació i modelització i sobre les estratègies de control utilitzades en aquest treball.

### 2.1 Llenguatges de modelització

Nombrosos llenguatges de simulació han estat desenvolupats durant els darrers anys. En general es poden classificar d'acord als criteris següents: basats en gràfics versus basats en equacions, temps continu versus temps discret, multi-domini versus únic-domini i llenguatges funcionals versus orientats a objectes.

#### 2.1.1 Basats en gràfics versus basats en equacions

Els gràfics serveixen per representar sistemes interconnectats. Un exemple d'aquest paradigma de modelització és la modelització per blocs, com utilitza el SimuLink de MatLab. Els senyals són transmesos a través de la sortida d'un bloc a la entrada d'un altre. Les dades d'entrada són processades en els blocs obtenint-se dades de sortida.

Normalment els diagrames de blocs exposen l'estructura de càlcul més que el model real del sistema, és a dir, aquest paradigma de modelització sobreposa l'estructura de computació sobre la estructura del comportament del sistema.

En canvi, els llenguatges basats en equacions no fixen la causalitat del model, sinó que el model es crea establint un conjunt d'equacions que relacionen els estats. És el mateix software de simulació el que s'encarrega de convertir aquestes equacions en accions que poden ser evaluades pel computador.

El modelatge acausal també se'l denomina *modelatge físic* ja que és adient per representar l'estructura física dels sistemes que es vulguin modelar.

Per acabar de demostrar aquesta idea final, es pot realitzar un exemple de modelat a través dels dos paradigmes (exemple extret de [9]) mitjançant Modelica i SimuLink, Figures 2.1 i 2.2 respectivament.

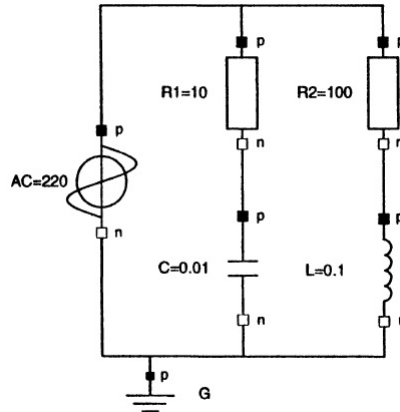


FIGURA 2.1: Modelat Acausal

El circuit de la figura 2.1 modelat acausalment conserva íntegrament l'estructura física del problema (modelat físic).

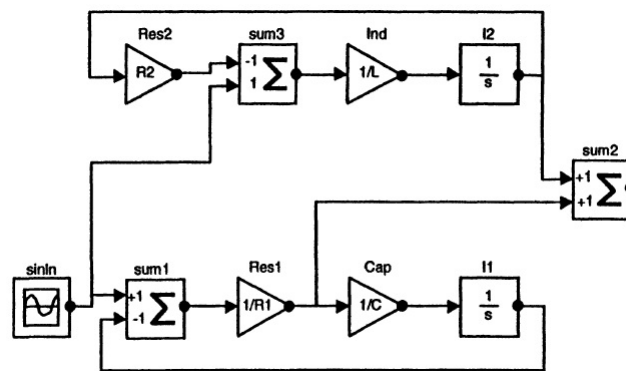


FIGURA 2.2: Modelat Causal

En canvi, si s'utilitza un paradigma de modelització causal queda evident (Figura 2.2) que la topologia física s'ha perdut completament i no és fàcil realitzar una assignació entre el diagrama de blocs i el model real. A més, un model inclòs un de tant senzill com el presentat no comporta una tasca de modelatge trivial.

### 2.1.2 Temps continu versus temps discret

Els processos físics es desenvolupen generalment en funció del temps, per la qual cosa sempre és millor realitzar un modelatge per un conjunt de equacions diferencials algebraiques (DAE per les seves sigles en anglès) en temps continu, mentre que els components digitals com controladors

es dissenyen i implementen en temps discret. Aquest fet genera l'aparició de llenguatges de simulació que combinen aquests dos dominis, el continu i el discret, és a dir els anomenats llenguatges híbrids. La introducció dels dos dominis comporta dificultats per a la resolució matemàtica però que queden sopesades per la avantatge que suposa en quant a les seves possibles aplicacions.

### 2.1.3 Multi-domini versus únic-domini

Modelitzacions i simulacions d'un únic domini com per exemple l'elèctric és un àrea madura. Existeixen moltes companyies que ofereixen software de qualitat per a la realització d'aquesta tasca. Ara bé, els projectes d'enginyeria d'avui dia es realitzen a gran escala, el que implica que moltes vegades hagin de coexistir elements de diferents dominis físics, com l'elèctric o el mecànic. Existeixen diferents llenguatges de simulació que ofereixen la possibilitat, mitjançant biblioteques, d'utilitzar o interconnectar sistemes de diferents dominis.

### 2.1.4 Funcionals versus orientat a objectes

Els llenguatges de modelització orientats a objectes utilitzen el paradigma de l'orientació a objectes per a la construcció de models. Aquest paradigma enfoca la construcció de models mitjançant un conjunt d'objectes relacionats entre si i que col·laboren entre si segons els requeriments establerts per al sistema d'objectes. Aquest tipus de modelatge comporta, entre d'altres, una característica pròpia de tot llenguatge orientat a objectes: l'herència. Aquesta característica permet utilitzar un model definit prèviament i incorporar noves característiques així com modificar algunes de les seves. Aquesta propietat permet construir models complexos sense la necessitat de definir el model com tantes vegades es necessiti (veure Secció 5.3).

Els llenguatges que es basen en el paradigma de la programació funcional impliquen que el problema s'ha de crear com un conjunt de funcions que es deuen executar. En aquest paradigma és necessari definir amb cautela les entrades a cada funció per saber amb certesa que en retornen. El paradigma de la modelització mitjançant blocs és un exemple de llenguatge funcional.

## 2.2 Control

Aquesta secció introdueix breument una descripció de l'estat actual dels controladors utilitzats en aquest treball, així com algunes referències bibliogràfiques on es detallen les seves característiques més rellevants.

### 2.2.1 Controladors PID

El controlador PID s'utilitza normalment en estructures tipus *feedback* (retroalimentació) (Figura 4.6) i és utilitzat en un ampli rang d'aplicacions industrials gràcies a la seva,relativament, fàcil implementació i la seva efectivitat. Aquest controlador és el més utilitzat en la indústria que requereix control de processos en la qual més del 95% són controladors PI o PID [17].

Encara ser el controlador més utilitzat, aquest sempre ha rebut un interès marginal de cara als investigadors de teoria de control. Ara bé, durant els darrers anys aquest control ha rebut més interès gràcies als avanços en la seva sintonització automàtica (*automatic tuning*) ja que simplifica encara més el seu ús [16]. Segons estudis [17], el 75% de tots els PID implementats en una estructura tipus *feedback* estan malament sintonitzats i fins i tot que el 25% s'utilitzen tal com venen de fàbrica sense haver-hi sintonitzat adequadament el controlador.

És difícil establir les línies en que s'està investigant més la utilització de PID, però en quant a la indústria de processos es podria dir que la recerca essencial s'ha dut a terme en la sintonització de PID per processos no autoregulables (processos que per canvis forçats, e.g., entrada graó, no arriben a una nova condició d'operació) que són inestables en llaç obert (més de 800 regles han estat desenvolupades en la darrera dècada).

### 2.2.2 Control òptim

El control òptim té com objectiu determinar les accions de control que satisfaran unes restriccions físiques i al mateix temps maximitzaran o minimitzaran un criteri de rendiment. En general, es vol aconseguir trobar unes accions de control  $\mathbf{u}^*(t)$  (l'asterisc indica òptim) que portaran al sistema des d'un estat inicial fins a un estat final a la vegada que minimitzen el criteri de rendiment  $J$ . Aquesta teoria de control es pot catalogar com a control modern ja que es diferencia del clàssic en que no es basa en la transformada de Laplace sinó que es basa en la representació dels estats del sistema mitjançant un conjunt d'equacions diferencials [14].

En la indústria, com s'ha comentat, el controlador més important és el PID, el controlador òptim té un major pes en el món de la robòtica. En el cas en què es vulgui portar un manipulador d'una configuració inicial a una final, normalment aquest tipus de moviments té distintes possibles solucions. És per això que s'implementa un control òptim que minimitzi un criteri en concret, e.g., el temps total del moviment.

En la indústria referent al control òptim està guanyant pes el control predictiu basat en models (MPC per les seves sigles en anglès). Aquesta tècnica de control es fa servir per processos multivariables sobretot del sector químic/petroquímic en el que mitjançant el model dinàmic del procés s'intenta predir el comportament futur del sistema. Aquest tipus de control requereix la resolució iterativa de problemes de control òptim. Aquest tipus de control normalment va acompanyat de controladors PID.



Normalment en la indústria s'utilitzen controladors lineals, linealitzant el model lineal d'un procés al voltant d'un punt d'operació i dissenyant el controlador. Si es realitza el disseny d'un controlador no lineal (NMPC) s'obté un problema de grans dimensions, el qual requereix un gran esforç de computació. Gràcies a l'evolució de la computació i nous mètodes com el de punts interiors per resoldre l'optimització, aquesta és una línia de recerca molt activa en els darrers anys.



## Capítol 3

# Modelica i Entorns de Simulació

### 3.1 Antecedents

Quan va néixer Modelica per l'any 1997 havia al mercat un gran nombre de llenguatges de simulació. La gran part d'aquests llenguatges estaven disponibles sota pagament i es basaven en el mateix principi: el paradigma de modelització mitjançant diagrama de blocs com el ACSL o SimuLink de MatLab. També existien llenguatges dedicats a la modelització d'un únic domini físic com el domini electrònic (SPICE), dinàmica del cos (ADAMS), processos químics (ASPEN Plus), etc. Tècniques de modelització física basades en procediments no causals van ser desenvolupats a partir dels anys 90 amb exemples de llenguatges com ASCEND, NMF, ObjectMath, fruit de la recerca en les universitats i del desenvolupament en petites empreses. Modelica va néixer amb el propòsit d'unificar els llenguatges ja existents sota una mateixa semàntica i crear un llenguatge de modelització per al modelat de sistemes físics.

### 3.2 Introducció a Modelica

Modelica és un llenguatge de modelització de sistemes dinàmics amb estats que evolucionen al llarg del temps. Modelica, a més, és un llenguatge orientat a objectes basat en equacions, la qual cosa el dota de gran capacitat per estudiar sistemes de gran complexitat que requereixen d'un gran rendiment. Les característiques més importants d'aquest llenguatge són:

1. Llenguatge basat en equacions
2. Capacitat de reutilització de classes
3. Varietat de camps de modelització que inclouen l'elèctric, mecànic, entre d'altres.
4. Llenguatge gratuït que compta amb diversos entorns també gratuïts.

A més de les característiques de tot llenguatge orientat a objectes, Modelica ofereix eines per establir les relacions entre els elements constitutius del model i la seva interacció. Modelica compta amb una sèrie de llibreries tant d'ús comercial com gratuïtes en les quals estan definits els principals components de diferents dominis físics. Ara bé, la utilització d'aquestes llibreries requereix de l'ús d'entorns de simulació els quals poden ser tant d'ús comercial com gratuïts:

- Comercials: Dymola, SimulationX, MapleSim, Wolfram SystemModeler, entre d'altres.
- Gratuïts: JModelica.org, OpenModelica, Modelicac

En l'elaboració d'aquest treball s'ha utilitzat l'entorn OpenModelica i el JModelica.org. El primer compta, per exemple, amb una interfície per treballar en blocs d'una forma més visual permetent tant una modelització física mitjançant llibreries com una modelització causal mitjançant blocs elementals i un altra interfície per escriure models escrivint el conjunt d'equacions que defineixen la seva dinàmica. El segon entorn, JModelica.org, s'utilitza per a la implementació de controladors òptims, simulacions i anàlisis de complexos sistemes dinàmics. Aquests dos entorns s'expliquen amb més profunditat a continuació.

### 3.3 OpenModelica

OpenModelica és un *open source* desenvolupat per la Linköping University de Suècia [4]. L'objectiu d'aquesta eina és crear un complet entorn de modelització, compilació i simulació, de codi lliure, basat en Modelica. Aquest entorn compta amb una sèrie de subsistemes relacionats entre si, de la manera següent:

- OMShell: interfície d'interacció amb el llenguatge. Aquesta eina no és utilitzada en aquest treball ja que s'utilitza en el seu cas el OMNotebook.
- Compilador de Modelica: tradueix el llenguatge Modelica a codi C.
- Mòdul d'execució en temps real: aquest mòdul executa el codi binari format en les traduccions de les expressions.
- OMEdit - OpenModelica Connection Editor: és una interfície gràfica per a l'edició de models. Aquesta eina permet la creació de models dinàmics mitjançant la utilització de llibreries o blocs (eina molt parecida al SimuLink). A més et fa una traducció de blocs a llenguatge Modelica en temps real.
- OMNotebook: és el quadern interactiu de programació de l'entorn OpenModelica. Permet tant la programació de models en llenguatge Modelica com la simulació d'aquests models. Incorpora també un tutorial anomenat *Dr. Modelica* que tutoritza l'ensenyament del llenguatge i teoria control.

- OMOptim: subsistema per al desenvolupament de problemes d'optimització.
- Depurador de models d'equacions en Modelica: notifica la localització d'errors en el model executat.

Aquest software es troba encara sota desenvolupament, en el qual s'està intentant millorar la compilació, així com la capacitat per realitzar problemes d'optimització dinàmica, entre d'altres tasques. La versió utilitzada durant aquest projecte és la 1.9.1 en estat de *beta 4*. Al moment de finalitzar aquest projecte ha estat llançada la versió 1.9.3. Per més informació consultar [5].

### 3.4 JModelica.org

Per a la implementació del control òptim, s'ha optat per utilitzar un altre entorn de simulació molt més orientat a la resolució d'aquest tipus de problemes: JModelica.org [3].

JModelica.org és un software dedicat a la implementació de problemes d'optimització dinàmica basats en llenguatge Modelica. Modelica es va dissenyar com a llenguatge de simulació i no d'optimització, fet que es va arreglar desenvolupant una extensió del mateix llenguatge anomenada Optimica [6]. Optimica afegeix nova semàntica al llenguatge Modelica otorgant la possibilitat de declarar restriccions i funcions objectius d'una manera senzilla però de gran capacitat de resolució.

Una de les característiques d'aquest entorn és que la interacció programa-usuari es realitza a través del llenguatge de programació Python, el qual dona suport a través dels seus paquets tant a la computació numèrica com a la graficació de resultats. A més, Python és també un llenguatge de codi lliure el qual s'ha treballat durant la realització dels estudis del grau en enginyeria en tecnologies industrials.

### 3.5 Estructura d'un model en Modelica

Es pot realitzar un model en Modelica amb modelització d'alt nivell o baix nivell. La modelització d'alt nivell consisteix en la interconnexió de blocs (cada bloc té el seu model definit) mentre que la de baix nivell consisteix en escriure el programa en llenguatge Modelica. En aquesta memòria s'utilitza en quasi la seva totalitat el baix nivell.

Els programes a Modelica es construeixen mitjançant classes, anomenades en aquest entorn, models. Una vegada definida una classe, es poden realitzar instàncies d'aquesta classe per tal de formar programes més complexos. Tot programa de Modelica conté seccions on es declaren variables i seccions on s'escriuen les equacions que descriuen el comportament de la classe.

El programa comença amb la capçalera on es defineix el tipus de classe que s'utilitza. En aquest projecte únicament es fa servir la capçalera `model` per descriure les classes necessàries menys

quan es vol definir una optimització, per la qual cosa s'utilitzarà la capçalera `optimization` [6]. Tot seguit, es defineixen els paràmetres i les constants necessàries que es necessiten per descriure el funcionament del sistema.

Els paràmetres són quantitats que romanen constants durant una simulació però que poden canviar d'una simulació a un altra. En canvi, les constants són quantitats que sempre seran constants, sense possibilitat de canvi d'una simulació a un altra. Aquestes quantitats es defineixen mitjançant `parameter` i `constant`, respectivament.

Finalment es declaren les variables. Aquestes quantitats no requereixen de cap tipus d'instància per ser declarades, ja que en Modelica es considera variable tota aquella magnitud que es funció del temps. Acabat tot aquest procés, es comencen a introduir les equacions que regeixen el comportament del sistema. En el model és necessari introduir la capçalera `equation` per començar a declarar les equacions.

Per entendre millor l'estructura d'un programa en aquest llenguatge, es posa un exemple extret de [9], llibre que s'ha utilitzat com a guia per aprendre el llenguatge.

El programa simula el model del comportament de l'oscil·lador de Van der Pol que segueix aquesta dinàmica (la definició de les variables i constants utilitzades en aquest exemple es poden consultar a [9]):

$$\ddot{x} - \lambda(1 - x^2)\dot{x} + x = 0. \quad (3.1)$$

El seu corresponent codi en Modelica és el següent:

---

```

model VanDerPol "Van der Pol oscillator model"
  Real x(start = 1) "Descriptive string for x"; \\S'inicialitza la x
  Real y(start = 1) "Descriptive string for y"; \\S'inicialitza la y
  parameter Real lambda = 0.3;
equation
  der(x) = y;
  der(y) = -x + lambda*(1 - x*x)*y;
end VanDerPol;

```

---

Cal notar que hi ha la necessitat d'inventar un estat `y` ja que no existeix una comanda que derivi dues vegades respecte al temps.

## Capítol 4

# Cas d'Estudi 1: Motor de Contínua

El primer objecte d'estudi que es va dur a terme en aquest projecte va ser la realització del model i la corresponent simulació en Modelica d'un motor de contínua. En un principi, es volia implementar la planta de laboratori que es va utilitzar a les pràctiques de l'assignatura Control Automàtic i en aquest sentit es va procedir en els primers passos, però finalment es va descartar la idea perquè això implicava realitzar una sèrie d'experiments d'identificació de sistemes no lineals, tema que està fora dels objectius d'aquest projecte.

Encara això, es va procedir a realitzar el model amb una estratègia de menor a major dificultat, així es va realitzar amb dues aproximacions distintes: mitjançant blocs amb la interfície OMEdit del OpenModelica i mitjançant programa en llenguatge Modelica respectivament.

En conclusió el flux de treball va ser:

1. Es va realitzar la simulació en llaç obert de la funció de transferència obtinguda a les pràctiques de Control Automàtic (consultar Annex A).
2. Es va tancar el llaç i es va simular mitjançant el controlador PI obtingut a les pràctiques.
3. Mitjançant el model matemàtic d'un motor de contínua, es va realitzar el model en llenguatge Modelica. A partir d'aquí es romp amb la temptativa d'intentar simular la planta de laboratori amb la identificació dels seus paràmetres des del model no linial i es decideix utilitzar un model lineal d'un motor extret de [10].
4. Es realitza el disseny del controlador PI.
5. S'implementa el controlador anterior al model en llenguatge Modelica i es simula en llaç tancat.

Les dues primeres simulacions no s'inclouen a la memòria sinó que es reporten a l'Apèndix A ja que no tenen cap sentit d'estudi, sinó que es van realitzar com a primer contacte amb l'eina de simulació.

## 4.1 Model matemàtic

Per la realització d'aquest primer objecte d'estudi, es realitza el model del motor de contínua (veure Figura 4.1), velocitat del qual es controla mitjançant el voltatge aplicat a l'induït del motor ( $V_a$ ). Amb aquest tipus de control s'obté la màxima capacitat de parell. Per la realització del model és necessari utilitzar l'equació relacionada amb l'induït així com l'equació que regeix la dinàmica del moviment.

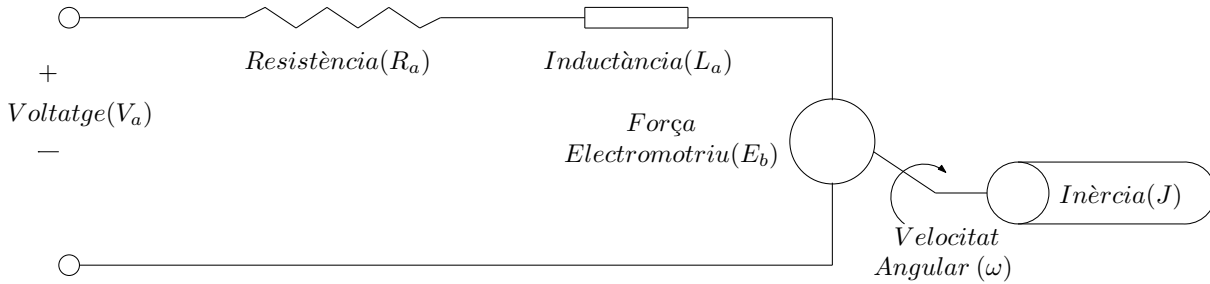


FIGURA 4.1: Esquemàtic del motor de contínua

Generalment, el parell generat pel motor ( $\Gamma_e$ ) és directament proporcional a la intensitat de corrent que passa per l'induït ( $i_a$ ) i al camp magnètic produït. En aquest cas, es considera constant el camp magnètic i que el parell del motor és únicament proporcional a la intensitat de l'induït. També es considera que la força electromotriu ( $E_b$ ) és proporcional a la velocitat angular ( $\omega$ ) multiplicada per una constant electromotriu  $K_e$  diferent per a cada motor.

Així, amb les simplificacions realitzades, les equacions que representen el comportament del sistema són:

$$V_a = i_a(t)R_a + L_a \frac{d}{dt}i_a(t) + E_b \quad (4.1)$$

$$\Gamma_e(t) = J \frac{d\omega(t)}{dt} \quad (4.2)$$

$$E_b(t) = K_e \omega(t) \quad (4.3)$$

$$\Gamma_e(t) = K_T i_a(t). \quad (4.4)$$

En el model no s'ha inclòs cap parell produït per la càrrega (es podria modelitzar com a pertorbació del sistema) ni tampoc cap tipus de fregament del rotor. Els paràmetres per realitzar la simulació es van obtenir de l'article [10] i són els següents:



TAULA 4.1: Paràmetres del model del motor de contínua

Paràmetres	Valor
Voltatge a l'induït, $V_a$	Variable [V]
Intensitat a l'induït, $i_a$	Variable [A]
Resistència a l'induït, $R_a$	10 [ $\Omega$ ]
Inductància a l'induït, $L_a$	5 [mH]
Contant del parell, $K_T$	0.09 [Nm/A]
Constant electromotriu, $K_E$	0.09 [V s/rad]
Inèrcia, J	0.01 Kg/m <sup>2</sup>

## 4.2 Implementació amb Modelica

El primer model que es presenta és el model del motor de contínua en llaç obert sense controlador. El model en llenguatge Modelica que es genera i simula és el següent:

---

```

model DCMotor_TimeDomain
  //Variables i paràmetres del motor
  constant Modelica.SIunits.Resistance Ra=10;
  constant Modelica.SIunits.Inductance La=5e-3;
  constant Modelica.SIunits.Inertia J=0.01;
  constant Modelica.SIunits.ElectromotiveForce Ke=0.09;
  constant Modelica.SIunits.ElectricalTorqueConstant Kt=0.09;
  parameter Modelica.SIunits.Voltage Vref=2;
  Real I;
  Real theta;
  Real omega;

  equation
    //Equacions del motor
    Vref=Ra*I + La*der(I) + Ke*omega;
    omega=der(theta);
    J*der(omega)=Kt*I ;

end DCMotor_TimeDomain;

```

---

Sobre aquest model ja es pot fer les simulacions que es vulguin per veure com es comporta, per exemple, la velocitat angular o la intensitat a l'induït en llaç obert. Aquestes simulacions es mostren a les Figures 4.2 i 4.3 respectivament.

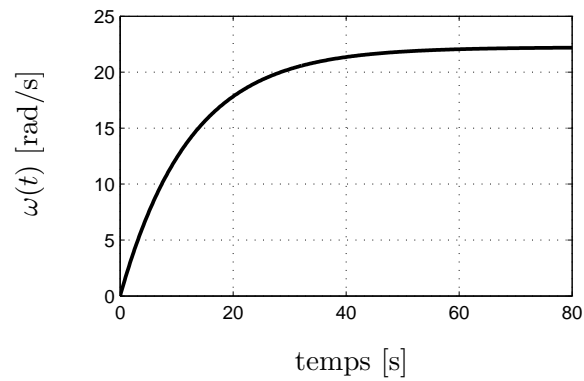


FIGURA 4.2: Resposta transitoria de la velocitat angular del motor de contínua

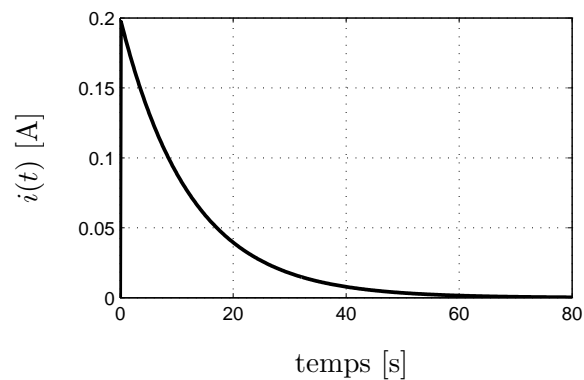
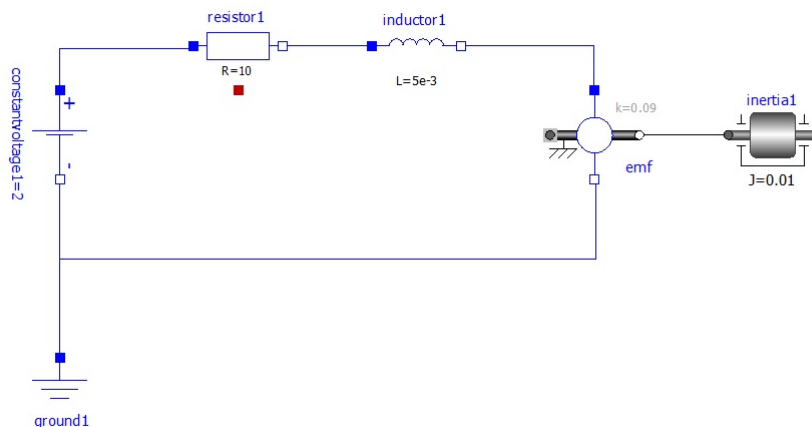


FIGURA 4.3: Resposta transitoria de la intensitat del motor de contínua

A part del model en Modelica en forma de codi, es pot obtenir el model a partir de la biblioteca anomenada *Modelica Standard Library* desenvolupada per l'associació de Modelica, la qual inclou components dels dominis elèctrics, tèrmics, fluids i sistemes de control. La Figura 4.4 mostra l'esquema del elements del motor utilitzant aquestes llibreries.

FIGURA 4.4: Modelització física del motor de contínua utilitzant la *Modelica Standard Library*

El motor de contínua també es pot modelitzar d'una forma causal mitjançant OpenModelica per comprovar com aquesta tècnica de modelatge perd tot sentit físic (veure Figura 4.5).

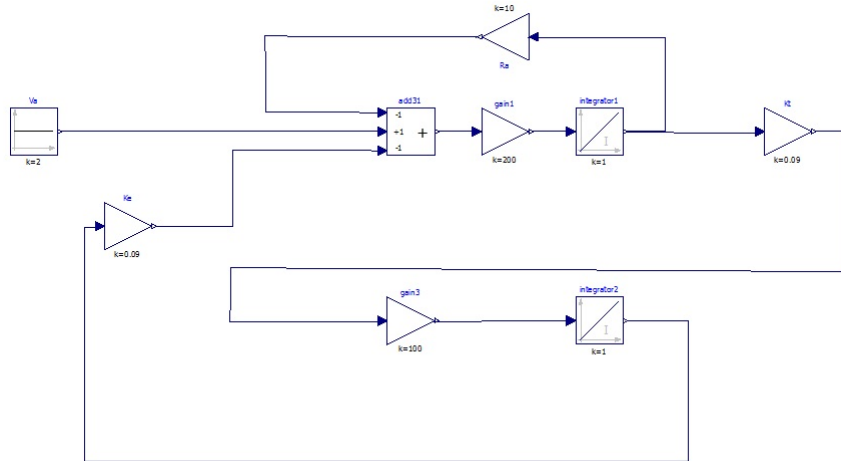


FIGURA 4.5: Modelització Causal Motor de Contínua

Cal notar que, mitjançant el segon paradigma de modelització, la modelització per blocs, es perd completament el sentit físic del sistema i no existeix cap relació a primera vista model-sistema.

### 4.3 Disseny i implementació en simulació del controlador PI

El control del motor de contínua es realitza mitjançant un controlador proporcional-integral (PI). El disseny del controlador es fa per prova i error, sabent que es vol aconseguir una velocitat sense sobrepuig i amb error zero a l'estacionari d'acord a l'esquema de la Figura 4.6.

El diagrama de blocs que queda és un clàssic llaç tancat amb realimentació unitària negativa, on a la cadena directa es troba el controlador i el motor d'acord a l'esquema de la Figura 4.6, on  $r$  és la referència o consigna i  $y_m$  és la sortida del sistema.

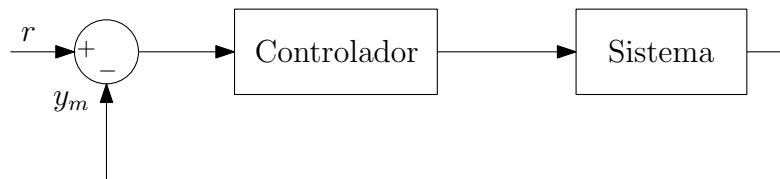


FIGURA 4.6: Llaç de control

La implementació del model del llaç de control en Modelica es realitza de la següent forma:

---

```

model DCMotor_TimeDomain
  //Variables i paràmetres del motor
  constant Modelica.SIunits.Resistance Ra=10;
  constant Modelica.SIunits.Inductance La=5e-3;
  constant Modelica.SIunits.Inertia J=0.01;
  constant Modelica.SIunits.ElectromotiveForce Ke=0.09;
  constant Modelica.SIunits.ElectricalTorqueConstant Kt=0.09;
  parameter Modelica.SIunits.Voltage Vref=2;

  Real I;
  Real theta;
  Real omega;

  //Variables i paràmetres del controlador
  parameter Real K=0.7511;          // Sintonitzar el controlador que es vulgui
  parameter Real T(unit="s")=0.3235; // Sintonitzar el controlador que es vulgui
  Real error;
  Real outCtr;
  Real x;
equation
  //Equacions del motor
  outCtr=Ra*I + La*der(I) + Ke*omega;
  omega=der(theta);
  J*der(omega)=Kt*I ;
  error=Vref-omega;
  // Equacions del controlador
  der(x)=error/T;
  outCtr=K*(error+x);
end DCMotor_TimeDomain;

```

---

El controlador PI segueix l'equació següent:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int e(\tau) d\tau, \quad (4.5)$$

on  $K_p$  és l'acció proporcional,  $T_i$  és el temps integral que ajusta l'acció integral,  $u(t)$  és l'acció de control i  $e(t)$  és l'error que es produeix entre el valor mesurat i la referència, i.e.,  $e(t) = r - y_m(t)$ .

En el llenguatge Modelica existeix la funció `der()` que deriva directament respecte al temps, però no existeix una que realitzi una integració indefinida. Així, s'implementa el controlador de la següent manera:

```
der(x) = error/T;
outCtr = K*(error+x);
```

on  $x$  és l'estat del controlador,  $T$  és el temps integral i  $K$  l'acció proporcional.

## 4.4 Simulacions i resultats

L'ajust dels paràmetres del PI es realitza amb el mètode prova i error. En aquest mètode l'acció principal és l'acció proporcional i l'integral serveix per refinar la sortida i per evitar l'error de sortida en règim permanent.

Per ajustar els paràmetres, l'acció integral es deixa al mínim (per exemple imposant un temps d'integració  $T_i$  molt elevat) i es modifica l'acció proporcional fins obtenir una sortida desitjada.

Es desitja una sortida sense sobrepuig (*overshoot*) i amb un error zero a l'estat estacionari (s'aconsegueix mitjançant l'acció integral). Les Figures 4.7 i 4.8 mostren l'ajust realitzat.

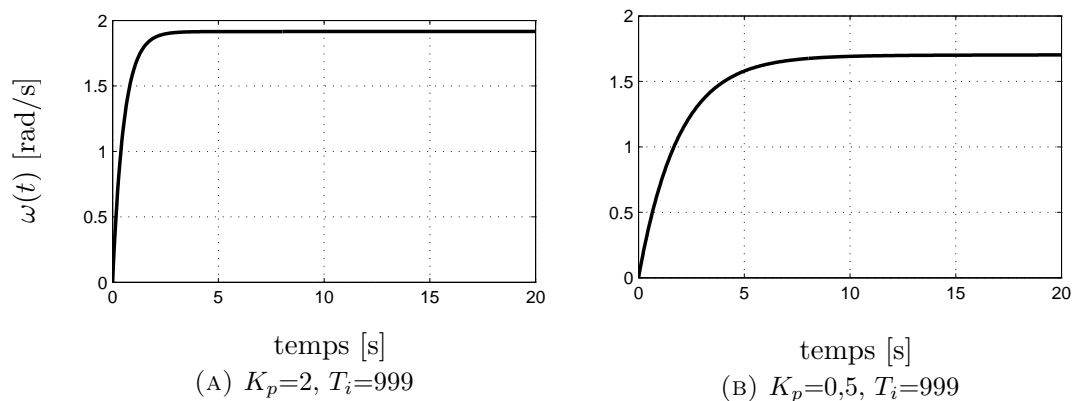


FIGURA 4.7: Resposta transitòria per  $T_i = 999$  i per a diferents valors de  $K_p$

S'eligeix la sortida amb la  $K_p$  de 0.5 per la seva forma més suau i es comença a ajustar l'acció integral.

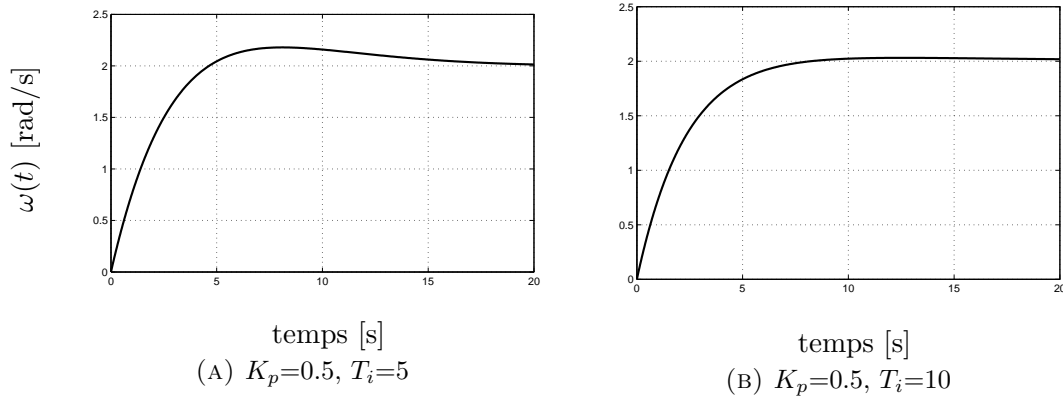


FIGURA 4.8: Resposta transitòria per  $K_p = 0.5$  i per a diferents valors de  $T_i$

Així doncs l'acció proporcional que s'acaba implementant és de 0,5 i el temps integral de 10s. Per sustentar els resultats obtinguts amb Modelica, es realitzen els càlculs teòrics sobre el sistema en laïç tancat per trobar el sobrepuig i valor en estat estacionari.

L'espai d'estats del sistema en laïç obert és el següent:

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} &= \begin{bmatrix} 0 & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V, \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix}. \end{aligned} \quad (4.6)$$

Com els valors de  $K_e$  i  $K_T$  són iguals, es considera  $K = K_e = K_T$ .

Per trobar la funció de transferència del sistema s'utilitza la transformada de Laplace amb condicions iguals a zero i s'obté

$$V_a(s) = I_a(s)R_a + L_a s I(s) + K\omega(s), \quad (4.7)$$

$$K I_a(s) = J s \omega(s). \quad (4.8)$$

Realitzant les operacions necessàries, la funció de transferència del sistema és la següent:

$$G_p(s) = \frac{\omega(s)}{V_a(s)} = \frac{K}{L_a J s^2 + R_a J s + K^2}. \quad (4.9)$$

Aplicant la transformada de Laplace a l'equació del controlador (4.5) s'obté la funció de transferència del controlador, i.e.,

$$G_c(s) = \frac{U(s)}{E(s)} = \frac{K_p s + K_I}{s}, \quad (4.10)$$

on

$$K_I = \frac{K_p}{T_i}. \quad (4.11)$$

Amb les dues funcions de transferència anteriors i sabent que s'implementa un llaç de control amb realimentació (Figura 4.6) es pot obtenir la funció de transferència global

$$G_{Total}(s) = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)}. \quad (4.12)$$

Es realitza el següent estudi sobre els dos darrers controladors (amb la  $K_p$  fixa a 0,5):

#### 4.4.1 $K_p = 0.5, T_i = 5$

$$\begin{aligned} G_p(s) &= \frac{0.09}{5 \times 10^{-5}s^2 + 0.1s + 0.0081} \\ G_c(s) &= \frac{0.5s + 0.1}{s} \\ G_{Total}(s) &= \frac{0.045(s + 0.2)}{5 \times 10^{-5}s^3 + 0.1s^2 + 0.0531s + 0.009} \\ &= \frac{900(s + 0.2)}{s^3 + 2000s^2 + 1062s + 180}. \end{aligned}$$

Es poden calcular els pols de la funció de transferència total:

$$p_1 = -1.9995 \times 10^3, \quad p_{2,3} = -0.2655 \pm 0.1397j.$$

El primer que es pot comprovar és si existeix dominància de pols per la qual cosa el sistema es podria aproximar com un sistema de segon ordre ja que no té sentit parlar de factor d'esmoreïment per a sistemes amb model d'ordre 3 o superior.

El criteri que s'utilitza per establir la relació de dominància entre pols és el següent:

$$\frac{\Re(p_{dominat})}{\Re(p_{dominant})} > 5. \quad (4.13)$$

Si la relació entre el pol dominat,  $p_{dominat}$ , i el pol dominant,  $p_{dominant}$ , és major que 5 es determina que existeix una relació de dominància [8]. Els pols que estan més a prop de l'eix imaginari són els que tenen un transitori més lent i per això dominen la sortida del sistema.

En el cas que es presenta ja és obvi que es sobrepassa aquesta relació:

$$\frac{\Re(p_{dominat})}{\Re(p_{dominant})} = \frac{-1999.5}{-0.2655} = 7531.07 \gg 5. \quad (4.14)$$

Per realitzar la reducció del sistema s'ha de mantenir el guany estàtic per a que el comportament en règim permanent sigui idèntic al real, i.e.,

$$\begin{aligned}\lim_{s \rightarrow 0} G_{Total}(s) &= \lim_{s \rightarrow 0} G_{equivalent}(s) \\ \lim_{s \rightarrow 0} G_{Total}(s) &= \lim_{s \rightarrow 0} \frac{900(s + 0.2)k^*}{s^2 + 0.5311s + 0.090} \\ 1 &= \frac{180k^*}{0.090} \\ k^* &= 5 \times 10^{-4}.\end{aligned}$$

Així la funció de transferència equivalent és la següent:

$$G_{equivalent} = \frac{0.45(s + 0.2)}{s^2 + 0.5311s + 0.090}.$$

Es pot graficar la resposta del sistema reduït per una entrada graó d'amplitud 2, gràfica que es mostra en la Figura 4.9.

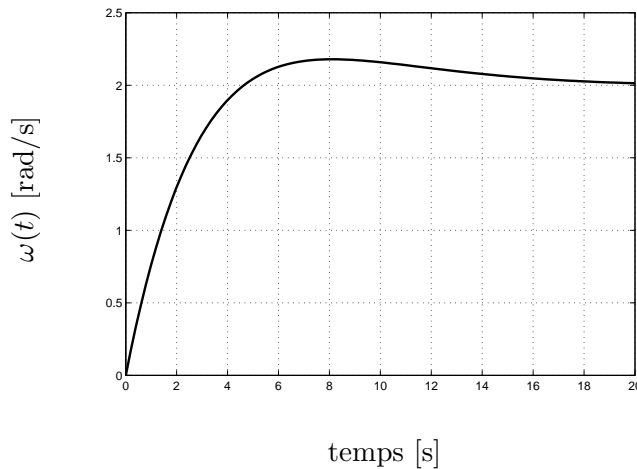


FIGURA 4.9: Resposta transitoria del model per al motor de contínua reduït

Es veu que la resposta del sistema de segon ordre és completament igual al sistema de tercer ordre (Figura 4.8 (A)). Es pot calcular el sobrepuig gràficament i surt un sobrepuig del 8.953%. El sobrepuig es produïx per que el zero de la funció de transferència està aprop de l'eix imaginari: quan més allunyat menys oscil·lació.

El valor en règim permanent es pot trobar aplicant el teorema del valor final, sabent que la entrada és un graó d'amplitud igual a 2 V, i.e.,

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s) \quad (4.15)$$

$$\lim_{s \rightarrow 0} sG_{equivalent}(s) \frac{2}{s} = 2. \quad (4.16)$$



El valor en estat estacionari és 2, és a dir, igual que la entrada el que significa que es té un error igual a zero. Això passa per l'acció integral del controlador que elimina l'error en estat estacionari.

#### 4.4.2 $K_p = 0.5, T_i = 10$

$$Gp(s) = \frac{0.09}{5 \times 10^{-5}s^2 + 0.1s + 0.0081}$$

$$Gc(s) = \frac{0.5s + 0.05}{s}$$

$$G_{Total}(s) = \frac{0.045(s + 0.1)}{5 \times 10^{-5}s^3 + 0.1s^2 + 0.0531s + 0.0045}$$

$$= \frac{900(s + 0.1)}{s^3 + 2000s^2 + 1062s + 90}$$

Els pols de la funció de transferència total són els següents:

$$p_1 = -1.9995 \times 10^3, \quad p_2 = -0.1058, \quad p_3 = -0.4253.$$

Al no existir cap parell de pols complexos conjugats, el sistema no hauria d'oscil·lar per la qual cosa tendria un sobrepuig igual a zero. Pel mateix procediment que al punt que a la Secció 4.4.1, es pot calcular el sistema reduït de segon ordre (torna a existir dominància):

$$G_{equivalent}(s) = \frac{0.45(s + 0.1)}{s^2 + 0.5311s + 0.045}$$

L'acció del zero produeix aquest petit sobrepuig i és menor que amb el controlador anterior degut a que el zero està entre els dos pols a l'eix real. El sobrepuig és de 1.54%.

A més pel mateix motiu que al punt anterior, el valor final del sistema serà igual al de l'entrada ja que es tracta d'un graó i el error en estat estacionari es zero gràcies a la acció integral del controlador.

## 4.5 Conclusions

En aquest problema s'ha estudiat un motor de contínua utilitzant un model lineal simplificat (obviant fregament i parell de càrrega) mitjançant Modelica. S'ha fet servir un controlador PI per a la regulació de la velocitat: l'acció integral al augmentar l'ordre del sistema elimina l'error en estat estacionari per una entrada graó.

S'ha vist que al disminuir el temps integral  $T_i$  per un guany constant en el controlador  $K_p$  es disminueix el temps d'establiment però augmenta el sobrepuig per la qual cosa es decideix agafar un temps integral major. Els resultats que s'obtenen es resumeixen a la Taula 4.2.

TAULA 4.2: Comparació de Resultats

	<b>Sis. Reduït A</b>	<b>Sis. Complet A</b>	<b>Sis. Reduït B</b>	<b>Sis. Complet B</b>
Sobrepuig [%]	8.953	8.953	1.5420	1.5420
Valor Final [rad/s]	2	2	2	2

La comprovació que s'ha realitzat ha estat comparar els valors del sistema real de tercer grau (Modelica) amb els del mateix sistema reduït a un segon grau.

## Capítol 5

# Cas d'Estudi 2: Reactor Contínuament Agitat (CSTR)

El *Continuous Stirred-Tank Reactor* (CSTR) és el reactor més bàsic utilitzat en l'enginyeria química. Aquests són sistemes oberts en els quals el material és lliure d'entrar i sortir del sistema i opera en condicions de temps de règim continu, és a dir, mentre té lloc la reacció química a l'interior del reactor. aquest és alimentat constantment de material i també s'extreu de manera ininterrompuda els productes de la reacció.

Degut a la forta no linealitat del model del CSTR, la tasca de modelatge d'aquest problema és sempre molt atractiva de cara a l'estudi en l'enginyeria de control. Si aquest model no és suficientment precís, l'anàlisi i disseny del controlador posteriors seran erronis. A més, aquesta dinàmica no lineal del sistema comporta la necessitat d'utilitzar controls diferents als clàssics ja que aquests últims es centren sobretot en sistemes lineals com pot ser el controlador PID.

### 5.1 Descripció del sistema

El model del CSTR proposat per Pottman and Seborg [15] s'ha utilitzat per a les simulacions i per al disseny de controladors en aquest capítol. En el procés es considera una reacció irreversible, exotèrmica  $A \rightarrow B$  (producte A de concentració  $C_{Af}$  és convertit en producte B de concentració  $C_A$ ) que es dona lloc en un reactor a volum constant que està envoltat per una jaqueta metàl·lica per on passa un cabal de refrigerant,  $q_c$  a temperatura  $T_c$  (veure Figura 5.1).

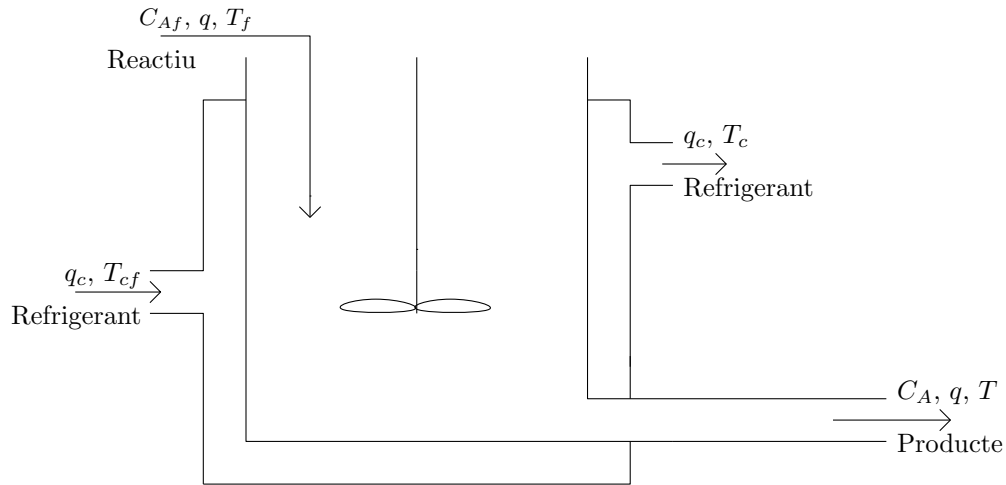


FIGURA 5.1: Esquema del CSTR

El model matemàtic basat en principis físics consta de dues equacions, un balanç de massa i un d'energia. El balanç de massa del component A és

$$\frac{dc_A}{dt} = \frac{q}{V}(c_{Af} - c_A) - k_0 c_A e^{-\frac{E}{RT}}. \quad (5.1)$$

El balanç d'energia dintre el reactor, considerant volum constant, calor específic a pressió constant  $C_p$ , densitat i sense considerar canvis en l'energia cinètica com en l'energia potencial, és

$$\frac{dT}{dt} = \frac{q}{V}(T_f - T) + \frac{(-\Delta H)k_0 c_A}{\rho C_p} e^{-\frac{E}{RT}} + \frac{\rho_c C_{pc}}{\rho C_p V} q_c \left[ 1 - e^{-\frac{hA}{q_c \rho_c C_{pc}}} \right] (T_c - T). \quad (5.2)$$

Els valors dels paràmetres relacionats amb el CSTR es troben a la Taula 5.1 i han estat extrets de l'article [13].

TAULA 5.1: Paràmetres del CSTR i els seus corresponents valors

Paràmetres	Valor
Volum del CSTR. $V$	10 [l]
Cabal d'alimentació. $q$	Variable [l/min]
Concentració a la alimentació. $c_{Af}$	1 [mol/l]
Temperatura a la alimentació. $T_f$	350 [K]
Cabal de refrigerant. $q_c$	100 [l/min]
Temperatura del refrigerant. $T_c$	350 [K]
Densitats. $\rho$ . $\rho_c$	1000 [g/l]
Calors específics. $C_p$ . $C_{pc}$	1 [cal/(gK)]
Factor pre-exponencial. $k_0$	$7.2 \times 10^{10}$ [l/min]
Factor exponencial. $E/R$	$9.98 \times 10^3$ [K]
Calor de la reacció. $-\Delta H$	$2.0 \times 10^5$ [cal/mol]
Carac. de transferència de calor. $hA$	$7.0 \times 10^5$ [1/(minK)]

## 5.2 Model matemàtic en Modelica

L'objectiu de realitzar aquest model, a part de realitzar un estudi de control sobre el mateix, consisteix en utilitzar la capacitat de Modelica en la reutilització de classes. Així, es realitza una classe *tanc* on es definirà el comportament propi del sistema per després poder reutilitzar aquesta classe per ajuntar diferents CSTR i fer un control més atractiu al poder controlar diferents tancs a la vegada.

Així doncs la implementació del model matemàtic de la Secció 5.1 segueix el següent codi:

---

```

model tank
  parameter Real V=100 "Tank volume";
  parameter Real q=100 "Feed flow rate";
  parameter Real c_af=1 "Feed concentration";
  parameter Real Tf=350 "Feed temperature";
  parameter Real qc=10 "Coolant flow rate";
  parameter Real Tc=350 "Coolant temperature";
  parameter Real rho=1000 "Density";
  parameter Real rhoc=1000 "Coolant density";
  parameter Real Cp=1 "Specific Heat";
  parameter Real Cp_c=1 "Coolant Specific Heat";
  parameter Real ko=7.2e10 "Pre-exponential factor";
  parameter Real EdivR=9.98e3 "Exponential factor";
  parameter Real dh=2e5 "Heat of reaction";
  parameter Real hA=7e5 "Heat transfer charact.";
  parameter Real T_init=350;
  parameter Real c_init=0;
  Real c(start=c_init,fixed=true);
  Real T(start=T_init,fixed=true);

equation
  der(c)= q*(c_af-c)/V - ko*c*exp(-EdivR/T);
  der(T)= q*(Tf-T)/V - ((dh*ko*c)/(rho*Cp))*exp(-EdivR/T)+
    ((rhoc*Cp_c)/(rho*Cp*V))*qc*(1-exp(-(hA)/(qc*rhoc*Cp_c)))*(Tc-T);

end tank;

```

---

Mitjançant aquest model ja es pot realitzar un anàlisi del comportament del sistema, en aquest cas, les variables a estudi són tant la concentració dintre del tanc com la temperatura a la que es troba el producte dins el tanc.

A l'Annex B es dona una possible solució de la modelització d'un CSTR mitjançant la modelització en blocs utilitzant SimuLink.

### 5.3 Problema de control

El problema que s'ha proposat és controlar un procés industrial que integra quatre CSTR, dels quals es vol mantenir les seves concentracions el més a prop possible a unes referències, en aquest cas  $c_{A,1}^{ref} = 0.1$  mol/l,  $c_{A,2}^{ref} = 0.15$  mol/l,  $c_{A,3}^{ref} = 0.2$  mol/l i  $c_{A,4}^{ref} = 0.25$  mol/l. La Figura 5.2 mostre un esquema del problema de control proposat.

La variable de control és el cabal d'alimentació de cada tanc, i.e.,  $u_i = q_i$ . Aquest cabal es suposa limitat dins un rang, i.e.,  $u_i \in [u_{i,min}, u_{i,max}]$ .

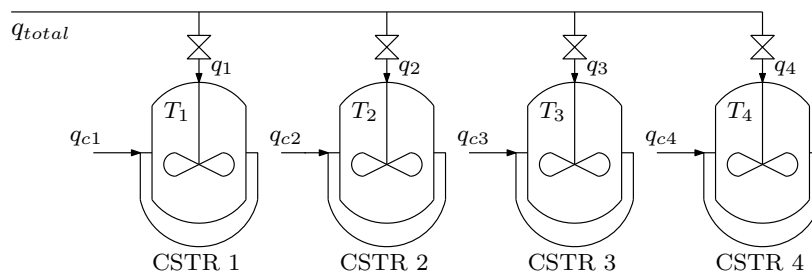


FIGURA 5.2: Esquema del problema de control proposat

El problema doncs que es vol resoldre és el següent:

$$\min_{\mathbf{u}_i(t)} J = \int_{t_0}^{t_f} (c_i^{ref} - c_i(t))^2 dt, \quad (5.3a)$$

subjecte a

$$u_i \in \mathcal{U}_i, \quad (5.3b)$$

$$MOC_i, \quad (5.3c)$$

amb  $i = 1 \dots 4$ ,  $\mathbf{u}_i(t) = [u_i(t_0) \quad u_i(t_0 + \tau) \quad \dots \quad u_i(t_f)]^\top \in \mathbb{R}^\phi$  (el paràmetre  $\phi$  serà discutit en la Secció 5.4) és una seqüència en l'interval  $[t_0 \quad t_f]$  amb intervals temporals de  $\tau \in \mathbb{R}$ . Notis que  $\tau = \frac{t_f - t_0}{\phi - 1}$ . A més la solució del problema (5.3) es denota per  $\mathbf{u}_i^*(t) \in \mathbb{R}^\phi$ . En aquest problema  $\mathcal{U}_i = \{u_i \in \mathbb{R} : u_{i,min} \leq u_i \leq u_{i,max}\}$ ,  $MOC_i$  és el model orientat a control de l' $i$ -èssim CSTR i  $c_i^{ref}$  és la concentració de referència de l' $i$ -èssim CSTR.

L'elecció d'aquests intervals de  $u_i$  no és una decisió trivial ja que al ser un sistema no lineal i no tenir un únic òptim global, pot haver-hi diferents solucions correctes. En aquest cas, es limita el cabal d'entrada de cada tanc entre  $[0,1]$  l/min. Encara no tenir un sentit molt aplicable a la indústria, en aquest treball interessa més estudiar el seu sentit matemàtic.

La implementació que es vol realitzar és un control enllaç tancat en el qual es tenen dos models, el model de control utilitzat per resoldre el problema d'optimització (5.3) i el model de simulació.

## 5.4 Disseny del laç de control

La implementació del laç de control compta dels següents passos que es van repetint per cada temps de mostreig del controlador. El laç que s'implementa es pot veure en la Figura 5.3.

1. Es resol la optimització pel primer interval del controlador ( $\Delta t = t_f - t_0 = 0.1$  min) i s'extreu els valors òptims de les variables de control  $\mathbf{u}_i^* \in \mathbb{R}^\phi$ .
2. Les trajectòries òptimes s'estableixen com les entrades per al model de simulació.
3. Es realitza la simulació amb el model de simulació i s'extreuen els estats  $\mathbf{x}_i = [c_{A,i} \quad T_i]^\top$ .
4. S'actualitzen els estats del model de control amb els extrets al pas anterior.
5. Es mou l'interval de simulació i es torna a repetir tot el procés.

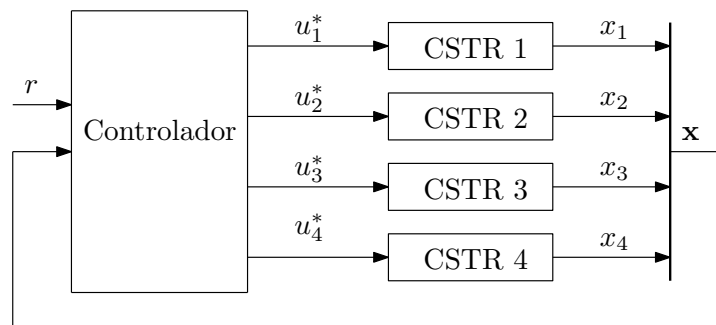


FIGURA 5.3: Llaç de Control de la Optimització

El problema d'optimització (5.3) es resol mitjançant l'estratègia que ve per defecte a l'entorn JModelica.org: mitjançant un algorisme de col·locació directa es transforma el problema en un programa no lineal, el qual posteriorment és resolt per un algorisme d'optimització de punts interiors (IPOPT) [2].

En termes generals, el procediment que realitza la plataforma JModelica.org és el següent: mitjançant el compilador processa el codi en Modelica i Optimica. Es crea una representació simbòlica del problema d'optimització gràcies a l'eina CasADi [7], representació que serà aproximada a un programa no lineal mitjançant el mètode de col·locació directe. Finalment el programa no lineal es resol mitjançant IPOPT.

Amb el mètode de col·locació directa s'aproxima un problema dimensionalment infinit en un programa no lineal dimensionalment finit. La idea fonamental és discretitzar les equacions diferencials utilitzant diferències finites, transformant així el problema en un programa no lineal dimensionalment finit. Les claus d'aquesta aproximació són tres variables:

$H_p$	L'horitzó del problema d'optimització
$n_e$	El nombre d'elements finits
$n_{cp}$	El nombre de punts de col·locació

El temps d'horitzó es divideix en  $n_e$  elements en els quals cada variable del problema que depengui del temps és aproximada mitjançant polinomis de grau  $n_{cp}$ , anomenats *polinomis de col·locació* [12].

La localització dels punts de col·locació es realitza per la col·locació de Radau (opció per defecte del programa), que sempre posa un punt de col·locació al final de cada interval per assegurar-ne la continuïtat als límits dels elements mentre que els altres punts serveixen per maximitzar la precisió de la solució. El nombre de punts de col·locació,  $n_{cp}$ , afecta al paràmetre  $\phi$  que és la longitud del vector  $\mathbf{u}_i$ :  $\phi = n_{cp} + 1$ . Si es desitja trobar una explicació més profunda de l'operació de càlcul que es realitza durant la optimització, consultar [12].

## 5.5 Tipus de models

La plataforma JModelica.org compta amb diversos models per a compilar, els quals s'expliquen seguidament.

### 5.5.1 Model de simulació

Per a la simulació de models dinàmics, JModelica.org compta amb objectes anomenats FMU (Functional Mock-up Unit) que segueixen la FMI (Functional Mock-up Interface), veure [1]. Aquests objectes són creats amb el paquet `Assimulo` de JModelica.org [3]. La simulació de models es realitza mitjançant els algorismes per defecte de JModelica, els quals utilitzen el *solver* `CVode` [3]. Els models de simulació basats en Modelica (sense referències a la classe `Optimica` per a optimitzacions) es guarden en archius `.mo`.

La compilació del model dels quatre CSTR es realitza mitjançant un `script` de Python:

---

```

from pymodelica import compile_fmu
from pyfmi import load_fmu

sim_fmu = compile_fmu("sim_model", "casadi.mop")
sim_model = load_fmu(sim_fmu)

```

---

L'objecte `sim_model` és un objecte FMU amb totes les seves funcions. Així, es pot simular mitjançant la funció pròpia dels FMU que és `simulate`:



---

```
res = sim_model.simulate(start_time = start_time, final_time = final_time,
                        input = input_object, option = opts)
```

---

Les característiques més importants de la funció `simulate` són:

- `start_time` i `final_time` indiquen el temps inicial i final de la simulació.
- `input` és una trajectòria de les variables de control ( $\mathbf{u}_i^*$ ).
- `options` és un diccionari d'opcions com el nombre de punts de col·locació o el *solver* a utilitzar.

### 5.5.2 Model d'optimització

Els models d'optimització estan expressats en format Optimica i han d'estar guardats en arxius `.mop` a diferència dels de simulació que es carreguen des de arxius `.mo`. En aquest problema, s'anomena a l'arxiu `casadi.mop` i la classe d'Optimica on es defineix la optimització `tank_opt`.

L'algoritme d'optimització per defecte de la plataforma JModelica.org es basa en el paquet de CasADi que utilitza l'algoritme de col·locació directa per a la resolució de problemes d'optimització dinàmiques [7].

Com al model de simulació, la compilació s'executa a través de comandes de Python:

---

```
from pyjmi import transfer_optimization_problem
op = transfer_optimization_problem("tank_opt", "casadi.mop")
```

---

L'objecte `op` es transforma en un problema d'optimització i hereta totes les funcions dels objectes de la interfície de CasADi. Entre aquestes funcions està la de `optimize` que invoca l'algoritme de col·locació directa i el de l'IPOPT i resol el problema d'optimització:

---

```
res_op = op.optimize(options=op_opts)
```

---

Les opcions de la funció `optimize`, entre d'altres que s'utilitzen, són el nombre d'elements i el nombre de punts de col·locació (per més informació sobre les possibles opcions de la comanda `optimize`, consultar [3]).

## 5.6 Models en Modelica/Optimica

### 5.6.1 Punt inicial

Els tancs inicialment es troben en un punt d'equilibri. Aquest punt ve marcat per les concentracions dels tancs (inicialment 0 mol/l), les temperatures inicials dels tancs (350 K) i el cabal d'entrada (0 l/min). Per comprovar que es tracta d'un punt d'equilibri es calcula el pla de fase, el qual es mostra a la Figura 5.4.

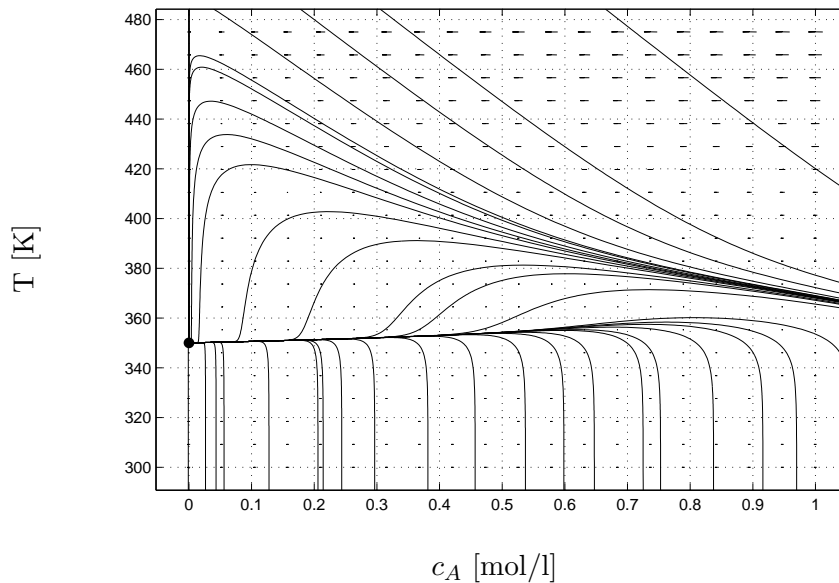


FIGURA 5.4: Pla de fase del CSTR i punt d'equilibri inicial

Es pot estudiar l'estabilitat d'aquest punt d'equilibri mitjançant el mètode indirecte de Lyapunov [11] com segueix. Es té  $x$  un punt d'equilibri pel sistema no lineal

$$\dot{x} = f(x), \quad (5.4)$$

on  $f : D \rightarrow \mathbb{R}^n$  és una funció contínua diferenciable i  $D$  és subespai de punts al voltant del punt  $x$ . A més, es defineix

$$A = \left. \frac{\partial f}{\partial t}(x) \right|_{x=x_p}, \quad (5.5)$$

amb  $x_p$  definit com un punt d'operació. A més  $\lambda$  és el  $i$ -èssim valor propi de la matriu  $A$ , on  $i = 1, \dots, n$ . Llavors,

1. El punt  $x_p$  és asimptòticament estable si  $\Re(\lambda_i) < 0$  per tots els valors propis de  $A$ .
2. El punt  $x_p$  és inestable si  $\Re(\lambda_i) > 0$  per un o més valors propis de  $A$ .

Així, si es calcula el Jacobià del model matemàtic no lineal del CSTR i s'evalua en el punt d'operació desitjat  $[c_{Ai} \ T_i]^\top = [0 \ 350]$  i  $q_i = 0$ :

$$A = \left. \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \right|_{x=x_p} = \begin{bmatrix} -0.028 & 0 \\ 5.623 & -0.999 \end{bmatrix}, \quad (5.6)$$

on els valors propis són  $\lambda_1 = -0.999$ ,  $\lambda_2 = -0.028$  per la qual cosa es pot establir que es tracta d'un punt asimptòtica-ment estable.

### 5.6.2 Model de simulació

Utilitzant la capacitat de Modelica per a la reutilització de classes resulta relativament senzill la implementació del problema amb quatre CSTR:

---

```

model sim_model
  input Real u1(start=q1_init, fixed=true) = tank1.q;
  input Real u2(start=q2_init, fixed=true) = tank2.q;
  input Real u3(start=q3_init, fixed=true) = tank3.q;
  input Real u4(start=q4_init, fixed=true) = tank4.q;

  tank tank1(c(start=c1_init, fixed=true, min=0),
    T(start=T1_init, fixed=true, min=0));
  tank tank2(c(start=c2_init, fixed=true, min=0),
    T(start=T2_init, fixed=true, min=0));
  tank tank3(c(start=c3_init, fixed=true, min=0),
    T(start=T3_init, fixed=true, min=0));
  tank tank4(c(start=c4_init, fixed=true, min=0),
    T(start=T4_init, fixed=true, min=0));

  parameter Real q1_init;
  parameter Real q2_init;
  parameter Real q3_init;
  parameter Real q4_init;

  parameter Real c1_init;
  parameter Real c2_init;
  parameter Real c3_init;
  parameter Real c4_init;

  parameter Real T1_init;

```

```

parameter Real T2_init;
parameter Real T3_init;
parameter Real T4_init;

end sim_model;

```

---

Les classes `tank1`, `tank2`, `tank3`, `tank4` hereten la classe `tank` (consultar la Secció 5.2) amb la diferència que es canvia la declaració del paràmetre `parameter Real q=100` per `input Real q`; declarant així el cabal d'alimentació com la entrada de cada CSTR.

Els valors inicials tant del cabal, concentració i temperatura es declaren al `script` de Python.

### 5.6.3 Model de control

En el model de control com ja s'ha comentat, es defineix el problema d'optimització (5.3). En aquest cas, el model de simulació i el de control són completament iguals ja que s'implementa un control no lineal. Les úniques diferències resideixen en que es canvia la capçalera de `model` per la de `optimization` que és la que utilitza el compilador `Optimica` per traduir els problemes d'optimització escrits en llenguatge `Modelica` i s'afegeix una secció anomenada `constraint` on es defineixen les restriccions a les que està sotmés el model. A més, es defineix una funció de cost la qual es minimitzarà per trobar els cabals d'entrada òptims que facin que les concentracions arribin als seus valors desitjats  $c_{A,i}^{ref}$ .

Així, la nova capçalera queda:

```

optimization tank_opt(objective=(J(finalTime)), startime=0,
                      finalTime=0.1)

```

---

on `startime` i `finalTime` són els temps inicial i final de la optimització (temps de mostreig del controlador).

En la funció de cost  $J$  de l'equació (5.3a) únicament es penalitza la diferència de concentració dels tancs respecta a la referència

```

der(J) = (c1_ref-tank1.c)^2 + (c2_ref-tank2.c)^2 + (c3_ref-tank3.c)^2
        + (c4_ref-tank4.c)^2;

```

---

on `ci_ref` són les concentracions a les que es volen mantenir els tancs en estat estacionari. Les restriccions que s'afegeixen són les que restringeixen els cabals dins el rang  $[0,1]$  l/min.

Si es vol comprovar el codi de l'optimització en `Optimica` consultar l'Apèndix B.

## 5.7 Escenaris de control

### 5.7.1 Escenari 1: Cabal d'alimentació com a variable de control

Es realitza la simulació per un temps igual a 200 minuts i es grafiquen les concentracions dels tancs (veure Figura 5.5) així com les accions de control necessàries (veure Figura 5.6).

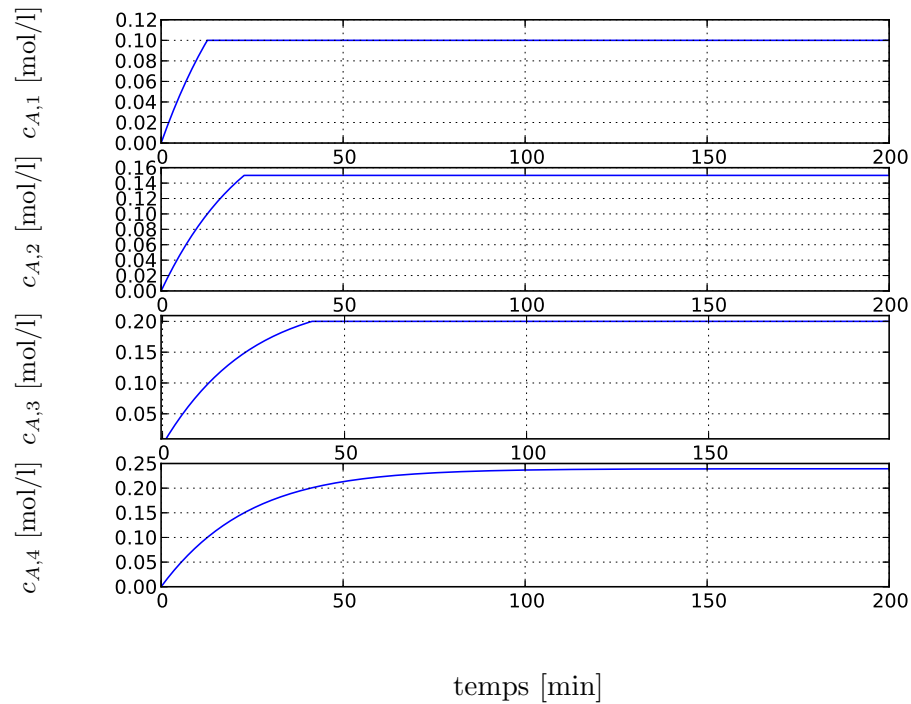
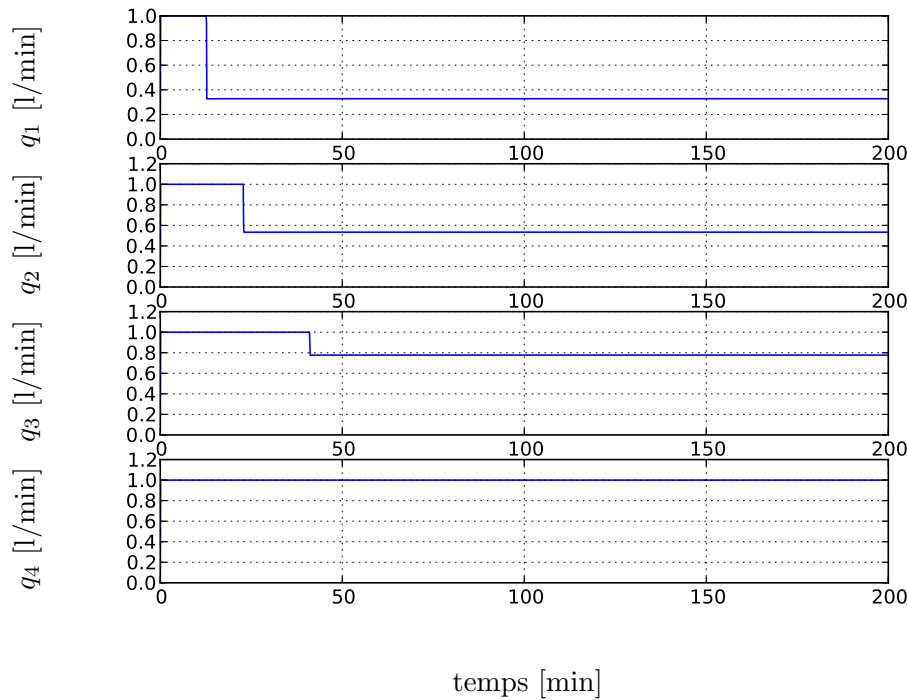


FIGURA 5.5: Concentracions per a l'Escenari 1

FIGURA 5.6: Variable de Control,  $q_i$ , per a l'Escenari 1

Les referències dels tancs són  $c_{A,1} = 0.1$  mol/l,  $c_{A,2} = 0.15$  mol/l,  $c_{A,3} = 0.2$  mol/l i  $c_{A,4} = 0.25$  mol/l (consultar Secció 5.3). La variable de control, al no tenir una penalització en la funció objectiu sobre el seu canvi, satura immediatament a 1 l/min ja que intenta arribar a la referència en el mínim temps possible. Passat un cert temps, el controlador troba un nivell de cabal que manté les concentracions a un valor que minimitza la diferència entre la referència i el valor de la concentració (funció objectiu  $J$  a l'equació (5.3a)).

Com es veu, les referències dels tres primers tancs arriben a la referència, mentre que el CSTR 4 arriba a un valor molt proper a 0.25, en concret de 0.238 mol/l. Això és degut a que amb la restricció de  $u_i \in [0,1]$  l/min és impossible arribar a aquesta concentració. Per arribar a la concentració de 0.25 mol/l, s'hauria de relaxar la restricció. Els resultats que s'obtenen es resumeixen a la Taula 5.2.

TAULA 5.2: Resultats de l'Escenari 1 després de 200 minuts de simulació

CSTR <sub><i>i</i></sub>	$c_{A,i}$ [mol/l]	$T_i$ [K]	$q_i$ [l/min]	Error <sub>relatiu</sub> $c_{A,i}$ %
CSTR <sub>1</sub>	0.10	350.59	0.327	0
CSTR <sub>2</sub>	0.15	350.90	0.534	0
CSTR <sub>3</sub>	0.20	351.23	0.777	0
CSTR <sub>4</sub>	0.239	351.50	1	-4.4

L'error relatiu ( $Error_{\text{relatiu}}$ ) es defineix com el quocient entre l'error absolut (diferència entre el valor de la mesura i el valor real) i el valor real de la mesura, i.e.,

$$Error_{\text{relatiu}} = \frac{y_{\text{mesura}} - y_{\text{real}}}{y_{\text{real}}}. \quad (5.7)$$

Per comprovar els resultats es pot dibuixar el pla de fase. per exemple del tanc 4 per les condicions finals i observar el seu comportament. Això es mostra a la Figura 5.7.

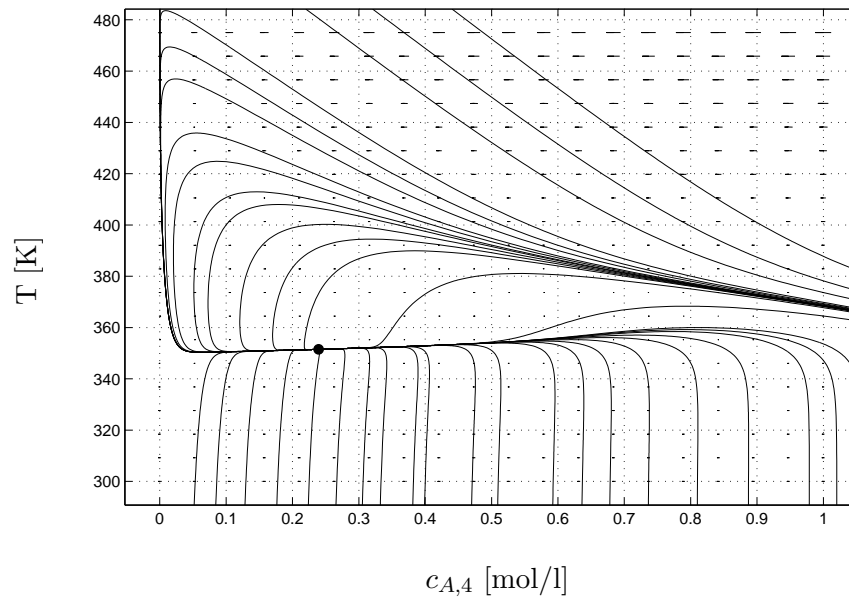


FIGURA 5.7: Pla de fase CSTR<sub>4</sub>

S'observa que es té un punt d'equilibri en el punt  $[c_{A,4} \quad T_4] = [0.239 \quad 351.50]$  per la condició de  $u_4 = 1$  l/min.

Com s'ha comentat la concentració del tanc 4 no arriba a la referència. per la qual cosa fa pensar d'una altra possible forma d'abordar el problema.

### 5.7.2 Escenari 2: Cabal d'alimentació i refrigeració com a variables de control

El que es pretén en aquest estudi és aconseguir les referències a tots els tancs. Amb aquest objectiu es realitza el mateix plantejament que l'Escenari 1 [5.7.1] però pensant en incorporar el cabal de refrigerant com a variable de control, per a que ajudi a augmentar la concentració del CSTR 4. Cal notar, que d'aquesta manera, el problema de control passa a tenir dues variables d'entrada.

Encara que únicament es necessita augmentar la concentració del CSTR 4, s'afegeixen tots els cabals de refrigeració com a variables de control.

Així el problema d'optimització es transforma de la següent manera:

$$\min_{\mathbf{u}_i(t), \mathbf{q}_{ci}(t)} J_{E2} = \int_{t_0}^{t_f} (c_i^{ref} - c_i(t))^2 dt, \quad (5.8a)$$

subjecte a

$$u_i \in \mathcal{U}_i, \quad (5.8b)$$

$$q_{ci} \in \mathcal{Q}_{ci}, \quad (5.8c)$$

$$MOC_i, \quad (5.8d)$$

amb  $i = 1 \dots 4$ , on  $\mathbf{q}_{ci}(t) = [q_{ci}(t_0) \quad q_{ci}(t_0 + \tau) \quad \dots \quad q_{ci}(t_f)]^T \in \mathbb{R}^\phi$ ,  $\mathcal{Q}_{ci} = \{q_{total} \in \mathbb{R} : q_{c,min} \leq q_c \leq q_{c,max}\}$ .

Es restringeix el  $q_{ci}$  en un interval de  $[100, 200]$  l/min. Aquest interval és 100 vegades superior al del cabal d'alimentació  $q_i$  degut a que el refrigerant afecta molt poc en la dinàmica de la concentració  $c_i$ .

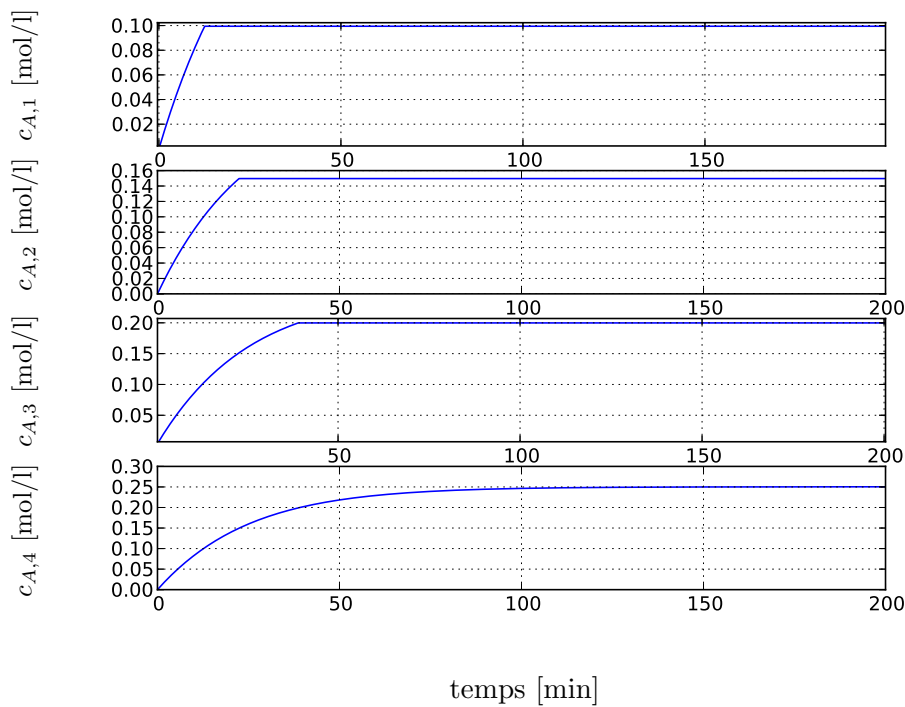


FIGURA 5.8: Concentracions Escenari 2



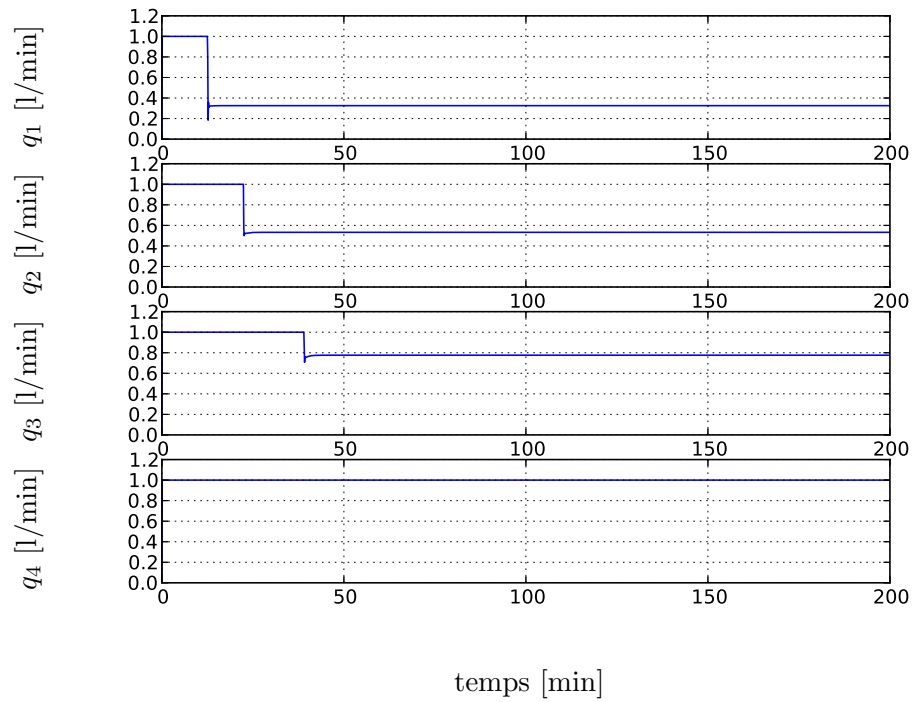


FIGURA 5.9: Cabal d'alimentació Escenari 2

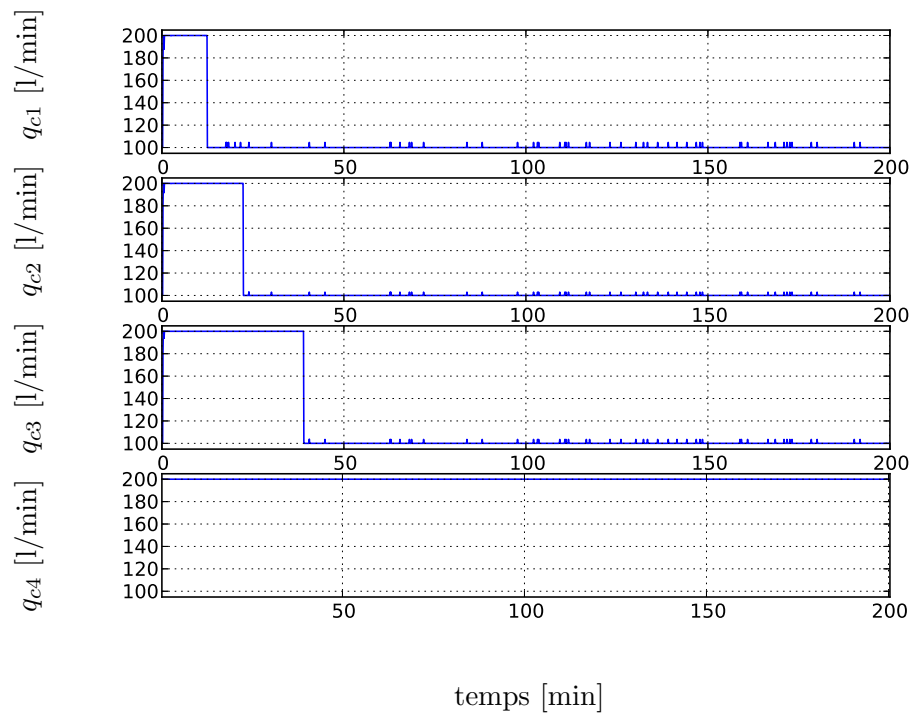


FIGURA 5.10: Cabal de refrigeració Escenari 2

Amb l'entrada del cabal de refrigeració com part del vector de variables de control, la concentració del tanc 4 arriba a la referència (necessita un cabal de refrigeració igual a 200 l/min). Els cabals de refrigeració dels tres primers tancs al principi saturen la restricció ja que intenten arribar a la referència en el menys temps possible per després disminuir fins els 100 l/min com al primer problema en el qual era constant i no variable.

TAULA 5.3: Resultats de l'Escenari 2 després de 200 minuts

CSTR <sub>i</sub>	$c_{A,i}$ [mol/l]	$T_i$ [K]	$q_i$ [l/min]	$q_{ci}$ [l/min]	Error <sub>relatiu</sub> $c_{A,i}$ %
CSTR <sub>1</sub>	0.10	350.59	0.327	100	0
CSTR <sub>2</sub>	0.15	350.90	0.534	100	0
CSTR <sub>3</sub>	0.20	351.23	0.777	100	0
CSTR <sub>4</sub>	0.25	351.77	1	200	0

Es pot tornar a dibuixar el pla de fase com a la Figura 5.11 per comprovar que el punt al que arriba el CSTR<sub>4</sub> es tracta d'un punt d'equilibri.

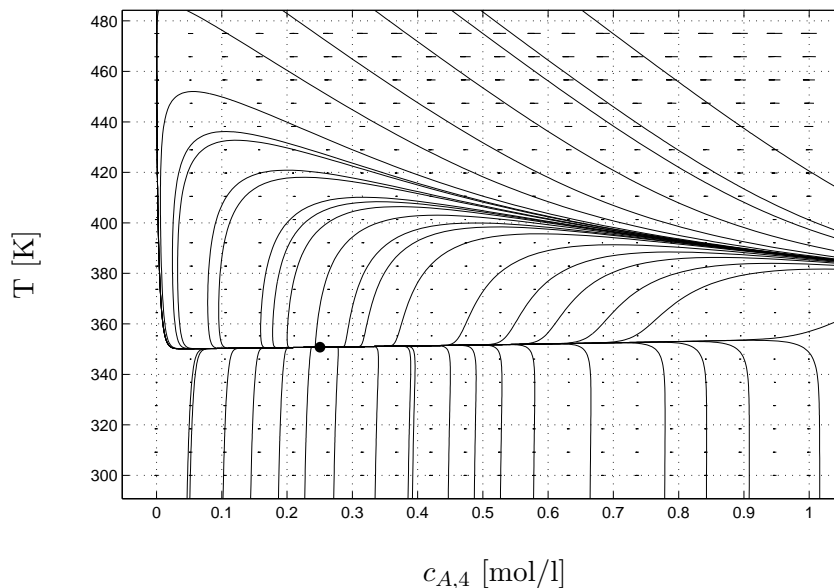


FIGURA 5.11: Pla de fase CSTR<sub>4</sub> amb l'Escenari 2

### 5.7.3 Distribució dinàmica de recursos

En l'àmbit tecnològic, un recurs és un mitjà que serveix per a la realització d'un procés o operació tècnica. Per exemple, en el cas dels CSTR presentat es pot pensar en el cabal d'alimentació o en el cabal de refrigeració (o ambdós). La característica d'aquest tipus de problema és que el recurs total sempre està limitat i sempre com a nombre no negatiu per tenir sentit físic. El repte de la resolució d'aquest problema és el cas de sistemes dinàmics en els quals la distribució de

recurs depèn de l'estat intern del sistema, on es necessita trobar la distribució òptima de recurs per aconseguir l'objectiu que es vulgui.

Per exemplificar aquesta problemàtica, es controlen les concentracions dels CSTR,  $c_{A,i}$ , únicament considerant el cabal d'alimentació com a variable de control (veure Escenari 1 a la Secció 5.7.1) però amb la condició que el cabal total està limitat.

Per aconseguir les referències s'ha vist que el cabal de cada tanca,  $q_i$ , no pot estar limitat en  $[0,1]$  l/min per la qual cosa es dobla l'interval,  $[0,2]$  l/min, però es limita el cabal total, i.e.,

$$q_{total} = \sum_{i=1}^4 q_i \in [0, 3] \text{ l/min.}$$

Així, la formulació matemàtica seria la següent:

$$\min_{\mathbf{u}_i(t)} J = \int_{t_0}^{t_f} (c_i^{ref} - c_i(t))^2 dt, \tag{5.9a}$$

subjecte a

$$u_i \in \mathcal{U}_i, \tag{5.9b}$$

$$q_{total} \in [0, 3], \tag{5.9c}$$

$$MOC_i. \tag{5.9d}$$

Així, si es fan les simulacions, la solució que es mostra a les Figures 5.12 i 5.13, respectivament.

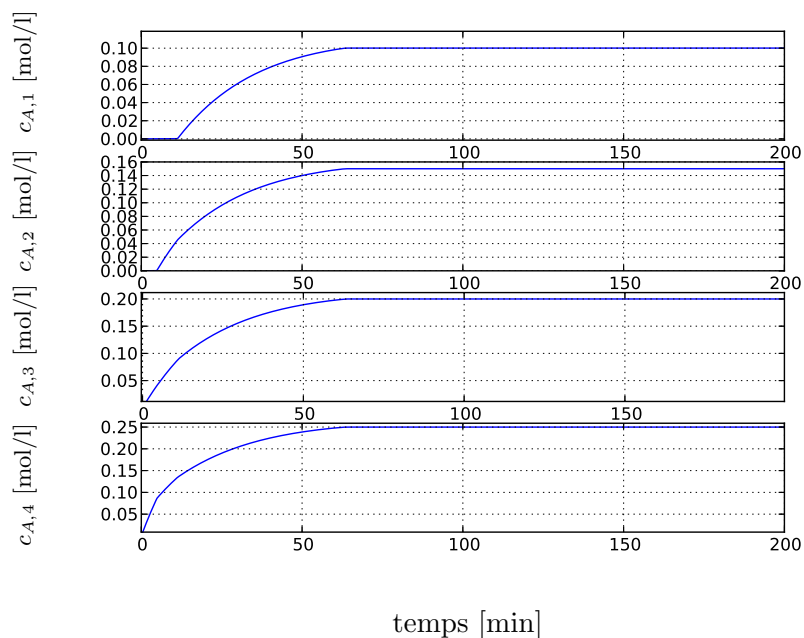


FIGURA 5.12: Concentracions Problema de Distribució Dinàmica

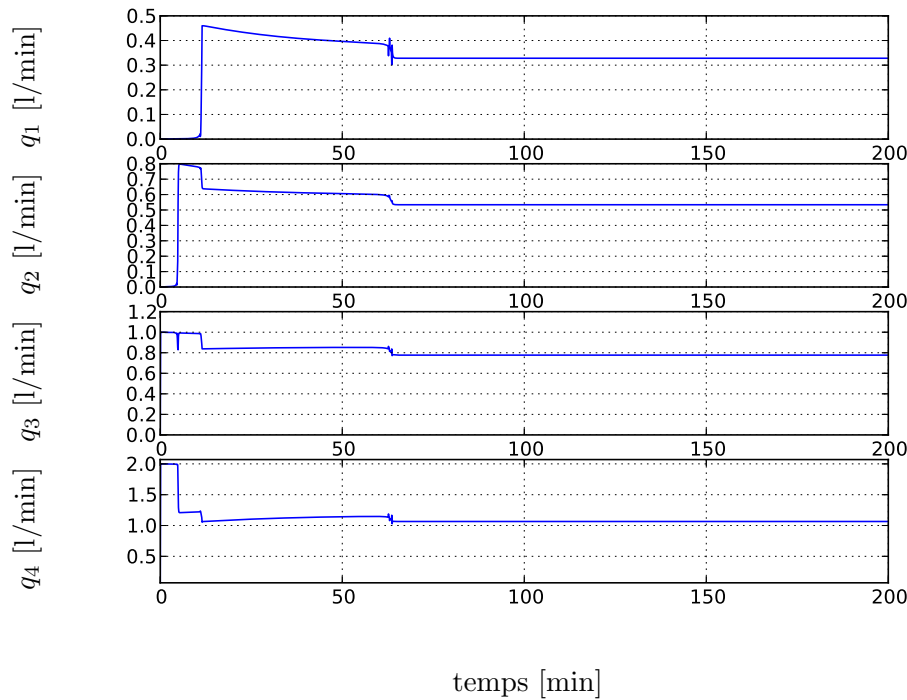


FIGURA 5.13: Cabals Problema de Distribució Dinàmica

S'observa que les concentracions arriben al valor de les referències però lo interessant d'aquest problema és el comportament de la variable de control  $q_i$ .

Clarament es pot observar que el controlador assigna prioritat als tancs als quals sels assigna unes concentracions de referència majors. en aquest cas els tancs 3 i 4. Per això, inicialment consumeixen ells dos solament tot el recurs disponible (1 l/min i 2 l/min) respectivament deixant als dos primers tancs sense recurs i per la qual cosa amb una concentració de zero (temps mort). Quan passa el temps, el control comença a assignar recurs als dos primers tancs baixant-ne els dels dos darrers, fins que troba un valor de cabal pel qual els tancs romanen a la concentració de referència.

## 5.8 Conclusions

En aquest problema s'ha estudiat el control de quatre CSTR. El problema s'ha desglossat en tres estudis. En els dos primers escenaris de control les entrades i sortides estan desacoblades per la qual cosa es dissenyen controladors locals per a cada CSTR. Cal notar que el problema 2 sorgeix de no poder aconseguir els objectius de control en quant al criteri que es requeria: la referència del quart CSTR no arriba a la concentració objectiu.

Els dos primers problemes, considerant el control per cada tanc desacoblat dels altres tancs, es tracten de dos problemes una entrada i una sortida (SISO per les seves sigles en anglès) i un

problema múltiples entrades i única sortida (MISO per les seves sigles en anglès), respectivament. Cal notar que, amb el disseny del controlador del segon problema, s'aconsegueixen els objectius del control.

El tercer escenari de control plantejat és la distribució de recursos. En aquest problema es consideren tots els tancs acoblats ja que l'entrada és compartida i afecta a tots els tancs, convertint tot l'escenari de control en un problema amb múltiples entrades i múltiples sortides (MIMO ja que el controlador ha d'estar pendent de les concentracions dels quatre tancs), i no únicament localment per a cada CSTR com passa al segon escenari de control.

En quant a Modelica, el llenguatge ha quedat validat en quant a la relativa facilitat que ofereix per descriure sistemes dinàmics, en aquest cas no lineals, (veure l'Annex B i comparar els models de Modelica amb la implementació mitjançant blocs) i la capacitat que ofereix de reutilitzar classes per enfrontar problemes més complexos.

JModelica.org, encara de les in comptables dificultats que s'han tingut alhora d'implementar el problema d'optimització dinàmica, ha quedat validada com una possible eina per resoldre problemes d'optimització basats en llenguatge Modelica.



## Capítol 6

# Anàlisi Econòmic

Aquest capítol contempla el cost total de realitzar el projecte desglossant el gast en equipament, llicències i recurs humà utilitzat durant la realització del mateix. El sou de l'estudiant de realitzar el treball de fi de grau (TFG) és de 8 €/hora que és el preu orientatiu que fixa la universitat politècnica de Catalunya (UPC) com a remuneració per a l'estudiant durant la realització d'un conveni de pràctiques a una empresa.

<hr/>			
1. SOFTWARE			
<hr/>			
Paquet bàsic de MatLab © amb llicència comercial			1368.80 €
	OpenModelica 1.9.1 beta 4		0 €
	JModelica.org		0 €
TOTAL SOFTWARE			1368.80 €
<hr/>			
2. HARDWARE			
	Toshiba Satellite S50-B-15Q Laptop		1185.00 €
TOTAL HARDWARE			1185.00 €
<hr/>			
3. SALARIS			
Tècnic de projecte			
	Remuneració d'estudiant de TFG	8 €/hora	
	Hores totals dedicades	360 hores	
	Total		2880.00 €
Director de projecte			
	Salari brut anual d'enginyer sènior	35000 €	
	Temps de dedicació	4 mesos	
	Coefficient de dedicació	10 %	
	Total		1166.67 €
TOTAL SALARIS			4046.67 €
<hr/>			
PRESSUPOST TOTAL			
<hr/>			
Software			1368.80 €
Hardware			1185.00 €
Salari			4046.67 €
TOTAL			6600.47 €
<hr/>			





## Capítol 7

# Impacte Ambiental

La realització d'aquest treball no ha suposat cap impacte ambiental de cap tipus. Ara bé, els estudis que comporta sobretot el dels quatre CSTR podrien tenir una dimensió ambiental que hauria de ser estudiada.

Els CSTR, com s'ha comentat al Capítol 5, són tancs molt utilitzats en la indústria química sobretot per part dels enginyers de control per implementar noves teories de control per a l'aplicació en la indústria. Els CSTR normalment s'utilitzen en processos de la indústria química orgànica i, en particular en la producció de plàstics i explosius, entre d'altres aplicacions. En quant a la fabricació de plàstics, el seu procés de fabricació és perjudicial per al medi ambient ja que la seva fabricació està basada en el petroli (un 4% de la producció mundial és utilitzat en la fabricació de plàstics) el que implica un gast d'energia de fonts no renovables. A més, la producció produeix gasos contaminants, com el  $\text{CO}_2$ , que són emesos a la atmosfera. És per això que és important l'enginyeria de control de processos per poder controlar de forma òptima les reaccions que es produeixen i poder optimitzar tot el procés d'obtenció de plàstics.

Un altre tema a tractar en quant a l'impacte ambiental és la simulació versus l'experimentació. La simulació com s'ha comentat en la introducció serveix per saber com es comporta un sistema real a partir de models matemàtics. Aquesta eina és molt profitosa de cara al possible impacte ambiental que pot produir l'experimentació. Per exemple, si es vol implementar el controlador PI dissenyat durant la memòria mitjançant el mètode prova i error, aquest mètode produeix una càrrega en el motor ja que per cada constant del controlador s'hauria de fer l'experiment i veure com es comporta la velocitat angular del mateix. Això, suposa un gast energètic que es veu relacionat amb un possible impacte ambiental. És per això, entre d'altres motius, que la simulació és una eina tan estesa en el món de la indústria.



## Capítol 8

# Conclusions i Treball Futur

### 8.1 Conclusions

En aquest treball s'ha realitzat un estudi sobre dos sistemes dinàmics, un de lineal i un de no lineal, mitjançant el llenguatge Modelica. Mitjançant aquests estudis s'ha pogut veure la gran capacitat del llenguatge de modelització de cara a la seva facilitat d'ús en la modelització de sistemes de considerable complexitat.

Modelica s'ha constituït com un estàndard en el modelatge i simulació de sistemes físics. Aquesta acceptació ve donada per la estreta relació en el seu desenvolupament entre la recerca i l'indústria. Aquest fet el fa adient no únicament per ser un llenguatge de modelització de cara a la docència si nó una potent eina de modelització en la indústria que ja està començant a tenir gran pes, sobretot en el sector automobilístic, aeroespacial i energètic, els quals tenen llibreries especialitzades per a la simulació dels seus components.

Com s'ha demostrat en aquest treball, Modelica té diferents entorns de simulació totalment gratuïts que, encara no estar a la altura de entorns comercials com el Dymola, són possibles alternatives de qualitat i el que siguin de codi lliure els dóna un valor afegit.

És difícil veure com Modelica amb un entorn de simulació suficientment competitiu pugui arribar a desbancar entorns de simulació tan establerts en la indústria, e.g. MatLab. Aquest fet no el resta valor sinó que és un valor afegit que aporta a un enginyer conèixer diferents formes de poder atacar un mateix problema i poder trobar diferents solucions depenent de les necessitats que es tinguin en el moment.

Deixant de banda Modelica, durant la realització d'aquest treball s'ha aprofundit en teoria de control ja que s'ha hagut de fer un estudi sobre un sistema no lineal amb un controlador òptim, no vist durant la realització del grau i que ha aportat un valor afegit al treball.

## 8.2 Treball futur

Els dos casos d'estudi desenvolupats al llarg de la memòria queden tancats i ara toca aprofundir més en el llenguatge Modelica ja que aquest té molts recursos que no han estat utilitzats.

Seria interessant poder enfrontar un problema d'enginyeria utilitzant un entorn comercial mitjançant llibreries d'elements per poder construir problemes més complexos.

En quant al que es refereix a teoria de control es considera el següent pas natural l'estudi del control predictiu basat en models molt utilitzat avui dia en la indústria i que s'espera adquirir durant la realització del màster.

Finalment aquest treball es pot considerar com una introducció a un treball futur, sigui un treball de fi de màster, que utilitzant els coneixements adquirits durant la realització d'aquest projecte i els nous que s'adquiriran durant la realització dels estudis doni un projecte interessant i si es pot que pugui aportar coneixement de cara a la recerca.

## Anex A

# Planta Laboratori

La planta de laboratori que es volia implementar en un primer moment és la planta de la empresa *L J Technical Systems* que compta amb un motor de corrent continu, que mitjançant un sistema de corretges acciona el moviment d'un eix de sortida.

La planta a més compta amb una dinamo tacomètrica utilitzada per mesurar la velocitat i un potenciòmetre per mesurar la posició de l'eix.

En la realització de la pràctica es demanava identificar la planta, la funció de transferència de la qual sortia de primer ordre:

$$G(s) = \frac{52.94}{0.28s + 1}, \quad (\text{A.1})$$

La sortida de la planta és una tensió proporcional a la velocitat de l'eix del motor produïda per la utilització de la dinamo tacomètrica ( $K_T = 0.017 \text{ V}/(\text{rad/s})$ ).

El diagrama de blocs que es va implementar mitjançant OpenModelica en el primer pas va ser el mostrat a la Figura A.1.

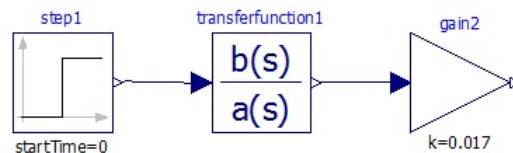


FIGURA A.1: Funció de transferència del motor

La interfície OMEdit d'OpenModelica per a la modelització gràfica també dóna traducció a codi en temps real:

```
model TFG_proba1
```

```
Modelica.Blocks.Sources.Step step1(height = 2) annotation(Placement(visible =
true, transformation(origin = {-40, 60}, extent = {{-10, -10}, {10, 10}},
rotation = 0)));
Modelica.Blocks.Continuous.TransferFunction transferfunction1(b = {52.94},
a = {0.28, 1}) annotation(Placement(visible = true, transformation(origin =
{0, 60}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));
Modelica.Blocks.Math.Gain gain2(k = 0.017) annotation(Placement(visible =
true, transformation(origin = {40, 60}, extent = {{-10, -10}, {10, 10}},
rotation = 0)));
```

```
equation
```

```
connect(step1.y, transferfunction1.u) annotation(Line(points = {{-29, 60},
{-27.3727, 60.2822}, {-10.4411, 59.8248}, {-14.2219, 59.8248}, {-12, 60}},
color = {0, 0, 127}));
connect(transferfunction1.y, gain2.u) annotation(Line(points = {{11, 60},
{37.6793, 60}, {28, 60.0465}, {28, 60}}, color = {0, 0, 127}));
annotation(Icon(coordinateSystem(extent = {{-100, -100}, {100, 100}},
preserveAspectRatio = true, initialScale = 0.1, grid = {2, 2})),
Diagram(coordinateSystem(extent = {{-100, -100}, {100, 100}},
preserveAspectRatio = true, initialScale = 0.1, grid = {2, 2})));
```

```
end TFG_proba1;
```

El segon pas que es va realitzar va ser mitjançant el controlador PID dissenyat al laboratori, implementar el llaç tancat com es mostra a la Figura A.2.

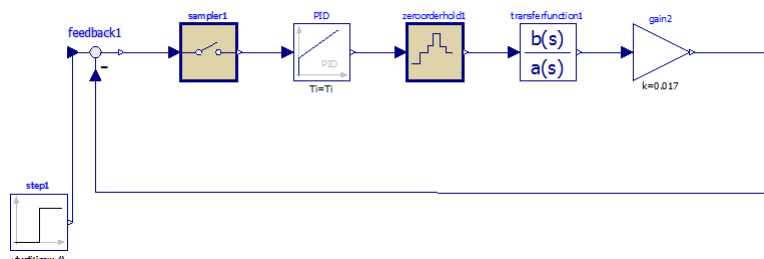


FIGURA A.2: Control Planta Laboratori

## Anex B

# CSTR

- CSTR Blocs

El model del CSTR també es va realitzar mitjançant el paradigma de modelització per diagrama de blocs amb el SimuLink de MatLab:

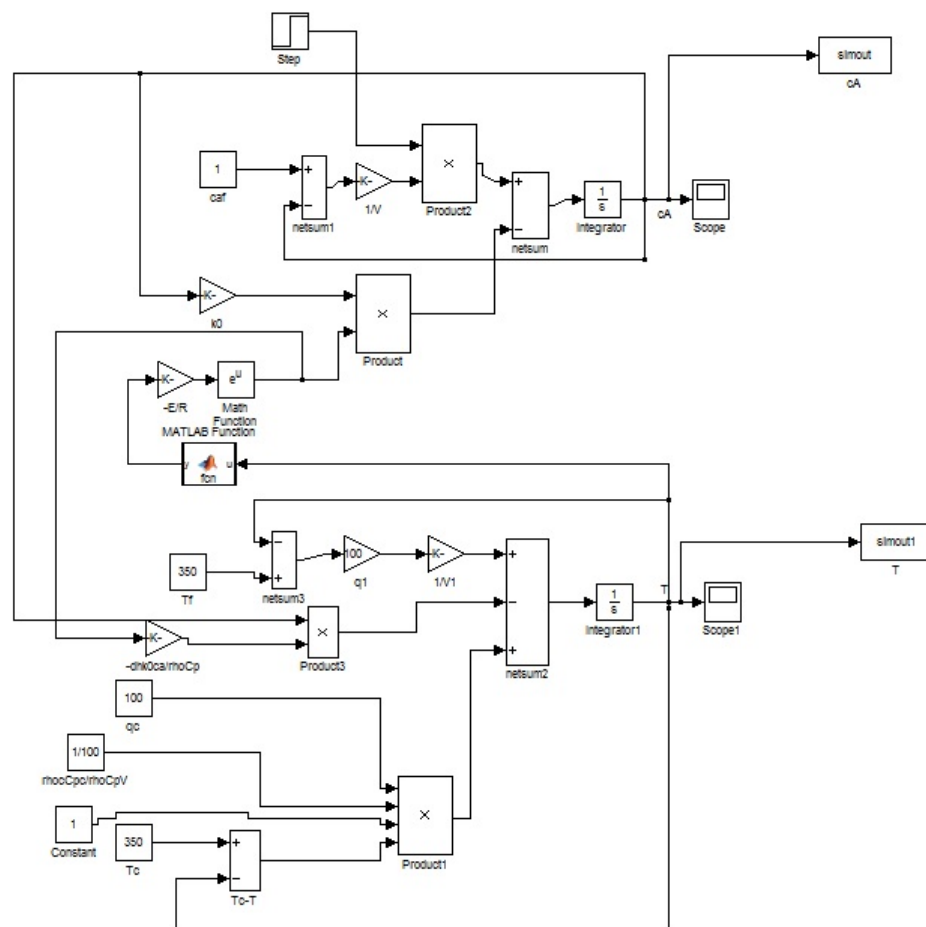


FIGURA B.1: CSTR mitjançant blocs MatLab/SimuLink

- Optimització amb codi Optimica (SISO)

```

optimization tank_opt(objective=(J(finalTime)),  startTime=0,
                      finalTime=0.1)

input Real u1(start=q1_init, fixed=true) = tank1.q;
input Real u2(start=q2_init, fixed=true) = tank2.q;
input Real u3(start=q3_init, fixed=true) = tank3.q;
input Real u4(start=q4_init, fixed=true) = tank4.q;

tank tank1(c(start=c1_init,fixed=true,min=0), T(start=T1_init, fixed=true, min=0));
tank tank2(c(start=c2_init,fixed=true,min=0), T(start=T2_init, fixed=true, min=0));
tank tank3(c(start=c3_init,fixed=true,min=0), T(start=T3_init, fixed=true, min=0));
tank tank4(c(start=c4_init,fixed=true,min=0), T(start=T4_init, fixed=true, min=0));

Real J (start=0, fixed=true, min=0);
Real q_total(start=0);

parameter Real q1_init;
parameter Real q2_init;
parameter Real q3_init;
parameter Real q4_init;

parameter Real c1_init;
parameter Real c2_init;
parameter Real c3_init;
parameter Real c4_init;

parameter Real T1_init;
parameter Real T2_init;
parameter Real T3_init;
parameter Real T4_init;

parameter Real c1_ref(fixed=true) = 0.1;
parameter Real c2_ref(fixed=true) = 0.15;
parameter Real c3_ref(fixed=true) = 0.20;
parameter Real c4_ref(fixed=true) = 0.25;

```



```
equation
der(J) = (c1_ref-tank1.c)^2 + (c2_ref-tank2.c)^2 + (c3_ref-tank3.c)^2
        + (c4_ref-tank4.c)^2 ;
q_total = u1 + u2 + u3 + u4;

constraint

u1>=0;
u1<=1.0;
u2>=0;
u2<=1.0;
u3>=0;
u3<=1.0;
u4>=0;
u4<=1.0;
q_total>=0;

end tank_opt;
```



# Bibliografia

- [1] Functional Mock-up Interface. Disponible Online: <https://www.fmi-standard.org/>. (darrer accés el 25 Juny 2015).
- [2] IPOPT. Disponible Online: <https://projects.coin-or.org/Ipopt>. (darrer accés el 20 Juny 2015).
- [3] JModelica.org User's Guide. Disponible Online: <http://www.jmodelica.org/page/236>. (darrer accés el 20 Juny 2015).
- [4] MODPROD Linköping University. Disponible Online: <https://www.modprod.liu.se/>. (darrer accés el 20 Juny 2015).
- [5] OpenModelica User's Guide. Disponible Online: <https://www.openmodelica.org/doc/OpenModelicaUsersGuide/OpenModelicaUsersGuide-latest.pdf>. (darrer accés el 20 Juny 2015).
- [6] J. Åkesson. Optimica—an extension of modelica supporting dynamic optimization. In *6th International Modelica Conference*, Bielefeld, Germany, 2008.
- [7] J. Andersson. *A General-Purpose Software Framework for Dynamic Optimization*. PhD thesis, KU Leuven – Faculty of Engineering Science, 2013.
- [8] R. Dorf and R H. Bishop. *Modern Control Systems, Twelfth Edition*. Prentice Hall, 2010.
- [9] P. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley-IEEE Press, 2004.
- [10] M. N. Kamarudin and S. M. Rozali. Simulink implementation of digital cascade control DC motor model - a didactic approach. In *PECon IEEE 2nd International Power and Energy Conference*, 2008.
- [11] H. K. Khalil. *Nonlinear Systems, Third Edition*. Prentice Hall, third edition, 2002.
- [12] F. Magnusson and J. Åkesson. Dynamic Optimization in JModelica.org. *Extended Version of the Paper Published in Proceedings of the 9th International Modelica Conference*, 2015.
- [13] P.H. Menold, F. Allgöwer, and R.K. Pearson. Nonlinear structure identification of chemical processes. *Computers and Chemical Engineering*, 21(0):137 – 142, 1997.

- [14] D. S. Naidu. *Optimal Control Systems*. CRC Press, 2003.
- [15] M. Pottman and D. E. Seborg. Identification of non-linear processes using reciprocal multiquadric functions. *Journal of Process Control*, 2(4):189 – 203, 1992.
- [16] K.J. Åström and T. Hägglund. The future of PID control. *Control Engineering Practice*, 9(11):1163 – 1175, 2001.
- [17] R. Vilanova and A. Visioli. *PID Control in the Third Millennium*. Springer London, 2012.