

# **Escola Universitària Politécnica de Mataró**

Centre adscrit a:



**UNIVERSITAT POLITÈCNICA  
DE CATALUNYA**

**Enginyeria Tècnica Industrial Especialitat en Electrònica Industrial**

## **Disseny de Controladors Avançats Aplicats a la Maqueta del Pèndol Invertit**

**Memòria**

**ALFREDO COLOMERA CASTELLANO  
PONENT: MIQUEL ROCA CISA**

**TARDOR 2011**

**Escola Universitària  
Politécnica de Mataró**



**TecnoCampus  
Mataró-Maresme**



## **Dedicatòria**

Dedicat a Manuel Varea professor gràcies al qual vaig adquirir tots el coneixements matemàtics suficients per arribar al punt final de la meva carrera universitària: la execució d'aquest projecte.



## **Agraïments**

Vull agrair a Miquel Roca Cisa, el meu professor de projecte, la dedicació a aquest treball. Sense els seus coneixements aquest projecte no hauria estat possible.



## **Resum**

El propòsit d'aquest projecte és conèixer i aplicar les possibilitats que dona el software Matlab en l'aplicació de mètodes de control d'espai-estats i lògica difusa. Aquests coneixements s'aplicaran sobre una maqueta formada per un pèndol acoblat a un carro. El pèndol té dos punts d'equilibri: el primer a baix estable i el segon a dalt inestable. Mitjançant els mètodes de control es portarà el pèndol fins al punt d'equilibri inestable i es mantindrà controlat en aquesta posició.

## **Resumen**

El propósito de este proyecto es conocer y aplicar las posibilidades que tiene el software Matlab en la aplicación de métodos de control de espacio-estados y control difuso. Estos conocimientos se aplicarán sobre una maqueta formada por un péndulo acoplado a un carro. El péndulo tiene dos puntos de equilibrio: un abajo estable y el segundo arriba inestable. Mediante los métodos de control se llevará el péndulo hasta el punto de equilibrio inestable y se mantendrá controlado en esta posición.

## **Abstract**

The purpose of this project is to learn and apply the possibilities by the software Matlab, in its methods of control state-space and control fuzzy. This knowledge is applied from a model consisting of a pendulum trailed to a cart. The pendulum has two equilibrium points: the first is below (stable) and the second is on top (unstable). Using the methods of control will lead the pendulum to the point of unstable equilibrium and will remain in control in this position.





## **Memòria descriptiva**



# Índex

Índex de taules.....	V
Glossari de termes.....	VII
1. Objectius.....	1
1.1. Propòsit.....	1
1.2. Finalitat.....	1
1.3. Objecte.....	1
1.4. Abast.....	1
1.5 Formulació del problema.....	1
1.5.1 Objectiu.....	2
1.6 Contingut del projecte.....	2
2. Descripció i modelatge del sistema.....	3
2.1. Descripció del pèndol invertit.....	3
2.1.1 Actuator.....	3
2.1.2 Driver.....	3
2.1.3 Adquisició de dades.....	4
2.1.4 Controlador.....	5
2.2. Model matemàtic del sistema.....	5
2.2.1. Equacions diferencials.....	5
2.2.1. Simulació del sistema modelitzat.....	9
2.3. Equació d'estat.....	11
2.4. Model discret.....	15
3. Estratègia de control.....	19
3.1. Disseny de sistema de control.....	19
3.1.1. Controlabilitat i observabilitat.....	19
3.1.2. Assignació del pols.....	20
3.2. Retorn d'estat.....	22
3.2.1. Disseny de la matriu K.....	23
3.2.2. Disseny de la matriu Kd.....	27
3.3. Observadors o Estimadors de variables d'estat.....	33
3.3.1. Disseny de la matriu L.....	34
3.3.2. Disseny de la matriu Ld.....	37
3.4. Control integral.....	40
3.4.1. Disseny de la nova matriu de realimentació Ka.....	42
3.4.2. Disseny de la nova matriu de realimentació Kda.....	44
4. Lògica Difusa.....	49
4.1. Introducció a la lògica difusa.....	49

## II

4.2. Lògica difusa.....	49
4.3. Fuzzyfication.....	50
4.4. Avaluació de regles.....	51
4.6. Defuzzyfication.....	52
4.7. Lògica difusa sobre el sistema.....	52
4.7.1. Fuzzyfication de les variables d'entrada.....	53
4.7.2. Inferència, defuzzification i defuzzification sobre el sistema.....	53
4.7.3. Simulació del sistema.....	57
4.7.3.1. Simulació sobre sistema no lineal.....	57
4.7.3.2. Simulació sobre sistema discret.....	58
5. Assajos dels controladors sobre el sistema real.....	61
5.1. Relació entre tensió i força.....	61
5.2. Linealització 360°.....	64
5.3. Aixecament de Feedback.....	65
5.4. Switch.....	66
5.5. Implementació del retorn d'estat .....	68
5.7. implementació del observador d'ordre complert més retorn d'estat.....	73
5.8. implementació control integral més retorn d'estat.....	75
6. Conclusions.....	83
6.1. Sobre Matlab.....	83
6.2. Sobre els algoritmes de control.....	83
6.3. Lògica difusa.....	83
6.4. Estudis futurs.....	83
7. Bibliografia.....	85

## Índex de figures

Fig. 2-1: unitat mecànica del pèndol.....	3
Fig. 2-2 unitat de control del pèndol.....	4
Fig. 2-3: adaptador per la tarja Advantech PCI-1711 i el driver DIGITAL PENDULUM CONTROLLER 33 – 201.....	4
Fig. 2-4: diagrama de blocs controlador.....	5
Fig. 2-5: diagrama de cos lliure del pèndul invertit.....	7
Fig. 2-6: diagrames de forces del pèndol i del carro.....	8
Fig. 2-7: model No lineal construït al Simulink.....	10
Fig. 2-8: resposta de la simulació del sistema per la posició del carro.....	10
Fig. 2-9: resposta de la simulació del sistema per la posició del pèndol.....	11
Fig. 2-10: model No lineal construït al Simulink.....	14
Fig. 3-1: Resposta esglaó dels sistemes de prototip de Bessel per[3].....	21
Fig. 3-2: diagrama de bolcs del control per retorn d'estats .....	22
Fig. 3-3: diagrama de blocs del control per retorn d'estat, sistema continu linealitzat.....	24
Fig. 3-4: configuració bloc State-Space.....	24
Fig. 3-5: configuració bloc Gain.....	25
Fig. 3-6: resposta del angle.....	25
Fig. 3-7: resposta de la derivada del angle.....	26
Fig. 3-8: resposta de la posició del carro.....	26
Fig. 3-9: resposta de la derivada de la posició del carro.....	27
Fig. 3-10: proposta 1 diagrama de blocs sistema discretitzat.....	29
Fig. 3-11: proposta 2 diagrama de blocs sistema discretitzat.....	29
Fig. 3-12: proposta 3 diagrama de blocs sistema discretitzat.....	30
Fig. 3-13: blocs d'entrada i sortida.....	30
Fig. 3-14: resposta del angle.....	31
Fig. 3-15: resposta de la derivada del angle.....	31
Fig. 3-16: resposta de la posició del carro.....	32
Fig. 3-17: resposta de la derivada de la posició del carro.....	32
Fig. 3-18: diagrama de bolcs del control per retorn d'estats mes observador.....	34
Fig. 3-19: diagrama de bolcs del control per retorn d'estats mes observador de Matlab.....	35
Fig. 3-20: configuració del bloc State-Space1.....	36
Fig. 3-21: resposta del angle.....	36

Fig. 3-22: resposta de la posició.....	37
Fig. 3-23: diagrama de bolcs del control per retorn d'estats mes observador discret.....	38
Fig.3-24: configuració del bloc Satae-Sapce1.....	39
Fig. 3-25: resposta del angle.....	39
Fig. 3-26: resposta de la posició.....	40
Fig. 3-27: diagrama de bolcs del sistema augmentat.....	42
Fig. 3-28: diagrama de bolcs del sistema augmentat de Matlab.....	43
Fig.3-29: muntatge dels senyals i la senyal d'entrada.....	43
Fig. 3-30: diagrama de bolcs del sistema augmentat no lineal discretitzat.....	45
Fig. 3-31: muntatge dels senyals i la senyal d'entrada.....	45
Fig. 3-33: muntatge dels senyals i la senyal d'entrada.....	46
Fig. 4-1: funcions de pertinença.....	51
Fig. 4-2: univers del discurs de la posició.....	53
Fig. 4-3: univers del discurs de l'angle.....	53
Fig. 4-4: divisió del espai recorregut pel pèndol en quadrants.....	54
Fig. 4-5: definició d'etiquetes per la variable theta amb l'editor de funcions de pertinença de Matlab .....	55
Fig. 4-6: definició d'etiquetes per la variable amb l'editor de funcions de pertinença de Matlab.....	55
Fig. 4-7: definició d'etiquetes per la variable amb l'editor de funcions de pertinença de Matlab.....	56
Fig. 4-8: Regles editades amb l'editor de regles de Matlab.....	56
Fig. 4-9: diagrama de blocs muntatge simulació lògica difusa.....	57
Fig. 4-10: resposta del la variable theta (angle).....	57
Fig. 4-11: resposta del la variable x (posició).....	58
Fig. 4-12: Diagrama de blocs simulació sistema discretitzat amb el bloc de Feedback per al pèndol .....	59
Fig. 4-13: resposta del la variable theta (angle).....	59
Fig. 4-14: resposta del la variable (posició).....	60
Fig. 5-1: diagrama de blocs de Matlab en enllaç obert.....	62
.....	62
Fig. 5-2: a l'esquerra configuració bloc Ram; a la dreta configuració bloc Saturation.....	62
Fig. 5-3: taula que relaciona tensió d'entrada i força.....	63
Fig. 5-4: bloc Look Table.....	63
Fig. 5-5: configuració bloc Look Table.....	64

### III

Fig. 5-6: bloc Embedded MATLAB Function.....	64
Fig. 5-7: programació bloc Embedded MATLAB Function.....	65
Fig. 5-8: blocs del exercici Swing Hold Pendulum Extra de Feedback.....	66
Fig. 5-9: blocs Swicth.....	66
Fig. 5-10: programació bloc Embedded MATLAB Function.....	67
Fig. 5-11: configuració del bloc switch.....	68
Fig. 5-12: diagrama de blocs del control per retorn d'estat.....	69
Fig. 5-13: diagrama de blocs intern del bloc Control.....	70
Fig. 5-14: gràfiques resultat de l'aplicació del disseny de Kd sobre el sistema real.....	70
Fig. 5-15: gràfiques resultat de l'aplicació del disseny de sobre el sistema real amb pertorbacions.....	71
Fig. 5-16: gràfiques resultat de l'aplicació del disseny de sobre el sistema real amb control difús al swing.....	72
Fig. 5-17: blocs interns del bloc Swing amb controlador difús.....	73
Fig. 5-18: diagrama de blocs intern del bloc Control per a un observador d'ordre complet.....	73
Fig. 5-19: gràfiques resultat de l'aplicació del disseny del observador sobre el sistema real.....	74
Fig. 5-20: diagrama de blocs del control per retorn d'estat més integrador.....	75
Fig. 5-21: diagrama de blocs intern del bloc integrador.....	76
Fig. 5-22 comptador de segons.....	77
Fig. 5-23 comptador de segons.....	77
Fig 5-24: Detall del muntatge per commutar al control integral.....	78
Fig. 5-25: gràfic d'un senyal quadrat d'entrada, integrador amb pols de Bessel.....	78
Fig. 5-26: gràfic d'un senyal quadrat d'entrada, integrador amb pols LQR.....	79
Fig. 5-27: gràfic d'un senyal triangular d'entrada, integrador amb pols de Bessel.....	80
Fig. 5-28: gràfic d'un senyal triangular d'entrada, integrador amb pols LQR.....	80
Fig. 5-29: gràfic d'un senyal quadrat d'entrada, integrador amb pols de Bessel.....	81





## **Índex de taules**

Taula 2-1: paràmetres del sistema.....	7
Taula 4-1: definició de regles.....	56



## Glossari de termes.

EUPMT	Escola Universitària Politècnica de Mataró
$m$	Massa del pèndol
$M$	Massa del carro
PFC	Projecte Final de Carrera
TFC	Treball Final de Carrera
$b$	Fricció entre el carro i les guies
$d$	Coefficient d'esmortiment del pèndol
$F$	força
$g$	Gravetat
$I$	Moment d'inèrcia
$K$	Valor del producte de realimentació per retorn d'estat
$K_d$	Valor del producte de realimentació per retorn d'estat discret
$K_a$	Valor del producte de realimentació augmentat
$K_{da}$	Valor del producte de realimentació augmentat discret
$L$	Valor del producte de realimentació observador
$L_d$	Valor del producte de realimentació discret observador discret
$\theta$	Angle que forma el pèndol respecte a la vertical
$\dot{\theta}$	velocitat angular
$x$	Posició lineal de carro
$\dot{x}$	Velocitat lineal del carro



# 1. Objectius

## 1.1. Propòsit

Obtenir coneixements sobre mètodes de control avançat.

## 1.2. Finalitat

Conèixer millor l'entorn Matlab per aplicar mètodes de control.

## 1.3. Objecte

Treballar sobre la maqueta de Feedback plc que consisteix en un pèndol invertit Digital Pendulum Controller 33-201. El desenvolupament del estudi i l'aplicació dels disseny es farà sobre Matlab i Simulink.

## 1.4. Abast

Disseny de controladors d'espai-estat i lògica difusa.

## 1.5 Formulació del problema

Aquest treball és una continuació d'un anterior fet a l'EUPMT que ja va tractar alguns dels problemes que pot plantejar el pèndol invertit. Aquí es tracta de millorar i evolucionar la tasca començada i planteja qüestions addicionals no tractades anteriorment.

El control del pèndol invertit és una tasca que es proposa molt sovint com a projecte amb la finalitat d'aplicar els diferents teories de control que s'imparteixen a les escoles universitàries de caràcter tecnològic. Es pot dir que és un problema clàssic dins de la teoria de control.

És objecte de desenvolupament d'un projecte perquè la seva consecució té parts que són interessants ja que aquest exercici fa al estudiant fer un estudi físic del sistema, aplicar lleis mecàniques i posar en marxa diferents tipus de control donant l'oportunitat de posar en pràctica diferents àrees d'estudi.

S'haurà de saber interpretar l'aplicació de forces sobre la maqueta del pèndol i reflectir-ho bé a les equacions que ens permetran obtenir els controladors adients.

Com se sap el pèndol té dos punts d'equilibri: un a baix estable que es situa al  $180^\circ$  o  $\pi$  radians i un a dalt a  $0^\circ$  o zero radians inestable. L'objectiu del control és obtenir un mètode que ens faciliti l'estabilitat, mitjanant l'aplicació d'una força, en el punt  $0^\circ$ .

Altre dels problemes que es planteja es aixecar el pèndol, ja que el exercici no consta només de poder-lo controlar una vegada està a dalt si no que també s'ha de aixecar i fer un switch en el moment adequat per poder passar d'un exercici (aixecar el pèndol) a un altre (mantenir-lo en el punt inestable però controlat).

### **1.5.1 Objectiu**

L'objectiu és aplicar aquesta força de control a través de mètodes basats en el disseny de espai-estat (retorn d'estat, disseny d'observador, control integral) i lògica difusa.

## **1.6 Contingut del projecte**

Capítol 1: Es proposen els objectius del present treball.

Capítol 2: descripció de l'entorn de treball i modelització del sistema. Es modelitzarà el sistema partint d'un estudi del diagrames de forces expressant-lo de forma matemàtica, es linealitzarà sobre el punt d'equilibri i s'obindrà un model continu ja simplificar per finalment aconseguir la discretització del sistema proposat.

Capítol 3: Es dissenyaran els diferents estratègies de control proposats per aconseguir l'estabilització sobre el punt desitjat: retorn d'estat, observador i control integral.

Capítol 4: Es dissenyarà la lògica difusa per aixecar el pèndol des de el punt d'equilibri estable, es a dir  $180^\circ$ , fins al punt d'equilibri inestable,  $0^\circ$ . Es farà per mitjà del control difús o fuzzy.

Capítol 5: Implementació dels diferents controladors dissenyats sobre el sistema real.

Capítol 6: Es desenvoluparan les conclusions generals sobre la totalitat del projecte i es proposaran futurs treballs o ampliacions del mateix.

Capítol 7: Bibliografia

## 2. Descripció i modelatge del sistema

### 2.1. Descripció del pèndol invertit

#### 2.1.1 Actuator

El pèndol invertit és un sistema format per un carro que es mou per uns rails horitzontalment. Del carro penja un pèndol a cada banda del mateix que giren lliurement respecte al carro però solidàriament entre ells. El pèndol té uns sensors que informen de la posició angular i la posició del carro dins dels rails.

El proveïdor del pèndol és Feedback plc. i el model és *DIGITAL PENDULUM MECHANICAL UNIT 33 – 200*.



*Fig. 2-1: unitat mecànica del pèndol*

Feedback dona una limitació degut al actuator; la tensió que es pot aplicar al motor DC que mou el carro a través d'engrenatges a d'estar dins d'una tensió mínima de  $-2,5V$  i una màxima de  $+2,5V$  que es traduiran a una força de  $-20N$  i  $+20N$

#### 2.1.2 Driver

El driver o caixa de control *DIGITAL PENDULUM CONTROLLER 33 – 201* a la part frontal té el botó de *power*, el botó *start* que permet que el PC obtingui el control del pèndol i tres LEDs que indiquen *ready*, *internal fuse –* i *internal fuse +*.

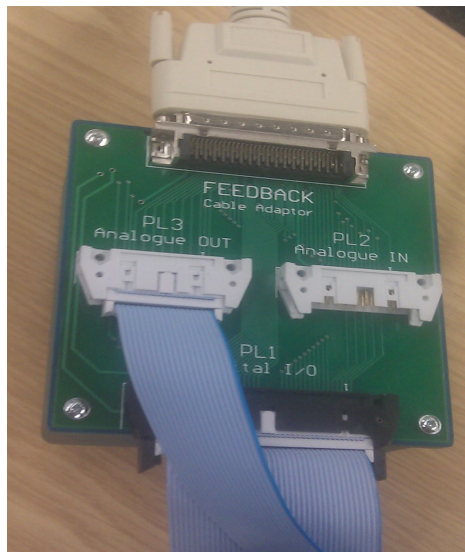
A la part posterior té el bus de dades que fa que el PC i el pèndol puguin interactuar.



*Fig. 2-2 unitat de control del pèndol*

### **2.1.3 Adquisició de dades**

La targeta d'adquisició que proporciona Feedback és una Advantech PCI-1711 amb un bus de dades de 16 bits i porta un adaptador que actua entre en driver i la pròpia targeta d'adquisició de dades.



*Fig. 2-3: adaptador per la targeta Advantech PCI-1711 i el driver DIGITAL PENDULUM CONTROLLER 33 – 201*



### 2.1.4 Controlador

La implementació del controlador es realitzarà amb un PC amb S.O. Windows XP que incorpora la tarja d'adquisició de dades descrita anteriorment. El conjunt de programes són el *Real Time Workshop*, *Real Time Windows Target*, *MATLAB* i *SIMULINK*.

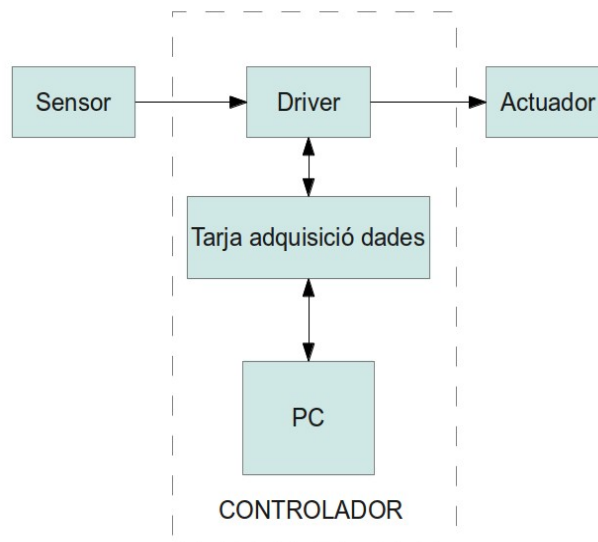


Fig. 2-4: diagrama de blocs controlador

## 2.2. Model matemàtic del sistema

La modelització és el primer pas del disseny d'un sistema de control, quan millor sigui el coneixement del sistema millor serà el resultat del control.

Per deduir el comportament del sistema es farà a partir de les lleis físiques que el regeixen: lleis del moviment de Newton. Per obtenir el model s'utilitzarà la segona llei de Newton aplicada a un diagrama de forces que actuen sobre el sistema.

Fruit del desenvolupament d'aquest estudi s'arribarà a dues equacions de segon ordre i d'aquestes es trobaran les *equacions d'estat*, formades per quatre equacions de primer ordre no lineals.

A continuació s'estudiarà una instrucció que, executada al entorn Matlab, es proporcionarà la linealització del sistema; la instrucció *linmod*. La linealització es fa amb el pèndol aixecat, es a dir al voltant dels zero graus.

### 2.2.1. Equacions diferencials

Les variables que hem de considerar per obtenir el sistema d'equacions que expressi el

comportament del pèndol són  $\theta$  i  $x$ .

Si es descriu el pèndul com un cos dos dimensions podem utilitzar les equacions fonamentals del moviment pla d'un cos rígid :

$$\sum F_i = m \cdot a_i \quad (2.1)$$

$$\sum F_j = m \cdot a_j \quad (2.2)$$

$$\sum M_G = I \alpha_g \quad (2.3)$$

Les equacions 2.1 i 2.2 són la segona llei de Newton per a les components horitzontals i verticals de la força  $F$ . L'equació 2.3 deriva també de la mateixa llei i indica que la suma de moments  $M$  de las forces que actuen sobre un cos rígid al voltant d'un punt  $G$ , és igual al moment d'inèrcia per l'acceleració angular  $\alpha$  al voltant d'un cos rígid.

En quant a la deducció de les fórmules que descriuen el sistema s'ha de dir que s'ha comprovat experimentalment que si s'aplica una tensió positiva al carro, aquest es mou a la esquerra del eix de coordenades  $X$  si ens imaginem el carro sobre un eix cartesià. I que la lectura que ens proporciona el sensor sobre la posició del angle es positiva si aquest últim gira en el sentit horari.

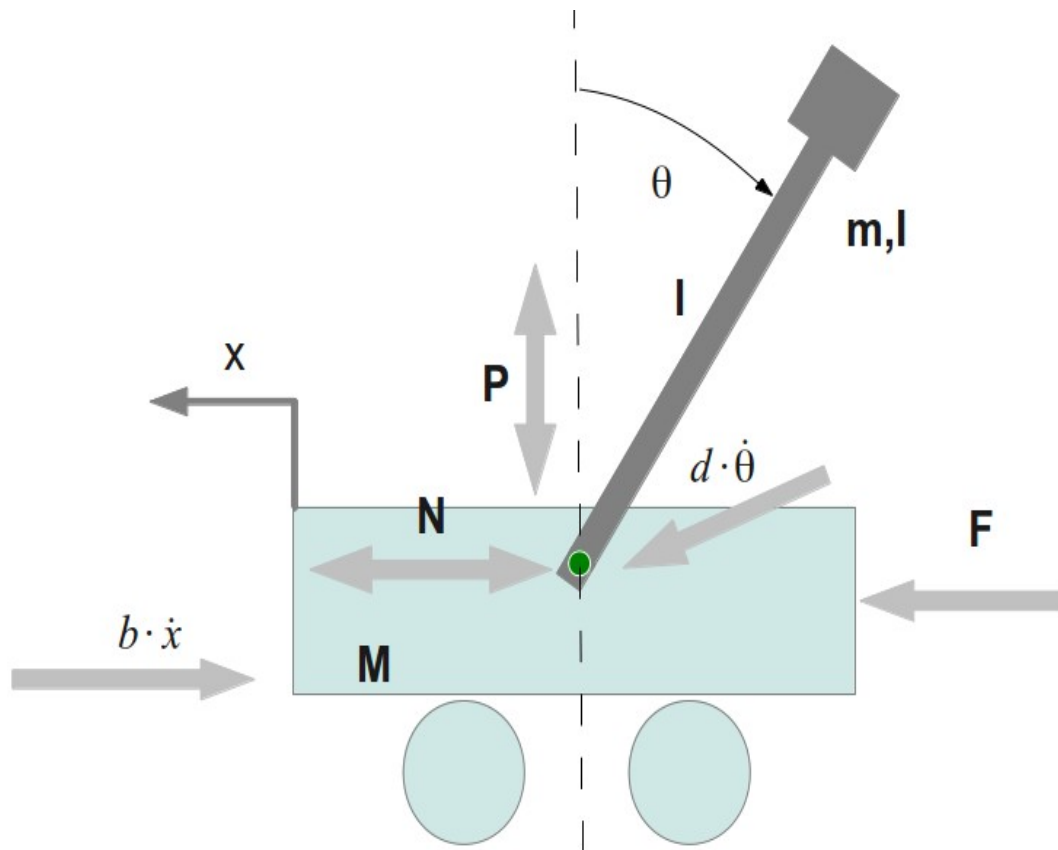


Fig. 2-5: diagrama de cos lliure del pèndol invertit

A la figura 2-5 es pot observar el diagrama de forces que actuen sobre el sistema, i a la taula 2-1 la definició dels paràmetres.

Paràmetre	Descripció	Valor
<b>g</b>	Gravetat	9,81 m/s <sup>2</sup>
<b>I</b>	Longitud de la barra del pèndol	0,4 m
<b>M</b>	Massa del carret	2,4 kg
<b>m</b>	Massa del pèndol	0,23 kg
<b>i</b>	Moment d'inèrcia del pèndol	0,099 kg·m <sup>2</sup>
<b>bb</b>	Coefficient de fricció del carret	0,05 Ns/m
<b>d</b>	Coefficient d'esmoreïment del pèndol	0,005 Nms/rad

Taula 2-1: paràmetres del sistema

Si es divideix el diagrama de forces en dues parts s'obté:

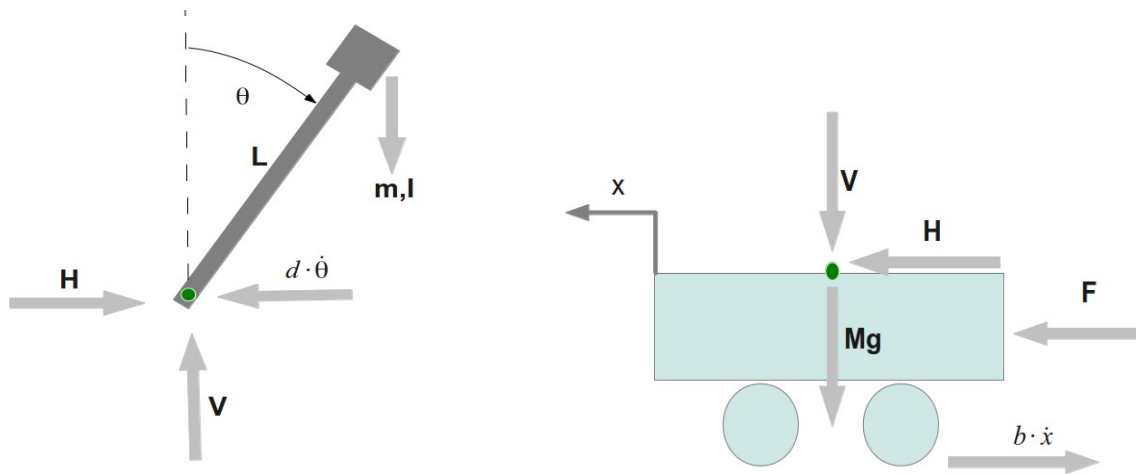


Fig. 2-6: diagrames de forces del pèndol i del carro

Sobre el pèndol s'observen les forces H i V actuant sobre l'articulació. Per obtenir l'acceleració horitzontal del pèndol s'ha de incloure la relació de  $x$  i  $\theta$  sobre aquest. De l'aplicació de la equació 2.1 s'obté:

$$-H = m \cdot \frac{d^2}{dt^2} \cdot (x - L \cdot \sin \theta) \quad (2.4)$$

Desenvolupant 2.4:

$$-H = m \cdot [+ \ddot{x} - L \cdot \ddot{\theta} \cdot \cos(\theta) + L \dot{\theta}^2 \sin(\theta)] \quad (2.5)$$

I de 2.2 s'obté:

$$V = m \cdot \left[ \frac{d^2}{dt^2} \cdot (L \cos \theta) + g \right] \quad (2.6)$$

Desenvolupant 2.6:

$$V = m \cdot [-L \cdot \ddot{\theta} \cdot \sin(\theta) - L \dot{\theta}^2 \cos(\theta) + g] \quad (2.7)$$

Sobre el centre de gravetat del pèndol actuen tres moments deguts a les forces F i H més la fricció de la pròpia articulació. De l'adequació 2.3 es dedueix:

$$I\ddot{\theta} = L \cdot V \cdot \sin(\theta) - L \cdot H \cdot \cos(\theta) - d\dot{\theta} \quad (2.8)$$

Aplicant 2.1 al carro s'obté:

$$M \cdot \ddot{x} = F + H - b \cdot \dot{x} \quad (2.8)$$

Substituint en 2.8 H:

$$M \cdot \ddot{X} = F + m \cdot [-\ddot{x} + L \cdot \ddot{\theta} \cdot \cos(\theta) - L \dot{\theta}^2 \sin(\theta)] - b \cdot \dot{x} \quad (2.9)$$

$$M \cdot \ddot{x} = F - m \cdot \ddot{x} + m \cdot [L \cdot \ddot{\theta} \cdot \cos(\theta) - L \dot{\theta}^2 \sin(\theta)] - b \cdot \dot{x} \quad (2.10)$$

$$\dot{x} \cdot (M + m) = F + m \cdot [L \cdot \ddot{\theta} \cdot \cos(\theta) - L \dot{\theta}^2 \sin(\theta)] - b \cdot \dot{x} \quad (2.11)$$

$$\ddot{x} = \frac{F + m \cdot [L \cdot \ddot{\theta} \cdot \cos(\theta) - L \dot{\theta}^2 \sin(\theta)] - b \cdot \dot{x}}{(M + m)} \quad (2.12)$$

Substituint en 2.8 H i V:

$$I\ddot{\theta} = L \cdot m \cdot \sin \theta \cdot [-L \cdot \ddot{\theta} \cdot \sin \theta - L \dot{\theta}^2 \cos \theta + g] - L \cdot m \cdot \cos \theta \cdot [-\ddot{x} + L \cdot \ddot{\theta} \cdot \cos \theta - L \dot{\theta}^2 \sin \theta] - d\dot{\theta} \quad (2.13)$$

$$I\ddot{\theta} = L \cdot m \cdot \theta [-m \cdot L \cdot \sin \theta \cdot \ddot{\theta} - m \cdot L \cdot \cos \theta \cdot \dot{\theta}^2 + m \cdot g] - m \cdot L \cdot \cos \theta \cdot [-\ddot{x} + L \cdot \cos \theta \cdot \ddot{\theta} - L \cdot \sin \theta \cdot \dot{\theta}] \quad (2.14)$$

$$\ddot{\theta} \cdot (I + m \cdot L^2) = L \cdot m \cdot g \cdot \sin \theta + L \cdot m \cdot \cos \theta \cdot \ddot{x} - d \cdot \dot{\theta} \quad (2.15)$$

$$\ddot{\theta} = \frac{L \cdot m \cdot g \cdot \sin \theta + L \cdot m \cdot \cos \theta \cdot \ddot{x} - d \cdot \dot{\theta}}{(I + m \cdot L^2)} \quad (2.16)$$

Les equacions 2.12 i 2.16 formen un sistema de dues equacions i dues incògnites que descriuen el comportament del pèndol.

### 2.2.1. Simulació del sistema modelitzat

Per comprovar el comportament de les equacions es munta un diagrama de blocs al Simulink de Matlab.

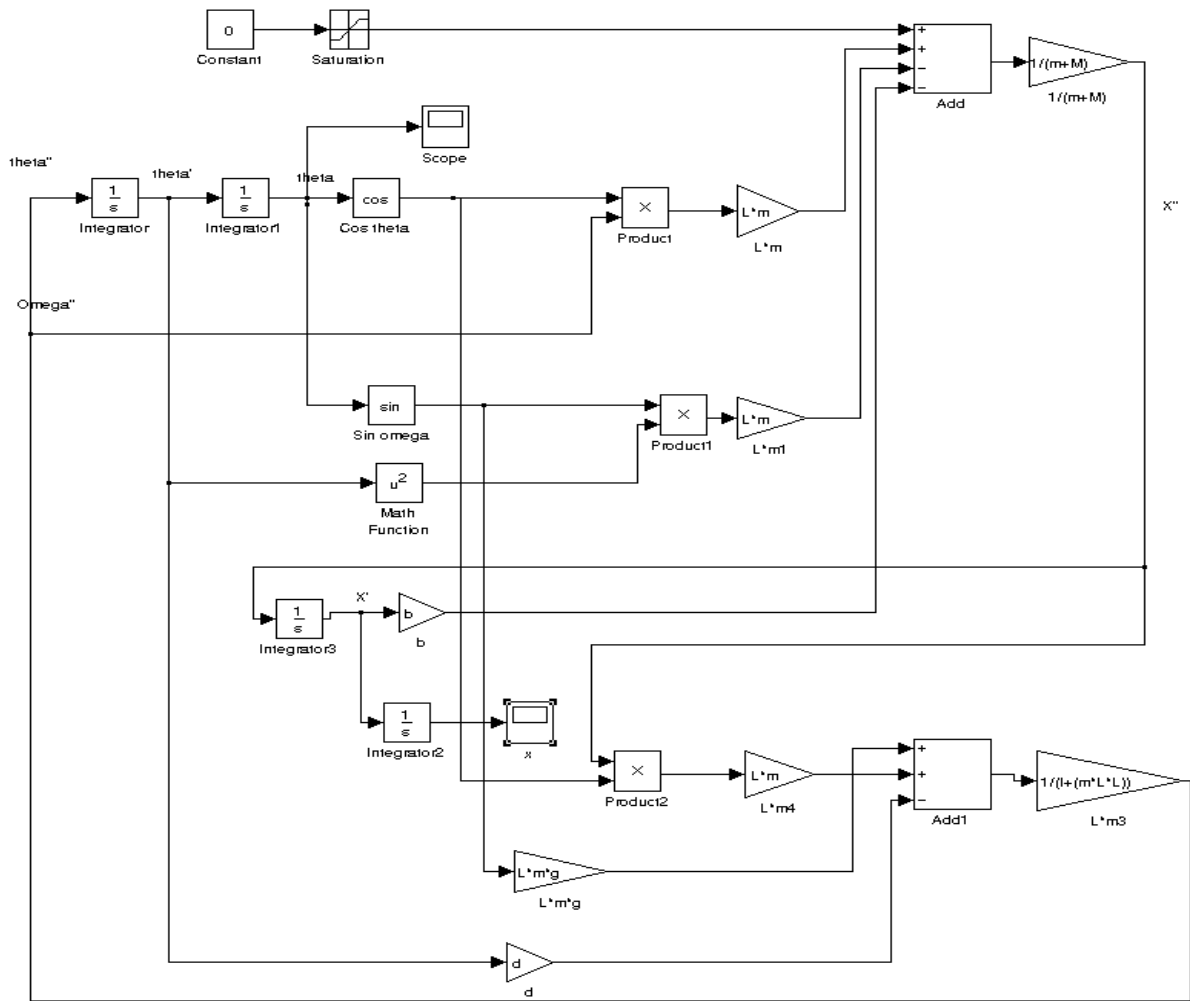


Fig. 2-7: model No lineal construït al Simulink

La resposta de la simulació del sistema per la posició del carro:

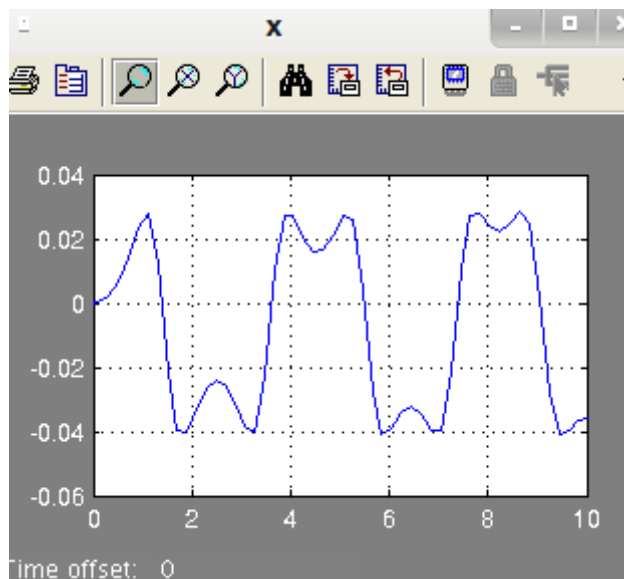


Fig. 2-8: resposta de la simulació del sistema per la posició del carro

La resposta de la simulació del sistema per la posició del pèndol:

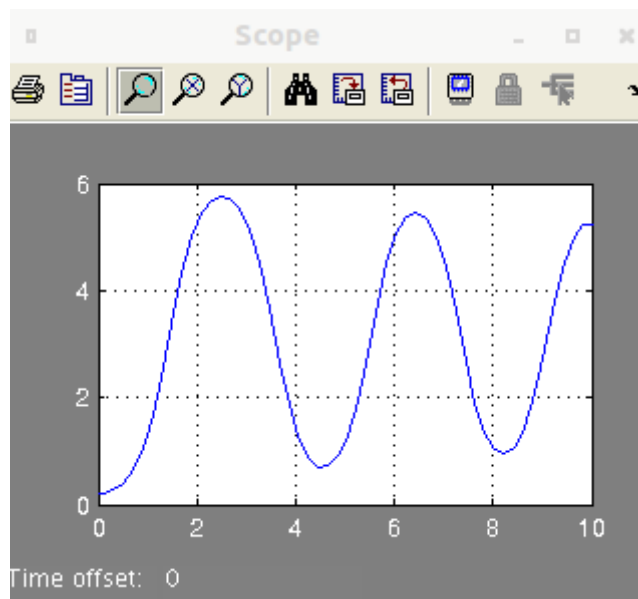


Fig. 2-9: resposta de la simulació del sistema per la posició del pèndol

Els resultats de la simulació són els que cal esperar. El pèndol al deixar-lo anar, figura 2-9, des de una posició de 0,2 radians ha de oscil·lar cada vegada amb una amplitud menor. Aquesta oscil·lació a més ha de fer que el carro es mogui d'un costat al altre com mostra la figura 2-8.

### 2.3. Equació d'estat

El moviment de qualsevol sistema dinàmic finit es pot expressar com un conjunt d'equacions diferencials ordinàries de primer ordre, es coneix com una *representació de variable d'estat*. Les equacions diferencials de primer ordre es poden representar mitjançant una notació matricial (veure Franklin, Powell i Emami-Naeini capítol 6).

$$\dot{x} = A \cdot x + B \cdot u \quad (2.17)$$

$$y = Cx + Du \quad (2.18)$$

$\dot{x}$  és un vector que representa el estat del sistema i està format per  $n$  elements si el sistema és de ordre  $n$ .

$u$  és el vector d'entrada i conté  $m$  elements

$y$  és el vector de sortida i està format per  $p$  elements.

$A$  és la matriu del sistema, de dimensió  $n \times n$ .

$B$  és la matriu d'entrada, de dimensió  $n \times m$ .

$C$  és la matriu de sortida, de dimensió  $n \times p$ .

$D$  és una matriu de dimensió  $p \times m$  i normalment el seu valor el zero.

Fins ara la representació de la posició del carro s'havia parametritzat amb  $x$ , a partir d'ara per evitar confusions  $x = r$ .

El vector  $\dot{x}$  està format per quatre variables:  $r$ ,  $\theta$ ,  $\dot{r}$ ,  $\dot{\theta}$ . A on  $r$  és la posició del carro,  $\theta$  és la posició del angle,  $\dot{r}$  és la velocitat de desplaçament de carro i  $\dot{\theta}$  és la velocitat angular.

$$\dot{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{bmatrix} \quad (2.19)$$

La força que s'aplica al carro és  $F$  per tant el vector  $u = F$

A la sortida del sistema volem obtenir  $r$ ,  $\theta$ ,  $\dot{r}$ ,  $\dot{\theta}$  per tant,

$$y = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (2.20)$$

De l'equació 2.19 es pot deduir que:

$$\dot{x}_2 = x_4 \quad (2.21)$$

$$\dot{x}_1 = x_3 \quad (2.22)$$

$$\dot{x}_3 = \ddot{r} \quad (2.23)$$

$$\dot{x}_4 = \ddot{\theta} \quad (2.24)$$

Per obtenir les matrius s'ha de treballar amb un sistema lineal i les equacions del sistema tenen funcions no lineals com son el  $\sin \theta$  i  $\cos \theta$ . Es pot linealitzar el sistema per mitja de la sèrie de Taylor o per la instrucció *linmod* de Matlab que ens permetrà obtenir directament les matrius del sistema linealitzat.

Prèviament a aquesta treball es va fer un altre TFC sobre el pèndol en el qual es desenvolupava el



procediment de linealització simplificant les funcions no lineals. Per tant, com en aquest projecte es tracta també d'aprendre a utilitzar les eines de que disposa Matlab es durà a terme el mètode del *linmod*.

*linmod* obté els models lineals de sistemes d'equacions diferencial ordinàries (EDO).

$[A, B, C, D] = \text{linmod}('sys')$ ,

obté el model d'estat-espai lineal del sistema d'equacions diferencials ordinàries descrit amb 'SYS' (nom del diagrama de blocs), quan les variables d'estat i les entrades es posen als valors predeterminats especificats en el diagrama de blocs.

$[A, B, C, D] = \text{linmod}('sys', x, u)$  permet que el vector d'estat,  $x$ , i d'entrada,  $u$ , es determini. El model lineal s'obté en el punt d'operació determinat..

El primer pas es agafar les equacions 2.23 i 2.24 i construir, a partir d'aquestes, el model no lineal en un diagrama de blocs al Simulink de Matlab. La figura 2-10, més a baix, no es exactament igual que la 2-7, hi han uns petits blocs anomenats  $F$ ,  $out1$ ,  $out2$ ,  $out3$  i  $out4$ . Aquests són necessaris per executar la instrucció *linmod* ja que és una de les maneres d'identificar entrades i sortides del sistema al Matlab, al capítol 3 es detallarà més aquest aspecte.

$$\ddot{x} = \frac{F + m \cdot [L \cdot \ddot{\theta} \cdot \cos(\theta) - L \dot{\theta}^2 \sin(\theta)] - b \cdot \dot{x}}{(M + m)} \quad (2.23)$$

$$\ddot{\theta} = \frac{L \cdot m \cdot g \cdot \sin \theta + L \cdot m \cdot \cos \theta \cdot \ddot{x} - d \cdot \dot{\theta}}{(I + m \cdot L^2)} \quad (2.24)$$

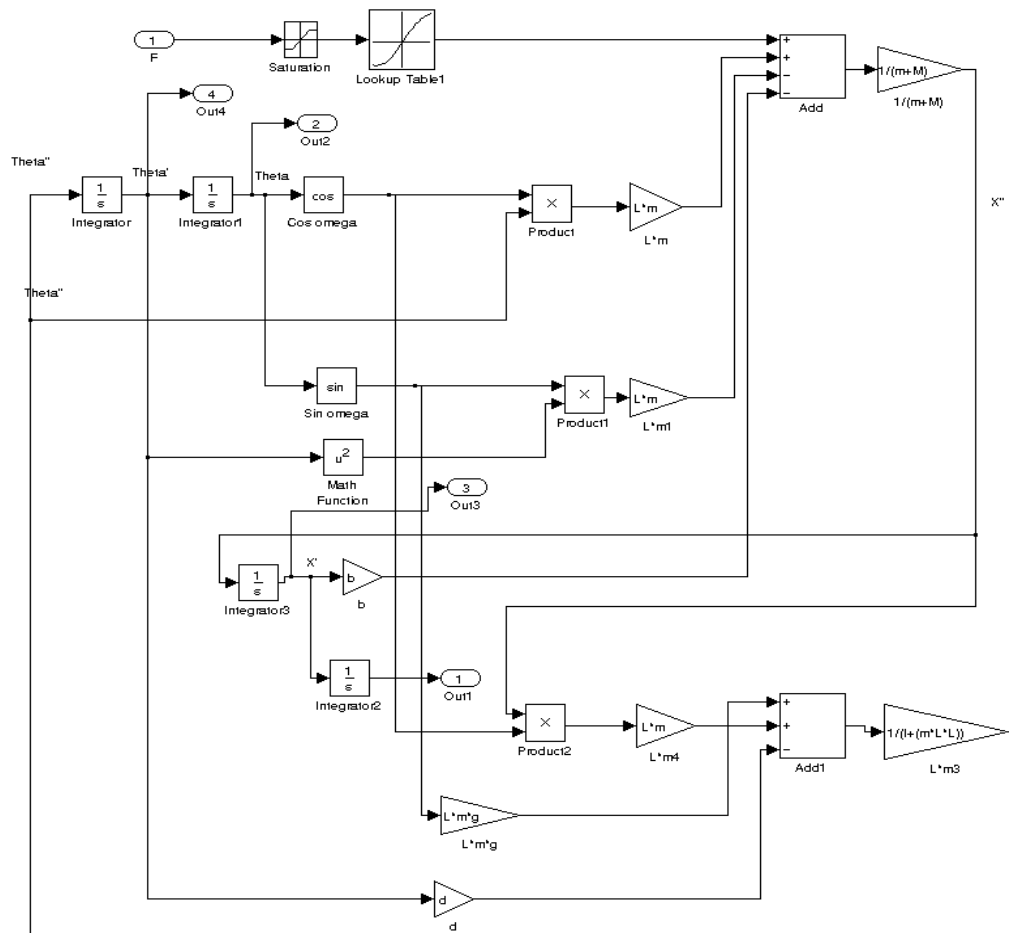


Fig. 2-10: model No lineal construït al Simulink

Es pot observar que el diagrama de la figura 2-10 té un saturador (bloc saturation) que serveix perquè no es sobrepassin els  $\pm 2,5V$  que recomana aplicar el fabricant com a màxim. A més té un bloc anomenat Lockup table, la raó per la qual apareix aquest bloc es desenvoluparà al capítol 5.

Es carregen els valors dels paràmetres al entorn de treball Matlab:

$$M=2.4$$

$$I=0.099$$

$$b=0.05$$

$$L=0.4$$

$$d=0.005$$

$$m=0.23$$

$$g=9.81$$

$$T_s=0.01$$

A continuació s'executa l'ordre `linmod`:

$$[A, B, C, D] = \text{linmod}('modelnolin-taula', [0; 0; 0; 0], 0)$$

A on  $A$ ,  $B$ ,  $C$ ,  $D$  són les matrius del sistema; `linmod` la instrucció; `modelnolin_taula` el nom del arxiu de Simulink amb extensió `.mdl`; `[0;0;0;0]` el estat de les variables a on es vol linealitzar tenint en compte, sobre tot, que el punt de linealització és quan el angle del pèndol sigui zero; i  $0$  l'entrada del sistema.

Els valors de retorn són:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0,2381 & -0,0195 & -0,0013 \\ 0 & 6,8073 & -0,0132 & -0,0377 \end{bmatrix} \quad (2.25)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 2,4403 \\ -1,6533 \end{bmatrix} \quad (2.26)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.27)$$

$$D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.28)$$

## 2.4. Model discret

Fins ara s'ha treballat sobre el model continu però Matlab treballa sobre models discrets. Es a dir agafar mostres de l'estat del sistema cada cert temps perquè es pugui treballar amb aquestes dades. A aquesta periodicitat d'obtenció de mostres se l'anomena temps de mostreig.

Per poder saber quin temps de mostreig és el ideal no hi ha cap mètode concret, s'han de fer proves mitjançant simulacions i en el sistema real fins trobar un que satisfaci un control òptim.

Però orientativament es pot escollir un temps de mostreig si ens fixem en les dinàmiques de les

figures 2-8 i 2-9. En aquestes figures s'observa que un cicle complet té uns 4 segons, els temps de mostreig no ha de ser molt excediu, agafar masses mostres, ni quedar-se curt. Si escollim mostres cada 10 centèsimes, per un cicle de 4 segons, s'obtidran les suficients dades com per reconstruir la senyal i no serà una quantitat excessiva.

Per tant el temps de mostreig escollit per aquest sistema és de 0,01 segons. Llavors les dades que obtindrà Matlab de l'estat del sistema tindran una periodicitat de 0,01 segons.

La discretització del sistema s'obté a partir del Matlab i s'utilitza l'ordre c2d:

$$\text{sysd} = \text{c2d}(\text{sysc}, Ts, \text{METHOD})$$

On *sysc* representa el sistema continu, *Ts* temps de mostreig, *sysd* el sistema discret i *METHOD* un dels diferents mètodes pel qual es vol aconseguir: zoh, foh, imp...

Primer es construeix el sistema continu:

$$\text{sys} = \text{ss}(A, B, C, D)$$

A continuació el discretitza el sistema:

$$\text{sysd} = \text{c2d}(\text{sys}, Ts)$$

S'obtenen les matrius discretitzades:

$$[Ad \ Bd \ Cd \ Dd] = \text{ssdata}(\text{sysd})$$

Retorna:

$$Ad = \begin{bmatrix} 1 & 0 & 0,1 & -0 \\ 0 & 1,0003 & -0 & 0,01 \\ 0 & 0,0024 & 0,9998 & -0 \\ 0 & 0,0681 & -0,0001 & 1 \end{bmatrix} \quad (2.29)$$

$$Bd = \begin{bmatrix} 0,0001 \\ 0,0001 \\ 0,0244 \\ 0,0165 \end{bmatrix} \quad (2.30)$$

$$Cd = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.31)$$

$$Dd = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.32)$$



### 3. Estratègia de control

Es aquest capítol es desenvoluparan diferents tècniques de control basades en el model matemàtic del sistema.

Primer es comprovarà que el sistema sigui controlable i observable. A continuació es farà una selecció de la localització dels pols per a un bon disseny. Es provaran tant els pols de Bessel com els obtinguts pel mètode LQR (Regulador Lineal Quadràtic) segons sigui el cas.

Una vegada s'han obtinguts els pols es calcularà les matrius de control  $K$  per al sistema continu i  $Kd$  per al sistema discret i es veuran les respostes per al control de retorn d'estat.

Després es calcularan les matrius  $L$  i  $Ld$  per al sistema continu i discret respectivament d'un observador. Amb la suposició de que només podem accedir a dues variables del sistema en compte de les quatre i obtingudes les matrius que permetran fer la simulació, es simularà.

Per últim s'augmentarà el sistema un ordre més, es a dir s'afegirà un integrador per aconseguir un seguiment robust. Aquest tindrà la finalitat de reduir l'error en una de les variables, en aquest cas la variable de posició. Una vegada s'aconsegueixin les matrius augmentades es calcularà la nova matriu de realimentació  $Ka$  i es podrà simular el sistema.

Matlab serà l'eina per calcular els controladors. Per poder comprovar els resultats mitjançant la simulació del sistema s'utilitzarà Simulink amb el que es podrà simular tant els sistemes lineals com els no lineals.

#### 3.1. Disseny de sistema de control

##### 3.1.1. Controlabilitat i observabilitat

Per dissenya el control el primer que hem de saber es sí el sistema és controlable,

Per a un sistema que compleix amb les equacions 2.17 i 2.18 amb el par de matrius (A, B), es diu que és controlable si per a qualsevol estat inicial  $x(0) = x_0 \in \mathbb{R}^n$  i qualsevol estat final  $x_1 \in \mathbb{R}^n$  existeix una entrada que transfereix el estat  $x_0$  a  $x_1$  en temps finit. En cas contrari es diu que el par (A,B) no és controlable [1].

Segons aquest teorema podem dir que en determinades condicions es pot trobar una matriu de realimentació K que porti el sistema allà a on volem.

L'equació d'estat (2.17 i 2.18) és observable si per a qualsevol estat inicial  $x(0)$  (desconegut), hi ha un temps finit  $t_1$  tal que el coneixement de l'entrada  $u$  i la sortida  $y$  sobre el interval  $[0, t_1]$  és suficient per determinar en forma única l'estat

inicial  $x(0)$ . En cas contrari el sistema no és observable [1].

La controlabilitat del sistema es pot calcular o comprovar amb Matlab i les instruccions necessàries són:

$$co = ctrb(Ad, Bd); \quad (3.1)$$

$$Controlabilitat = rank(co) \quad (3.2)$$

$$ob = obsv(Ad, Cd); \quad (3.3)$$

$$Observabilitat = rank(ob) \quad (3.4)$$

El valor que retorna és:

$$Controlabilitat = 4 \quad (3.5)$$

$$Observabilitat = 4 \quad (3.6)$$

Si les matrius de controlabilitat i observabilitat han de ser de  $n \times n$ , en aquest cas de  $4 \times 4$ , el seu rang és 4. Per tant el sistema es observable i controlable.

### 3.1.2. Assignació del pols

Per l'elecció de pols es podem obtenir pel mètode LQR o bé consultar les taules de ITEA o BESSEL [3].

La taula de Bessel ens donen un pols que tenen una forma determinada, sense sobre-impuls, però el problema que presenten es que són massa lents al igual que els de ITEA. Consultant documentació es va trobar un treball que aconseguia uns pols determinats amb el mètode LQR [2]. I com que la maqueta de laboratori que s'utilitzava en aquest treball era la mateixa que la de l'EUPMT es van provar a les simulacions d'aquest projecte i els resultats van ser satisfactoris.

La instrucció LQR no solament pot trobar uns pols per treballar si no que ens dona una matriu  $K$  de realimentació. Tot i que es van fer servir els pols d'aquell treball la matriu  $K$  del present projecte es va calcular amb la instrucció acker de Matlab.

Les instruccions per l'obtenció de pols són:

$$x = 15000; \quad (3.7)$$

$$y = 5000; \quad (3.8)$$

$$Q = diag([x; 0; y; 0]); \quad (3.9)$$

$$R = 1500; \quad (3.10)$$



$$[K, S, e] = lqr(A, B, Q, R); \quad (3.11)$$

Sortida:

$$ans = -3.1811 + 3.0446i \quad (3.12)$$

$$ans = -3.1811 - 3.0446i \quad (3.13)$$

$$ans = -2.4446 + 0.4400i \quad (3.14)$$

$$ans = -2.4446 - 0.4400i \quad (3.15)$$

A on  $x$  és el factor de ponderació del carro i  $y$  el factor de ponderació del pèndol.

Per al control per retorn d'estat i el observador s'han fet servir els pols obtinguts per LQR i per el control integral LQR i Bessel.

El fet de que el pols de Bessel siguin massa lents es pot solucionar per obtenir la mateixa forma de resposta, però més ràpida o més lenta modificant el valor de  $\omega_o$  que pels valors nominals val 1rad/s i podem posar valors més alts (més grans que 1) o més baixos (menors que 1 rad/s). En el primer cas la resposta serà més ràpida i en el segon més lenta. Com que el que es vol és una resposta més ràpida es fa  $\omega_o = 3$  rad/s que vol dir substituir la  $s$  per  $s/\omega_o$ , es a dir multiplicar per 3 els valors nominals dels pols.

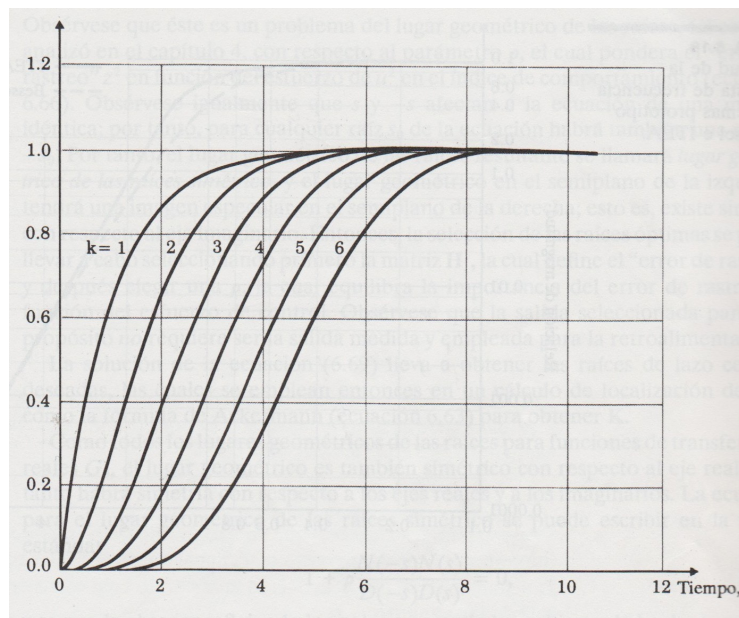


Fig. 3-1: Resposta esglaió dels sistemes de prototip de Bessel per  $\omega_o = 1 \text{ rad/s}$  [3].

Els pols de Bessel escollits són els de cinquè ordre ja que s'utilitzaran per al control integral, es a dir un ordre més. Que multiplicats per 3 són:

$$\begin{aligned}
 & -2.7792 \\
 & -1.7718 + 2.7216i \\
 & -1.7718 - 2.7216i \\
 & -2.5548 + 1.3281i \\
 & -2.5548 - 1.3281i
 \end{aligned}$$

### 3.2. Retorn d'estat

La tècnica de control per retorn d'estat parteixen de les equacions d'estat 2.17 i 2.18 que són representen el sistema en llaç obert.

$$\dot{x} = A \cdot x + B \cdot u$$

$$y = C \cdot x + D \cdot u$$

En el retorn d'estat es proposa una senyal de control lineal al estat:

$$u = r - K \cdot x \tag{3.16}$$

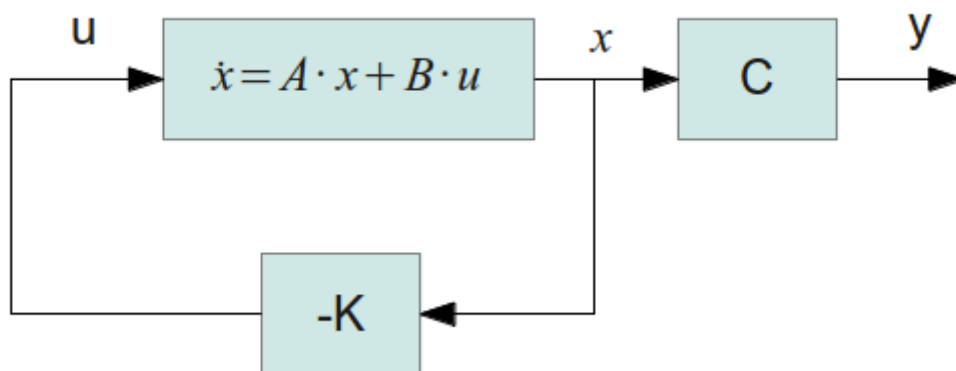


Fig. 3-2: diagrama de bolcs del control per retorn d'estats

La substitució de 3.16 en 3.17 tanca el llaç i obtenim el retorn d'estat:

$$\dot{x} = (A - B \cdot K) x + B \cdot r \tag{3.17}$$

$$y = C \cdot x + D \cdot u \quad (3.18)$$

### 3.2.1. Disseny de la matriu K

La instrucció que calcula la matriu  $K$  en Matlab s'anomena `acker`, la informació que ens dona la pròpia aplicació teclejant `help acker` és la següent:

$K = \text{Acker}(A, B, P)$  : Calcula la matriu o vector de guany  $K$  tal que el sistema d'una sola entrada  $\dot{x} = Ax + Bu$ , amb una llei de control  $u = -Kx$ , tingui els pols en bucle tancat en els valors especificats en el vector  $P$ . És una implementació de la fórmula de Ackerman.

Es pot veure un desenvolupament matemàtic aplicat de la fórmula d'Ackerman en el treball sobre el pèndol estudiat anteriorment a aquest a l'EUPMT [4].

Càlcul al Matlab de la matriu  $K$  per al sistema continu:

$$P_c = [-3.1811 + 3.0446i, -3.1811 - 3.0446i, -2.4446 + 0.4400i, -2.4446 - 0.4400i] \quad (3.19)$$

$$K = \text{acker}(A, B, P_c) \quad (3.20)$$

Valor de la matriu  $K$ :

$$K = [-7,3758 \quad 49,7310 \quad -8,3141 \quad 19,0434] \quad (3.21)$$

Muntatge del diagrama de blocs:

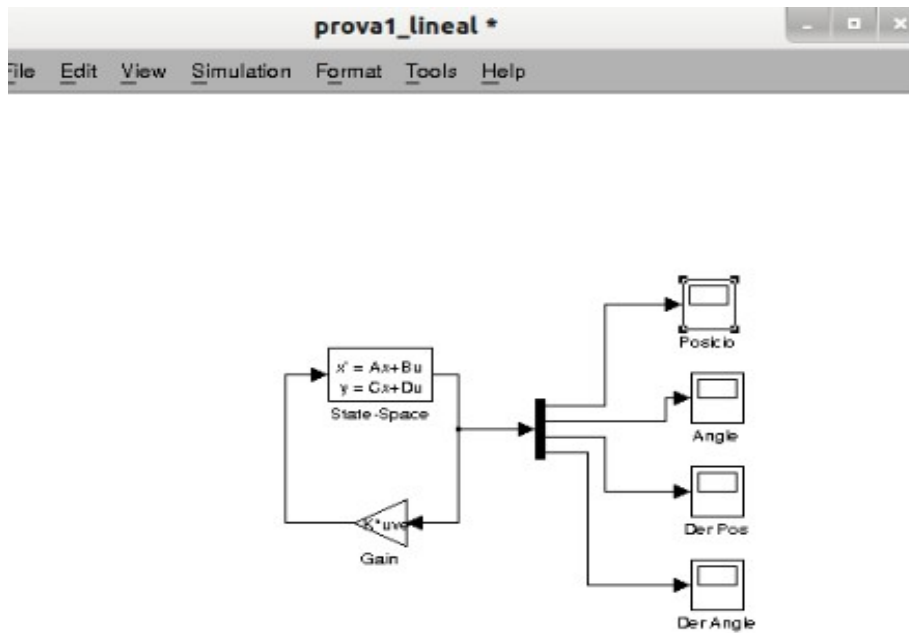


Fig. 3-3: diagrama de blocs del control per retorn d'estat, sistema continu linealitzat.

Configuració del bloc State-Space:

The screenshot shows the 'Function Block Parameters: State-Space' dialog box. The 'State Space' section displays the state-space model equations:  $dx/dt = Ax + Bu$  and  $y = Cx + Du$ . The 'Parameters' section includes input fields for matrices A, B, C, and D. The 'Initial conditions' field is set to [0;0.2;0;0]. The 'Absolute tolerance' is set to 'auto'. The 'State Name' field is empty. The dialog has 'OK', 'Cancel', 'Help', and 'Apply' buttons at the bottom.

Fig. 3-4: configuració bloc State-Space

A on A, B, C, D són las matrius d'estat i les condicions inicials son 0 per la posició, 0,2 radians per

l'angle, 0 per la derivada de la posició i 0 per la derivada de l'angle.

Configuració del bloc de guany  $K$ , Gain:

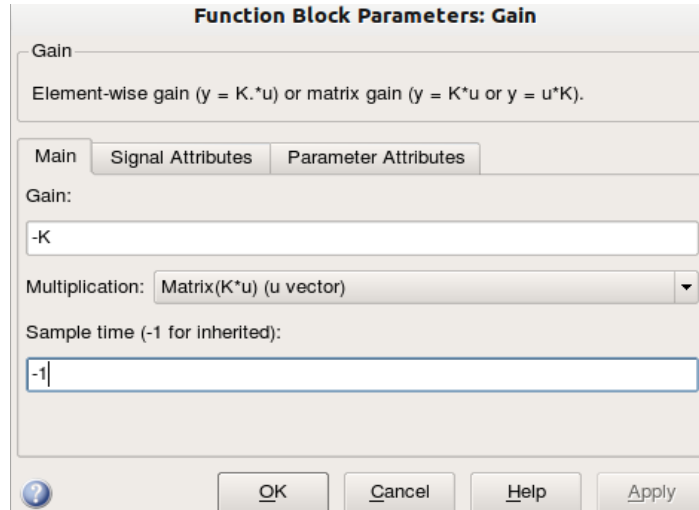


Fig. 3-5: configuració bloc Gain

El resultat de la simulació per cada variable:

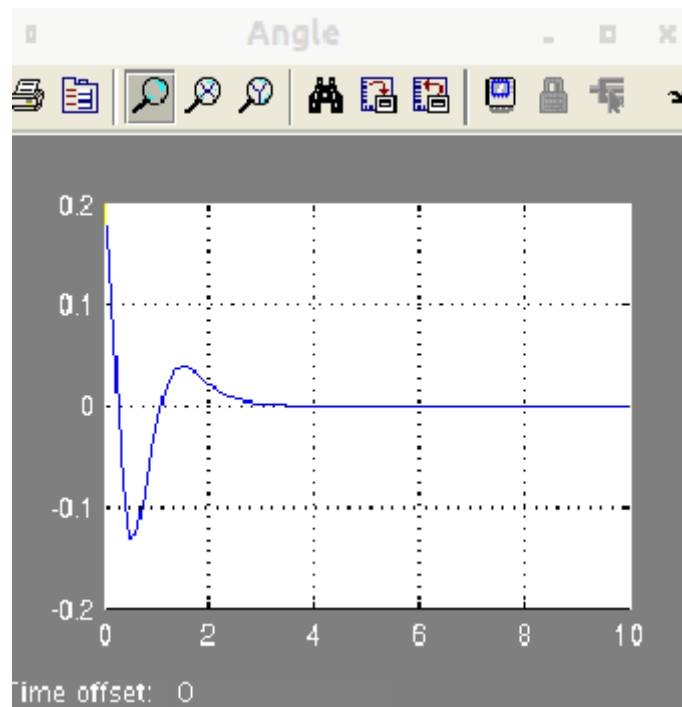


Fig. 3-6: resposta del angle

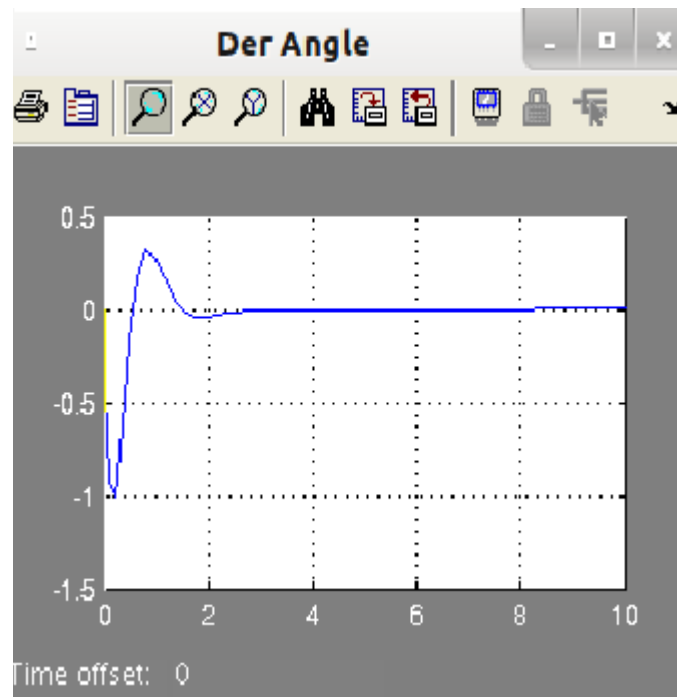


Fig. 3-7: resposta de la derivada del angle

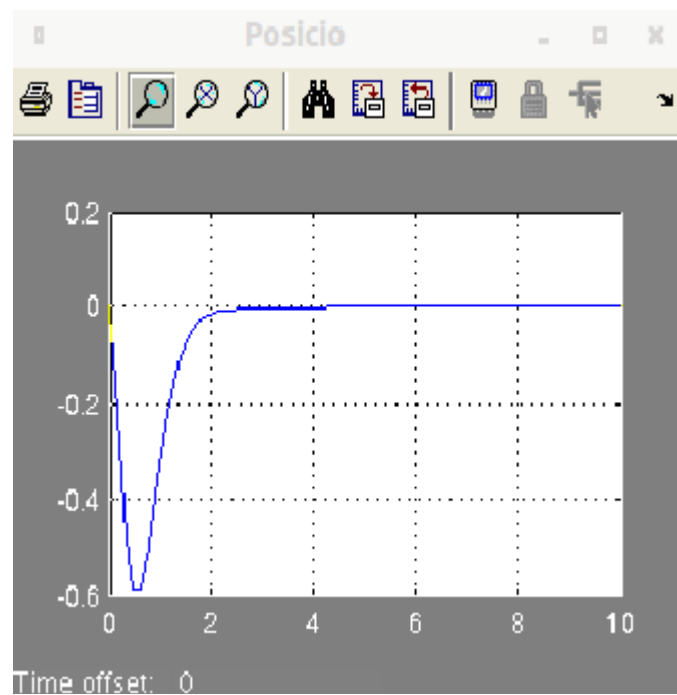


Fig. 3-8: resposta de la posició del carro

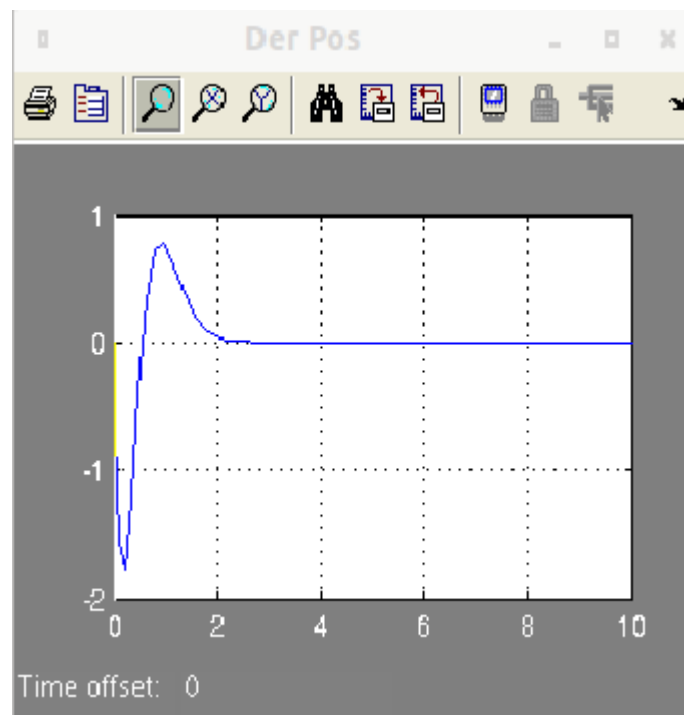


Fig. 3-9: resposta de la derivada de la posició del carro

Als resultats de la simulació es veu que el control per retorn d'estat es pot considerar efectiu. En el cas hipotètic de que el control es trobés el pèndol amb una desviació de 0,2 radians respecte al zero (figura 3-6) seria capaç de portar-lo fins al punt desitjat i mantenir-lo allà a on volem.

La figura 3-8 ens mostra com per poder controlar el pèndol fins portar-lo al punt desitjat la posició del carro es desplaça fins als 0,6 metres. Això pot representar un problema ja que el rail pel que corre el pèndol té una amplada d'un metre, 0,5 m. per cada costat ja que la posició inicial del control del sistema en quant a la posició serà sempre al centre de la longitud total del rail.

A les figures 3-7 i 3-9, derivada del angle i posició respectivament, s'observa que el comportament és coherent amb les variables ja comentades perquè aquestes han de mostrar activitat sempre que es produeixi un canvi de posició del sistema i deixar de tenir-la si no hi ha canvis.

El problema de la desviació pronunciada de la posició s'ha d'observar i tenir en compte però no es definitiu, s'ha de experimentar amb el sistema discretitzat i verificar què passa al sistema real ja que pot ser que a la aplicació practica no passi d'igual manera.

### 3.2.2. Disseny de la matriu $K_d$

El procediment pel que es troba  $K_d$  (matriu de realimentació per al sistema discretitzat) és semblant al continu però s'han de passar els pols utilitzats al sistema continu al pla  $\mathbb{Z}$  i utilitzar les matrius  $A_d$ ,  $B_d$ ,  $C_d$  i  $D_d$ . Per una banda tenim que els pols primer i segon són:

$$S_{12} = \xi \omega \pm j \omega \sqrt{1 - \xi^2} = -3.1811 \pm 3.0446i \quad (3.22)$$

I el tercer i quart són:

$$S_{34} = \xi \omega \pm j \omega \sqrt{1 - \xi^2} = -2.4446 \pm 0.4400i \quad (3.23)$$

El pas a pla  $Z$  :

$$Z = e^{-\xi \cdot \omega \cdot T} \pm \omega \cdot T \cdot \sqrt{1 - \xi^2} = r \pm \theta \quad (3.24)$$

Llavors r:

$$r = e^{-\xi \cdot \omega \cdot T} \quad (3.25)$$

I theta:

$$\theta = \omega \cdot T \cdot \sqrt{1 - \xi^2} \quad (3.26)$$

A on  $T$  és el temps de mostreig i per tant s'ha de substituir per 0,01, calculant:

$$Z_{12} = 0.975827253 \pm 0.0043i \quad (3.27)$$

$$Z_{34} = 0.968241 \pm 0.029i \quad (3.28)$$

Ara es pot obtenir la matrius de realimentació  $Kd$  per al sistema discret:

$$Pd = [0.9758 + 0.0043i \quad 0.9758 - 0.0043i \quad 0.9682 + 0.0290i \quad 0.9682 - 0.0290i] \quad (3.29)$$

$$Kd = place(Ad, Bd, Pd) \quad (3.30)$$

El valor retornat és:

$$Kd = [-0,68764 \quad 47,3421 \quad -7,8192 \quad 18,1307] \quad (3.31)$$



El diagrama de blocs que es poden proposar per muntar al Simulink i simular el sistema discret pot variar:

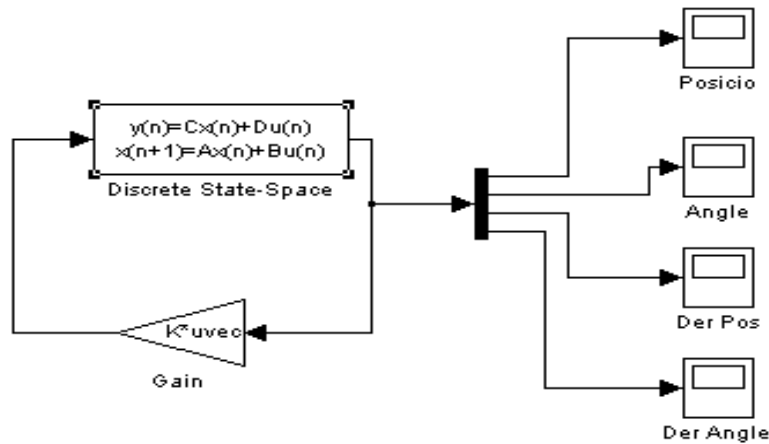


Fig. 3-10: proposta 1 diagrama de blocs sistema discretitzat

A la figura 3-10 es pot veure una de les propostes per simular el sistema discret. Al bloc Discret State-Space s'han d'introduir les matrius  $A_d$ ,  $B_d$ ,  $C_d$ , i  $D_d$  a més de posar les condicions inicials. La configuració dels blocs és molt semblant als del sistema continu. Al bloc Gain té com a valor el paràmetre  $-Kd$  (amb signe menys ja que la realimentació és sempre negativa).

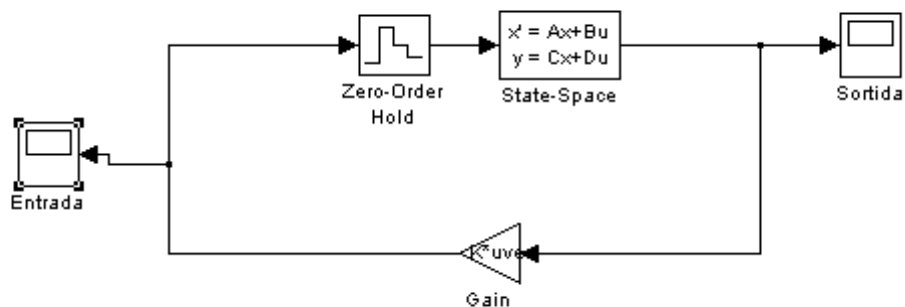


Fig. 3-11: proposta 2 diagrama de blocs sistema discretitzat

La proposta de la figura 3-11 és agafar la simulació del sistema continu amb les matrius contínues A, B, C i D, i posar a la entrada, abans de la realimentació, el bloc Zero-Order Hold amb un temps de mostreig de 0,01 (a la configuració del bloc *sample time*). Amb això el que es fa és que el sistema pren mostres cada 0,01 segons, es a dir es discretitza el sistema.

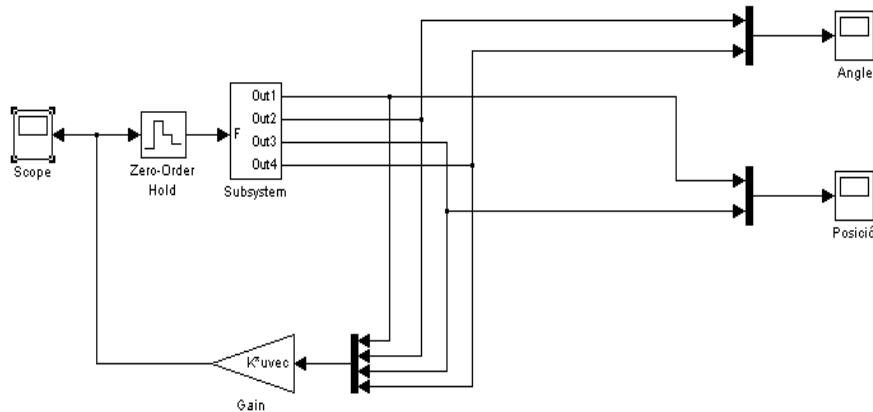


Fig. 3-12: proposta 3 diagrama de blocs sistema discretitzat

Finalment a la figura 3-12 tenim una altre proposta que consisteix en agafar un bloc anomenat subsystem i muntar dintre el diagrama de blocs de la figura 2-10 (sistema no lineal), el ordre de les sortides i entrades ve donat per aquests petits blocs que també es poden observar en la figura 2-10:

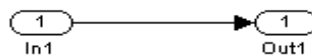


Fig. 3-13: blocs d'entrada i sortida

Si es vols que la sortida 1 sigui el angle s'ha de connectar a aquest, si la sortida dos a la derivada del angle s'ha de connectar a aquest segon i així successivament. L'ordre sortint al bloc subsystem serà el establert dins d'aquest projecte per cada variable (capítol 2): sortida 1  $x$ , sortida 2  $\theta$ , sortida 3  $\dot{x}$  i sortida 4  $\dot{\theta}$ .

Feta la simulació amb la primera proposta els resultats són:

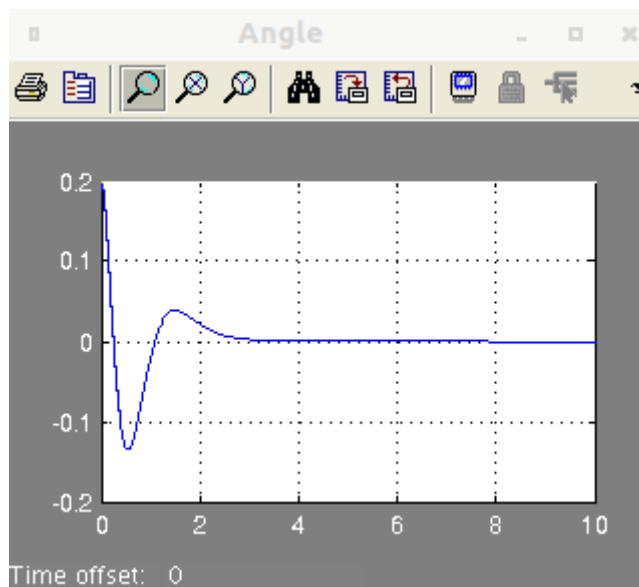


Fig. 3-14: resposta del angle

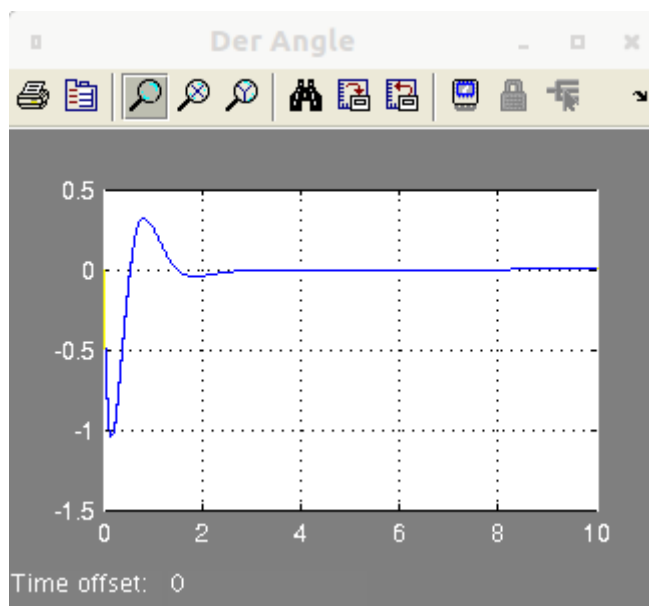


Fig. 3-15: resposta de la derivada del angle

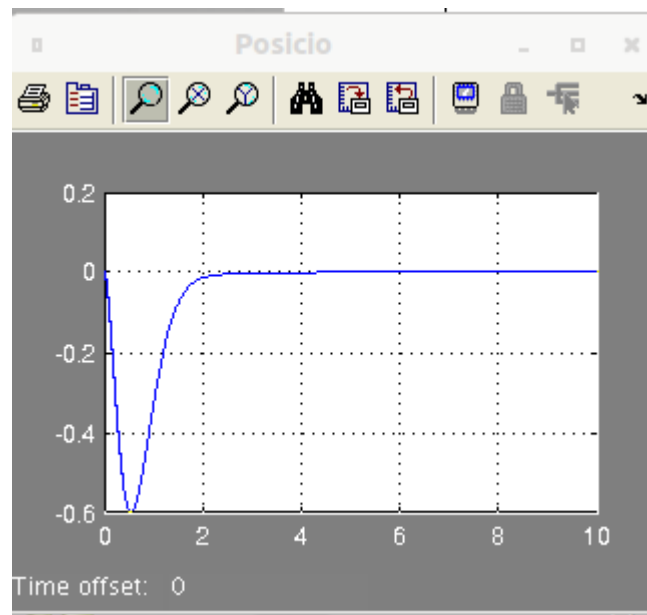


Fig. 3-16: resposta de la posició del carro

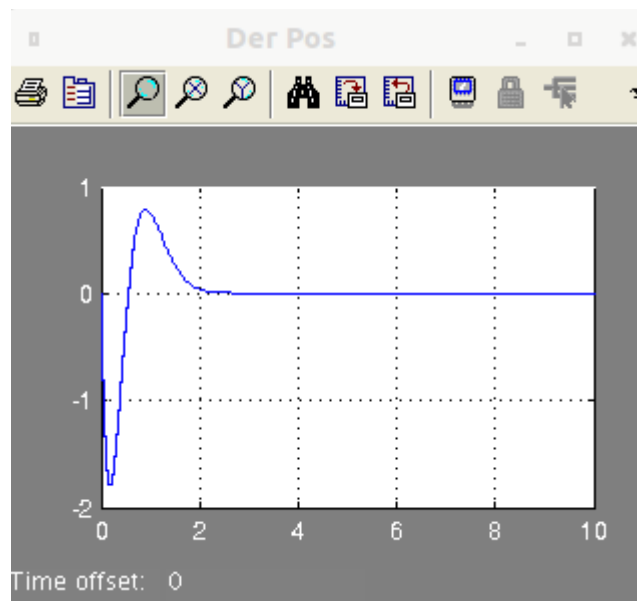


Fig. 3-17: resposta de la derivada de la posició del carro

Comparant els resultats de totes les figures de les senyals dels sistema continu amb el discret es pot veure que els resultats són gaire bé calcats.

Tant el angle com la posició i les seves respectives derivades tenen el mateix comportament, que és que cal esperar si tot està ben formulat.

Ara ja s'ha obtingut la matriu de realimentació que es farà servir al sistema real, es a dir a la maqueta de feedback.

La matriu que servirà doncs per realimentar el sistema és el vector  $Kd$  ja que correspon al sistema discretitzat i s'ha de recordar que Matlab treballa discretitzant el sistema i per tant s'utilitza sempre la matriu obtinguda del sistema discret.

### 3.3. Observadors o Estimadors de variables d'estat

El concepte d'observador o estimador surt del supòsit de que no es té un accés a totes les possibles variables del sistema. Així doncs la sortida del observador serà una estimació del vector d'estat.

Hi han dos tipus d'observador: el de ordre complet i el de ordre reduït. Aquí es simularà el d'ordre complet. Aquest té el mateix ordre que el propi sistema i és per això que fa una estimació de tota la matriu de realimentació del sistema.

La variables que se suposa coneixem són el l'angle i la posició i desconeixem les seves derivades.

Anomenarem  $\hat{x}$  a l'estimació de  $x$  i del les equacions 2.17 i 2.18 es suposa que coneixem A, B, C, D ; la sortida  $y$  i la entrada  $u$  són mesurables però no es sap el estat  $x$ . I  $\hat{y}$  serà l'estimació que s'obté de la sortida.

Com es coneix A, B, C i D podem duplicar el sistema:

$$\dot{\hat{x}} = A \cdot \hat{x} + B \cdot u$$

$$\hat{y} = C \cdot \hat{x} + D \cdot u$$

Ara la realimentació del sistema se li sumarà una matriu que s'anomena  $L(y - C \hat{x})$  que actuarà en cas de que hi hagi error i el farà tendir a zero; intruint el terme  $L$  i desenvolupant l'equació 2.17:

$$\dot{\hat{x}} = (A - LC) \cdot \hat{x} + B \cdot u + L \cdot y \quad (3.32)$$

Es defineix el error de càlcul com  $\tilde{x} = x - \hat{x}$ .

Per tant, la dinàmica del error en el sistema està donada per:

$$\dot{\hat{x}} = A \cdot \tilde{x} \quad (3.33)$$

Si es calcula  $\tilde{x}$  i es substitueix a 3.32 llavors  $\dot{\hat{x}} = (A - LC) \tilde{x}$ . Si s'aconsegueix que  $A - LC$  tinguin valors negatius el valor d'estimació tendria a zero.

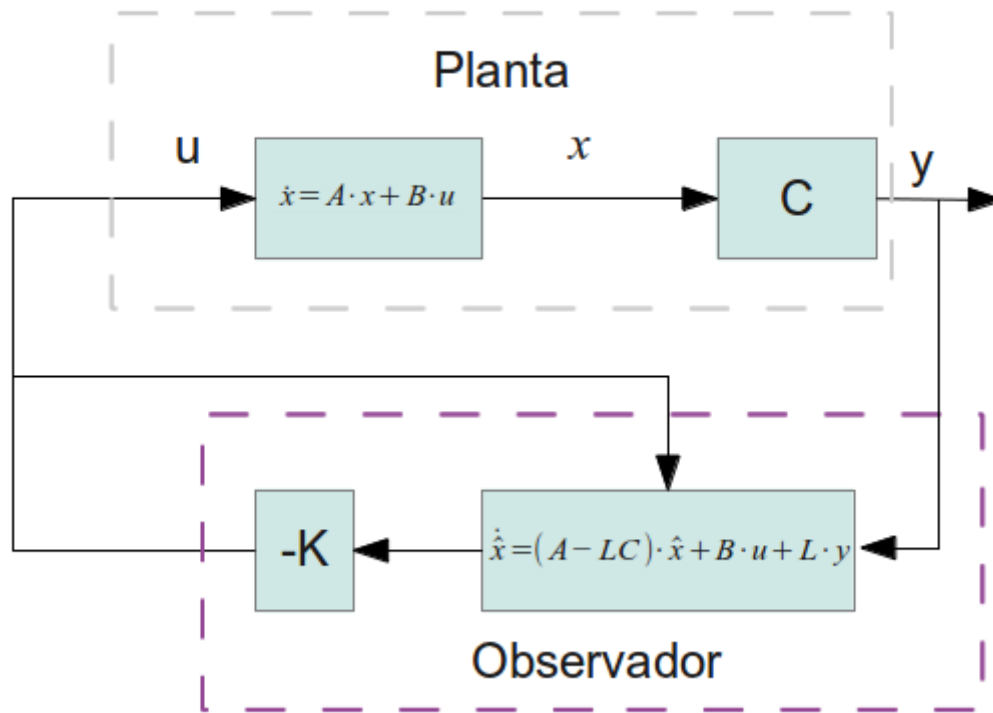


Fig. 3-18: diagrama de bolcs del control per retorn d'estats mes observador

### 3.3.1. Disseny de la matriu L

Per calcular la matriu  $L$ , igual que amb  $K$  i  $Kd$  del retorn d'estat, es recorre novament al Matlab. Els valor de les matrius  $A$ ,  $B$ ,  $C$ ,  $D$  ja s'han calculat abans i  $K$  és també la mateixa matriu del sistema continu per retorn d'estat i els pols ( $P_c$ ) són també els mateixos.

El pols d'un observador però han de ser entre 3 i 10 vegades més ràpids que els  $P_c$ , en aquest cas els multipliquem per 3,  $P_o = P_c * 3$ :

$$P_o = [-9.5433 + 9.1338i - 9.5433 - 9.1338i - 7.3338 + 1.3200i - 7.3338 - 1.3200i] \quad (3.34)$$

Una vegada es tenen els pols ja es pot obtenir la matriu L:

$$L = (\text{place}(A', C', P_o))' \quad (3.35)$$

Es important veure que tant  $A$ , com  $C$  i tota la  $L$  estan traspostes i a més també s'ha substituït la

instrucció *acker* per *place*, ja que la primera només serveis per una única entrada.

S'han de calcular les matrius que van dintre del bloc de realimentació:

$$acon = [A - B * K - L * C] \quad (3.36)$$

$$bcon = L \quad (3.37)$$

$$ccon = -K \quad (3.38)$$

$$dcon = [0 \ 0] \quad (3.40)$$

Ara ja es pot muntar el diagrama de blocs per fer la simulació:

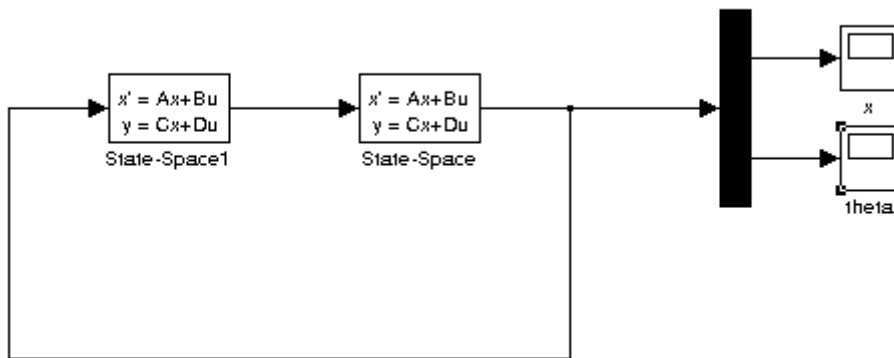
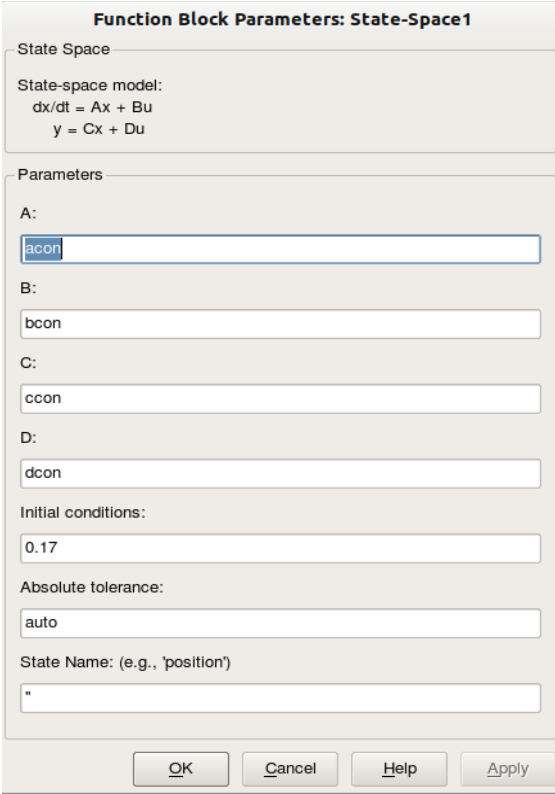


Fig. 3-19: diagrama de blocs del control per retorn d'estats mes observador de Matlab

El bloc State-Space és el de la configuració del sistema continu amb els valors A, B, C i D i el estat inicial és el mateix: totes les variables a zero excepte el angle a 0,2 radians.

El bloc State-Space1 és el observador.

La configuració del bloc State-Space1 és la següent:



**Function Block Parameters: State-Space1**

State Space

State-space model:  
 $dx/dt = Ax + Bu$   
 $y = Cx + Du$

Parameters

A:  
acon

B:  
bcon

C:  
ccon

D:  
dcon

Initial conditions:  
0.17

Absolute tolerance:  
auto

State Name: (e.g., 'position')  
"

OK Cancel Help Apply

Fig. 3-20: configuració del bloc State-Space1

Gràfics obtinguts de la simulació:

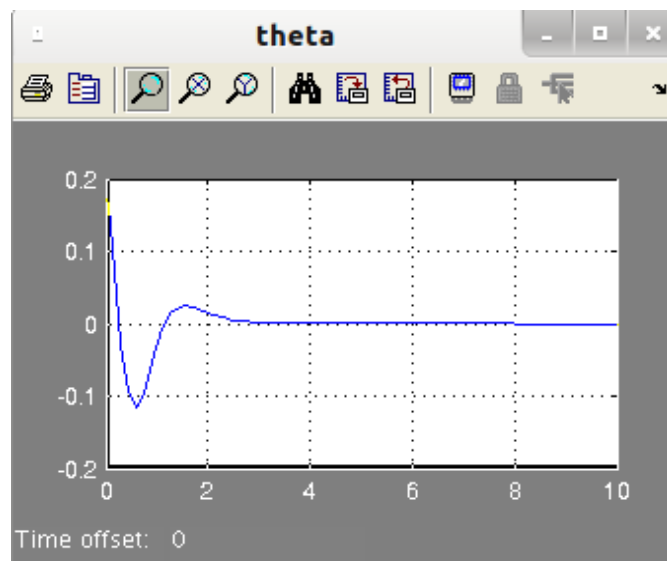


Fig. 3-21: resposta del angle



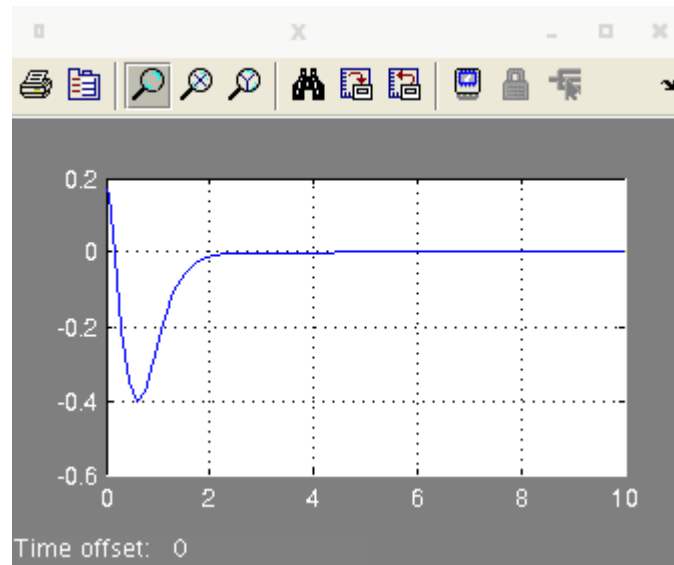


Fig. 3-22: resposta de la posició

La imatge de la figura 3-21 ens mostra l'evolució del angle quan parteix des de 0,2 radians i es pot observar que evoluciona cap al zero i s'estabilitza al seu voltant, que és el desitjat. Cal destacar el ràpid temps de resposta no solament d'aquesta gràfica si no la de la resta de simulacions.

La imatge de la figura 3-22 mostra la evolució de la posició i destaca que el seu moviment és una mica menor que les simulacions del retorn d'estat en les que el desplaçament d'aquesta variable quasi arribava fins als 0,6 metres. Tot es correcte.

### 3.3.2. Disseny de la matriu $L_d$

Per a dissenyar la matriu  $L_d$  el pols, al ser un sistema discret, els necessitem al pla  $\mathbb{Z}$ , però han de ser d'igual manera més ràpids. Com ja els tenim al pla  $\mathbb{Z}$  fer-los més ràpids ara significa dividir-los, en aquest cas triem per 10.

$$P_d = [0.975827253 + 0.0043i, 0.975827253 - 0.0043i, 0.968241 + 0.029i, 0.968241 - 0.029i] \quad (3.41)$$

Necessitem la  $K_d$  per construir les matrius del observador:

$$K_d = \text{acker}(A_d, B_d, P_d) \quad (3.42)$$

Fem els pols 10 vegades més ràpids:

$$P_o = \frac{P_d}{10} \quad (3.43)$$

$$P_o = [0.0976 + 0.0004i \quad 0.0976 - 0.0004i \quad 0.0968 + 0.0029i \quad 0.0968 - 0.0029i] \quad (3.44)$$

Ara ja es pot calcular la matriu  $L_d$  :

$$L_d = (\text{place}(A_d', C_d', P_o))' \quad (3.45)$$

El valor de  $L_d$  :

$$\begin{bmatrix} 1,8056 & 0,0024 \\ -0,027 & 1,8057 \\ -0,2536 & 81,5511 \\ 81,4978 & 0,2137 \end{bmatrix} \quad (3.46)$$

I per tant podem calcular els paràmetres del bloc observador discret:

$$a_{con} = [A_d - B_d * K_d - L_d * C_d] \quad (3.47)$$

$$b_{con} = L_d \quad (3.48)$$

$$c_{con} = -K_d \quad (3.49)$$

$$d_{con} = [0 \ 0] \quad (3.50)$$

Muntar el diagrama al Simulink de Matlab es permet simular:

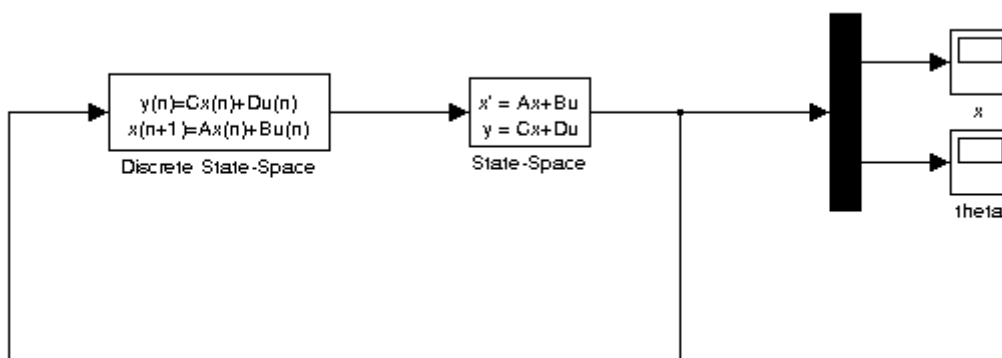


Fig. 3-23: diagrama de blocs del control per retorn d'estats mes observador discret

La configuració del bloc discret és igual que en el cas continu per al bloc Satae-Sapce1, però amb un temps de mostreig:

**Function Block Parameters: Discrete State-Space**

Discrete State Space

Discrete state-space model:  
 $x(n+1) = Ax(n) + Bu(n)$   
 $y(n) = Cx(n) + Du(n)$

Main State Attributes

A:  
acon

B:  
bcon

C:  
cccon

D:  
dcon

Initial conditions:  
0.2

Sample time (-1 for inherited):  
0.01

OK Cancel Help Apply

Fig.3-24: configuració del bloc Satae-Sapcel

Les gràfiques ens permeten analitzar els resultats:

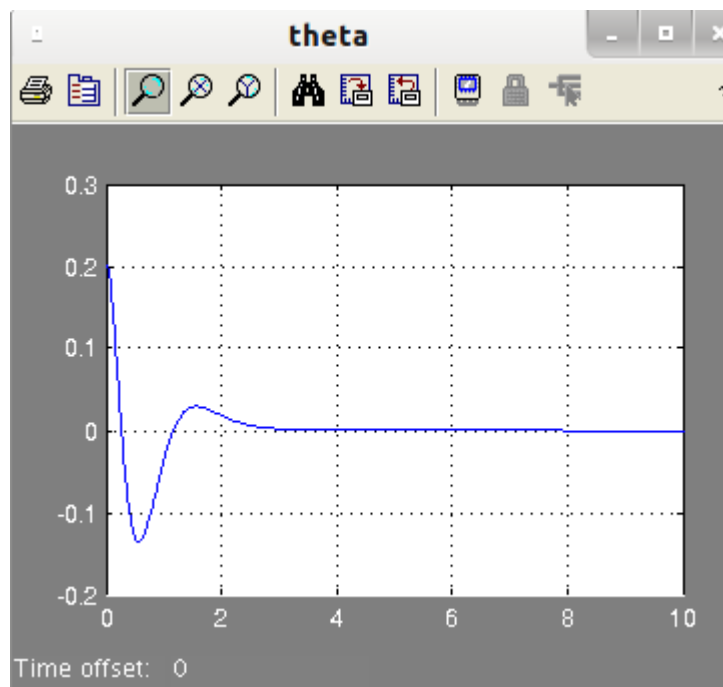


Fig. 3-25: resposta del angle

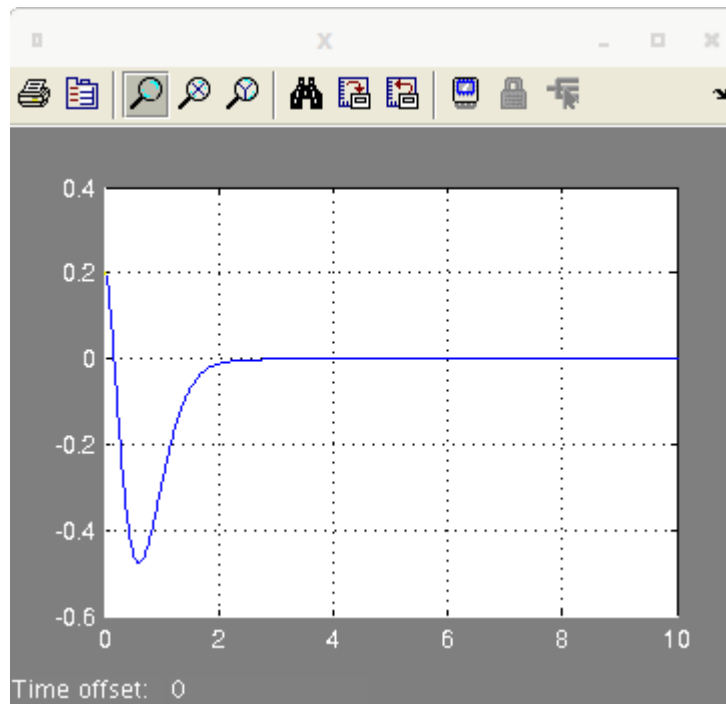


Fig. 3-26: resposta de la posició

Els resultats de les dues figures tant del angle com el del desplaçament del carro són molts semblants als simulats del sistema continu.

En ambdós casos en unes condicions inicials del angle a 0,2 radians i el desplaçament a zero mostra que el sistema tendeix a estabilitzar la situació del angle cap a zero i el temps d'estabilització continua sent molt bo.

Fins aquí podem dir que els càlculs estimats del observador són vàlids i per tant ja es podria provar la matriu de realimentació  $Ld$  en el sistema real per veure el seu comportament i saber si cal fer algun reajustament.

### 3.4. Control integral

L'acció integral serveix per eliminar el error estàtic ja que en cas de que hi hagi una entrada externa pot haver-hi aquest error.

A un sistema que se li afegeix un integrador augmenta el seu ordre, es a dir si un sistema és de segon ordre passaria a ser de tercer ordre. En el cas que ens ocupa després de afegir l'acció integral passarem a un sistema de cinquè ordre ja que el actual és de quatre.

Con que el que se li pot afegir al sistema és, per exemple, una entrada en esglaó, una senyal

triangular o quadrada, etcètera el que s'integrarà serà la posició.

Recordem que les equacions inicials són 2.17 i 2.18:

$$\dot{x} = A \cdot x + B \cdot u$$

$$y = Cx + Du$$

El nou estat és  $x_I$ ,

$$\dot{x}_I = C \cdot x = y \quad (4.51)$$

Amb una entrada externa  $r$ :

$$\dot{x}_I = C \cdot x - r = y - r \quad (4.52)$$

$$x_I = \int (y - r) \quad (4.53)$$

Per tant les noves equacions d'estat seran:

$$\begin{bmatrix} \dot{x}_I \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & C \\ 0 & A \end{bmatrix} \cdot \begin{bmatrix} x_I \\ x \end{bmatrix} + \begin{bmatrix} 0 \\ B \end{bmatrix} \cdot u + \begin{bmatrix} -1 \\ 0 \end{bmatrix} \cdot r \quad (4.54)$$

A on  $\begin{bmatrix} 0 & C \\ 0 & A \end{bmatrix}$  és la matriu  $A$  augmentada de  $n \times n$ ;  $\begin{bmatrix} 0 \\ B \end{bmatrix}$  és la matriu  $B$  augmentada de

$n \times 1$ ; i  $\begin{bmatrix} -1 \\ 0 \end{bmatrix} \cdot r$  és la matriu  $B$  en llaç tancat de  $n \times 1$ .

$$y = [0 \quad C] \cdot \begin{bmatrix} x_I \\ x \end{bmatrix} \quad (4.55)$$

A on  $[0 \quad C]$  és la  $C$  augmentada i la  $K$  augmentada serà:

$$K_a = [K_I | K] \quad (4.56)$$

I la sortida  $u = -K_a \cdot \begin{bmatrix} x_I \\ x \end{bmatrix}$ , la  $A$  augmentada en llaç tancat:

$$A_{a_{LLT}} = A_a - B_a \cdot K_a \quad (4.57)$$

El sistema discretitzat és molt semblant, només canvia la matriu A:

$$\begin{bmatrix} \dot{x}_i(K+1) \\ \dot{x}(K+1) \end{bmatrix} = \begin{bmatrix} 1 & Cd \\ 0 & Ad \end{bmatrix} \begin{bmatrix} x_i(K) \\ x(K) \end{bmatrix} \quad (4.58)$$

$$\dot{x}_i(K+1) = x_i(K) + e(K) \quad (4.59)$$

$$e(K) = (y(K) - R(K)) = Cd \cdot \dot{x}(K) \quad (4.60)$$

$$(Z-1) \cdot x_i = E \quad (4.61)$$

$$\frac{x_i}{E} = \frac{1}{(Z-1)} \quad (4.62)$$

Representat en diagrama de blocs:

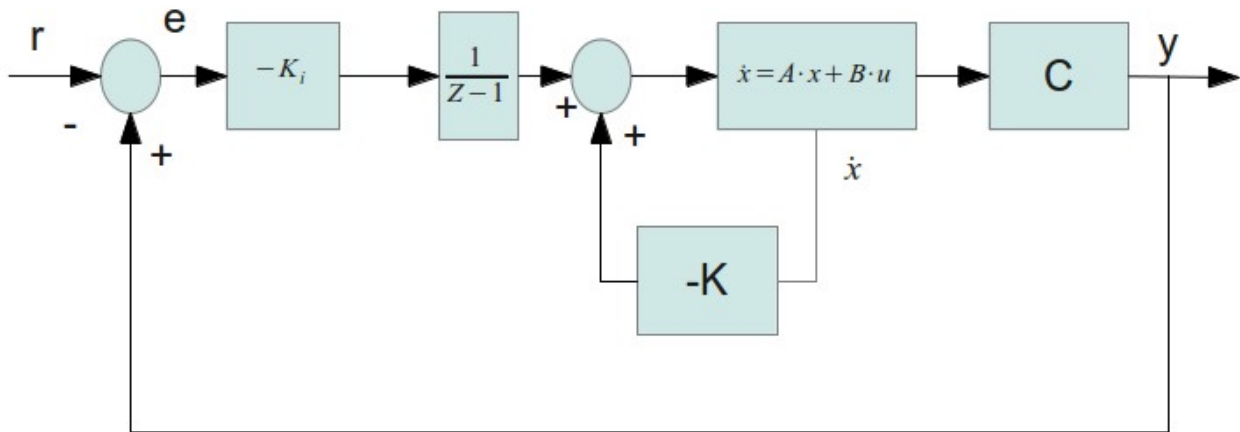


Fig. 3-27: diagrama de bolcs del sistema augmentat

### 3.4.1. Disseny de la nova matriu de realimentació Ka

Per al control integral es provaran els pols de Bessel de cinquè ordre multiplicats per 3:

$$P_c = [-2.7792, -1.7718 + 2.7216i, -1.7718 - 2.7216i, -2.5548 + 1.3281i, -2.5548 - 1.3281i] \quad (4.63)$$

Ara s'obtenen les matrius augmentades:

$$Ada = [0 \ 1 \ 0 \ 0; \text{zeros}(4,1), A] \quad (4.64)$$

$$Bda = [0; B] \quad (4.65)$$

$$Cda = [0 \ 1 \ 0 \ 0] \quad (4.66)$$

$$Dda = [0] \quad (4.67)$$

$$Ka = \text{acker}(Ada, Bda, Pc) \quad (4.68)$$

$$Ka = [-14.9836 \quad -19.7431 \quad 70.9041 \quad -13.8367 \quad 27.3047] \quad (4.69)$$

El digrama de blocs per al control integral del sistema continu és:

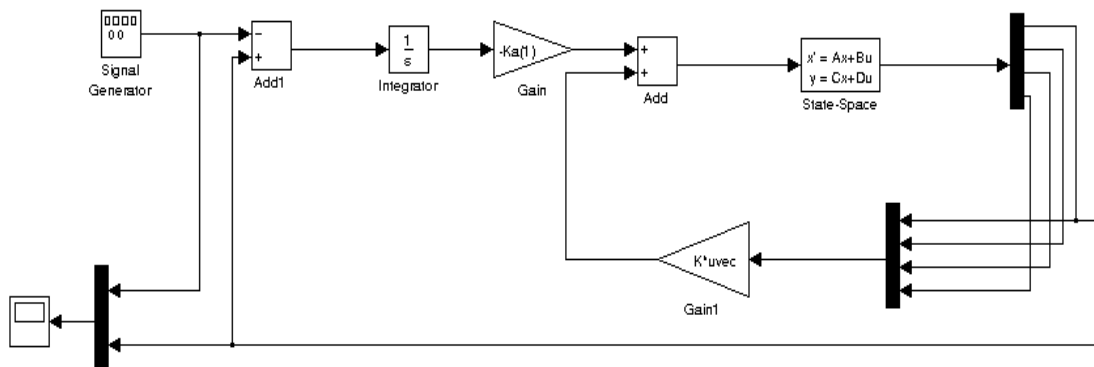


Fig. 3-28: diagrama de blocs del sistema augmentat de Matlab

Resultat de la simulació:

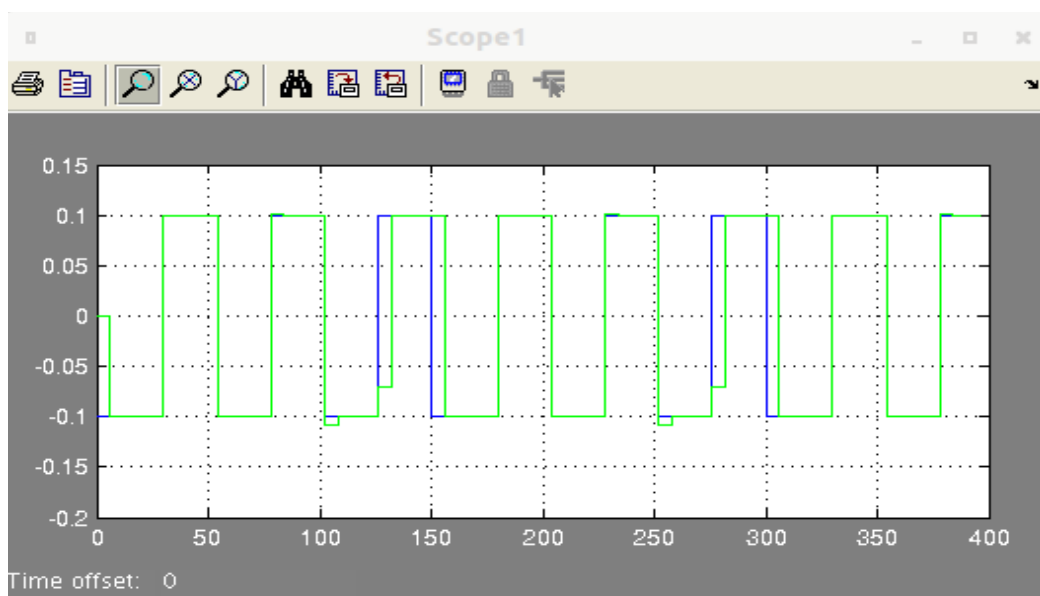


Fig.3-29: muntatge dels senyals  $x$  i la senyal d'entrada

A la imatge de la figura 3-29 hi han dues senyals:  $x$  i la  $x$  integrada, s'observa que el seguiment de la senyal és bo a excepció d'un petit desfasament al voltant del 150 i 300 segons, però sempre s'acaba per recuperar.

### 3.4.2. Disseny de la nova matriu de realimentació $K_{da}$

Es discretitza el sistema:

$$\text{sysd} = \text{c2d}(\text{sys}, 0, 01) \quad (4.70)$$

S'obtenen les matrius discretes:

$$[Ad \ Bd \ Cd \ Dd] = \text{ssdata}(\text{sysd}) \quad (4.71)$$

S'augmenta el sistema:

$$Ada = [1 \ 1 \ 0 \ 0; \text{zeros}(4,1), Ad] \quad (4.72)$$

$$Bda = [0; Bd] \quad (4.73)$$

$$Cda = [0 \ 1 \ 0 \ 0] \quad (4.74)$$

$$Dda = [0] \quad (4.75)$$

El pas a pla  $Z$  :

$$Z = e^{-\xi \omega T} \pm \omega \cdot T \cdot \sqrt{1 - \xi^2} = r < \pm \theta \quad (4.76)$$

Llavors  $r$ :

$$r = e^{-\xi \omega T} \quad (4.77)$$

I  $\theta$ :

$$\theta = \omega \cdot T \cdot \sqrt{1 - \xi^2} \quad (4.78)$$

Si s'aplica als pols de Bessel s'obté:

$$Pa = [0.9726, 0.9821 + 0.02673i, 0.9821 - 0.02673i, 0.97477 + 0.0129i, 0.97477 - 0.0129i] \quad (4.79)$$



Finalment ja tenim la  $Ka$  :

$$Kda = \text{acker}(Ada, Bda, Pa) \quad (4.80)$$

$$Kda = [-0,01404 \quad -18,6741 \quad 67,5610 \quad -13,0959 \quad 26,0149] \quad (4.81)$$

Ja es pot simular el sistema, es posarà una senyal quadrada d'entrada:

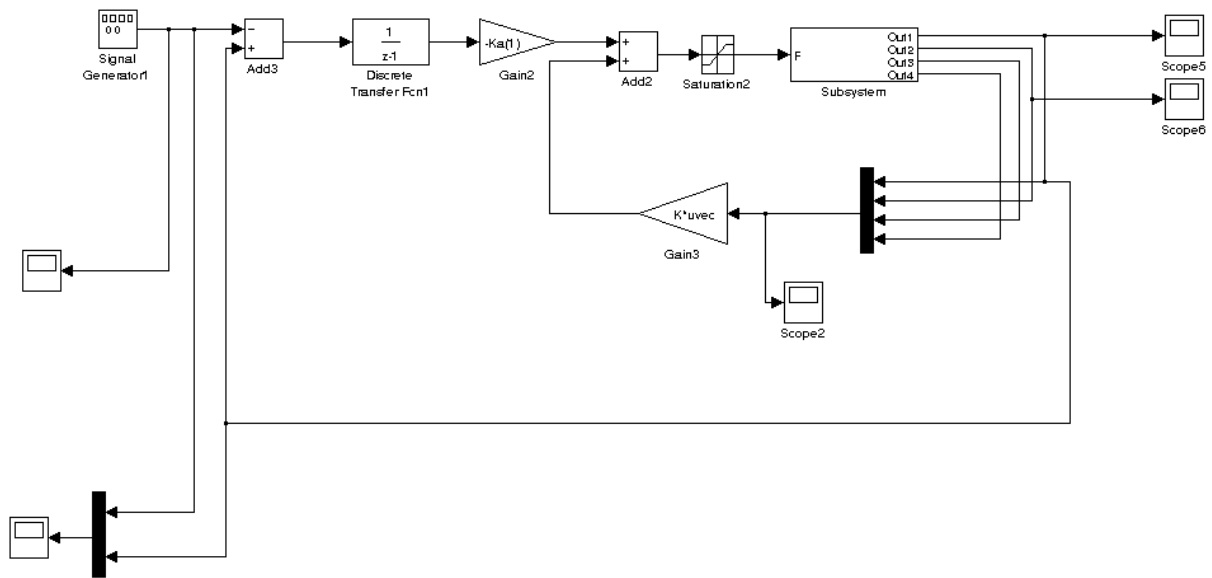


Fig. 3-30: diagrama de blocs del sistema augmentat no lineal discretitzat

Comentar que dins del bloc subsystem hi ha el diagrama de la figura 2-10.

Resultat:

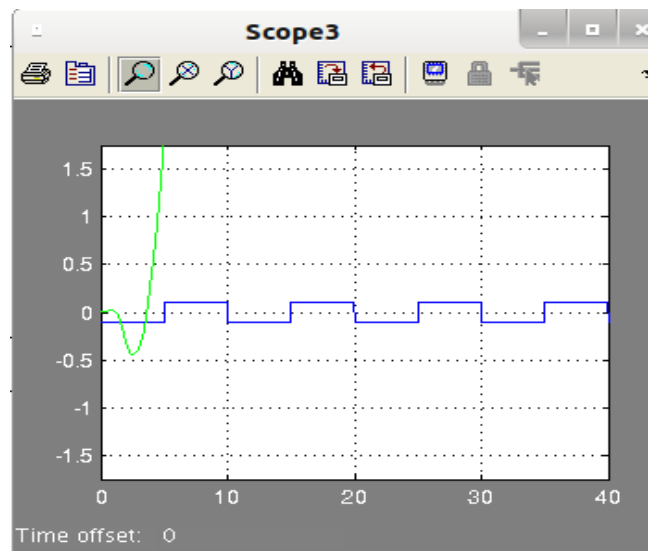


Fig. 3-31: muntatge dels senyals  $x$  i la senyal d'entrada

Aquesta resposta correspon al sistema no lineal discretitzat i el resultat no és satisfactori ja que la senyal de color verd hauria de seguir la senyal quadrada d'entrada, que és la blava.

La raó es perquè amb un integrador el sistema es torna massa crític i els fregaments  $b$  i  $d$  no terminen de comportar-se com deuriem.

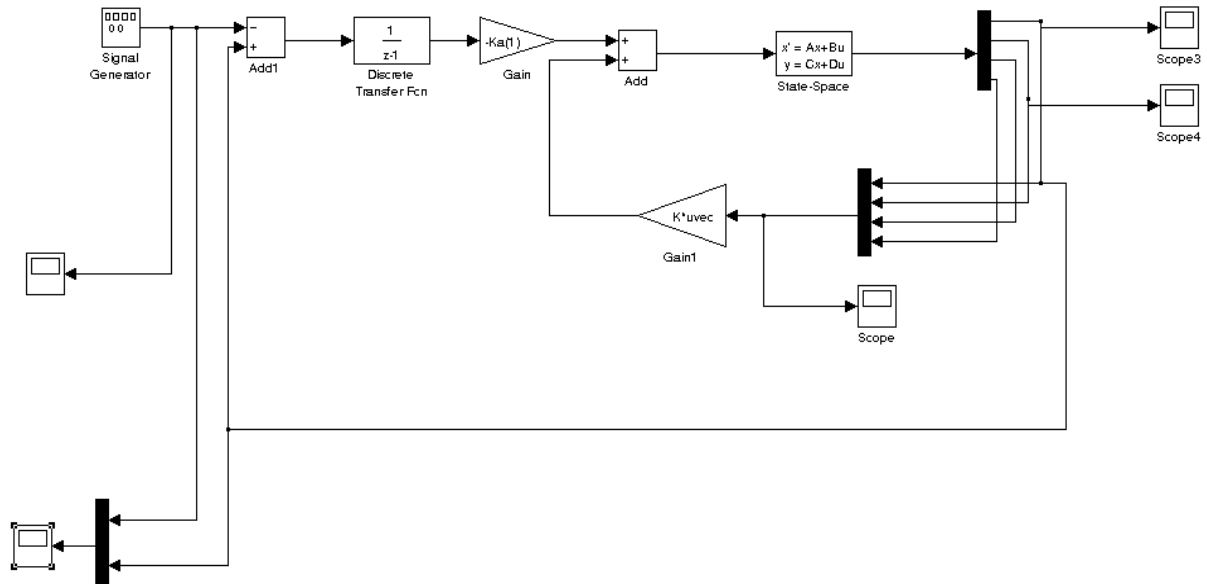


Fig. 3-32: diagrama de bolcs del sistema augmentat lineal discretitzat

Resultat:

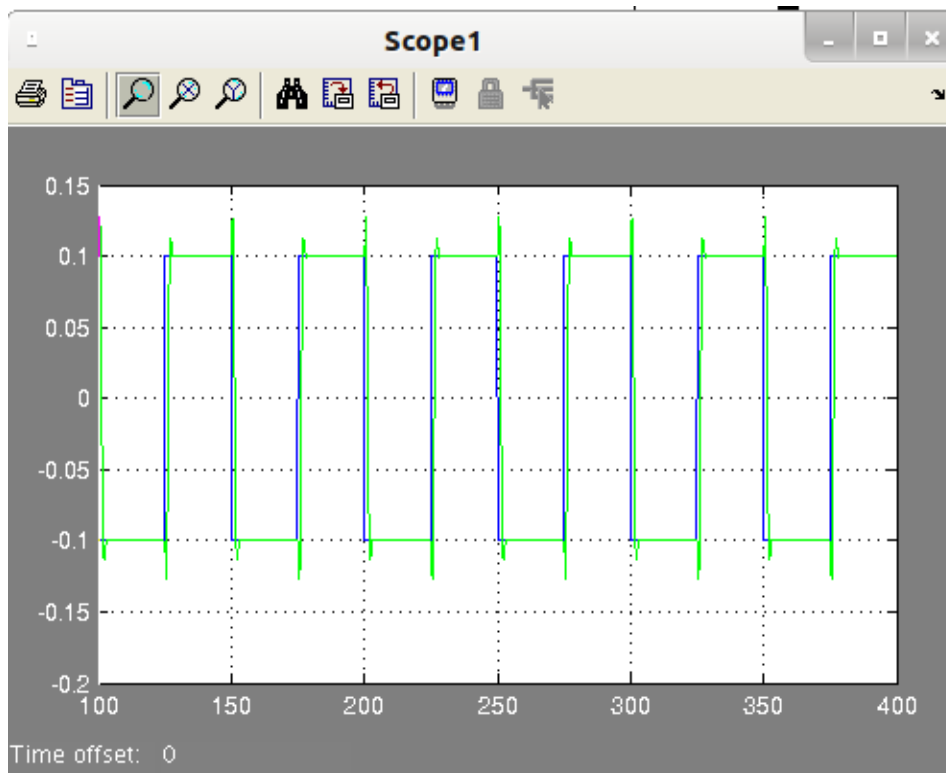


Fig. 3-33: muntatge dels senyals  $x$  i la senyal d'entrada

A la figura 3-32 es mostra el diagrama per al sistema discret. I la 3-33 la resposta.

Ara si, s'ha aconseguit una bona simulació, la senyal quadrada de color blau és l'entrada i la verd és la sortida de  $x$ . Com es pot comprovar las senyal pràcticament es superposen a excepció de quan canvia la polaritat que la sortida de  $x$  té una mena de pics però es pot considerar bona per fer la prova al sistema real.



## 4. Lògica Difusa

### 4.1. Introducció a la lògica difusa

El modelatge matemàtic com s'ha vist fins ara és una forma de poder controlar un sistema, però aquest mètode no és més que una de les possibles opcions per optar al control d'un sistema. No sempre la millor opció és la modelització o simplement no es pot plantejar com opció ja que no sempre es pot linealitzar, llavors podem triar altre alternativa com és la lògica difusa.

La lògica difusa neix a l'any 1965 quan Lofti A. Zadeh va plantejar la teoria de la lògica difusa basada en un sistema de regles. La novetat de la idea consistia en que l'elaboració d'aquestes regles es podia fer d'una manera que s'assemblés més a la lògica de raonament humà que a la lògica tradicional. La lògica difusa difereix de la lògica tradicional en que no tot és verdader o fals, si no que ara una variable pot tenir un grau de veracitat i a la mateixa vegada dependrà del grau de veracitat de les seves premisses. Així doncs tota regla que tingui un grau de veracitat tindran alguna cosa a dir.

### 4.2. Lògica difusa

Al igual que qualsevol altre sistema de control la lògica difusa treballa sobre un estat desitjat, estableix el valor requerit del sistema per mitja d'una variable mesurada i dur a terme la seva tasca sobre una variable controlada.

La lògica difusa es diferencia en que en el conjunt del procés no s'utilitzen equacions ni números, si no regles: donat un estat i els valors de les seves variables d'entrada; s'avalua la veracitat de cada regla i pren una decisió sobre els canvis a realitzar sobre cada variable de sortida; que produirà un canvi en el estat del sistema per tornar a obtenir el estat de les variables d'entrada i tornar a començar.

Aquest procés és més proper al raonament humà ja que permet realitzar operacions amb paraules:

Si la **temperatura** és alta i la **pressió** te tendència a baixar. Aleshores tanca la **vàlvula**  
una mica

A on són variables lingüístiques: temperatura, pressió i vàlvula. I que poden prendre una valors que són etiquetes:

$$\underbrace{\text{Alta Tendència a baixar tanca una mica}}_{\text{conjunts difusos difusos}}$$

Com es pot apreciar els conjunts difusos poden acceptar més de dos possibles valors (no estrictament zero o un) per a la pertinença a un element i estableixen valors de sortida en intervals:

$$v_p : X \rightarrow [0,1]$$

Mentre que la lògica clàssica es pot expressar com una funció que relaciona un element amb valors de sortida 0 i 1:

$$F : X \rightarrow \{0,1\}$$

La pertinença a un conjunt ve donada per una funció coneguda com *funció de pertinença* i que estableix el grau de pertinença d'un element amb el conjunt que el conté.

Tot aquest cicle pot ser dividit en tres parts amb el següent ordre:

1. Fuzzification
2. Avaluació de regles o inferència
3. Defuzzification

### 4.3. Fuzzyfication

Aquesta etapa del procés realitza la traducció de la variable mesurada al llenguatge del controlador. I es procedeix a adquirir els valors del sistema per calcular el grau de pertinença que aquests valors tenen amb als possibles conjunts fuzzy (fussy inputs) als que han estat assignades.

Si parlem d'un cas concret per exemple temperatura, es quantifica el seu grau de pertinença als conjunts representats amb les etiquetes: *molt\_baixa, baixa, temperatura, alta, molt\_alta*.

Per definir aquestes variables lingüístiques es creen les funcions de pertinença:

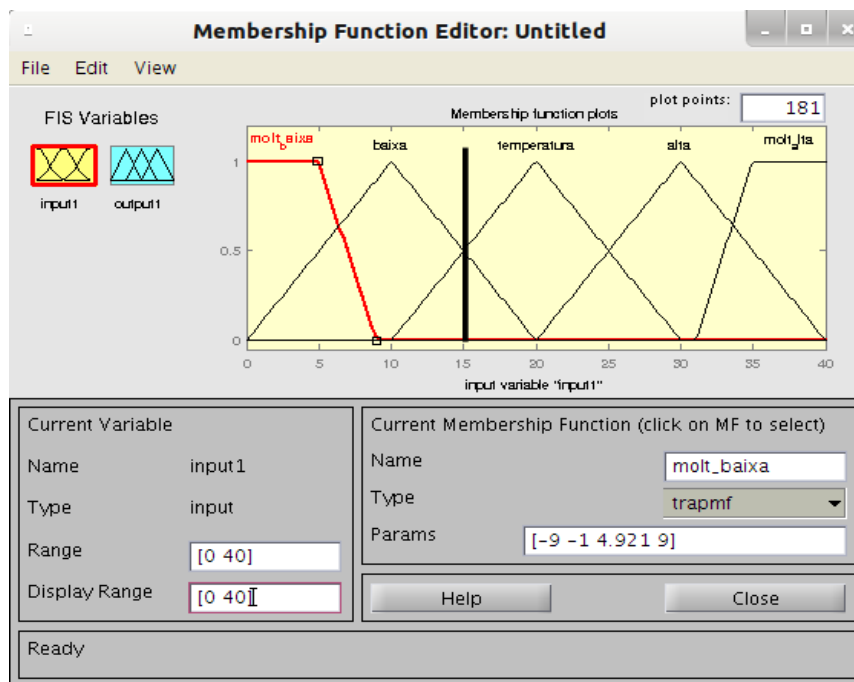


Fig. 4-1: funcions de pertinença

A la figura 4-1 es veu que per a una temperatura de 15° la funció de pertinença per l'entrada input és parcialment *baixa* i parcialment *temperatura*.

Per crear les funcions de pertinença s'han de fer sota uns conceptes que s'han d'aplicar:

- El univers del discurs: rang sobre el qual operen els conjunts difusos, ha de tenir tots els valors possibles de la variable que es mesura.
- Conjunts difusos: que poden tenir moltes formes, tals com triangles, trapezis, quadrats... per escollir la més òptima s'ha de establir aquella que representi amb major fiabilitat el comportament del sistema. La posició i forma dels conjunts han de tenir la forma més directa amb el coneixement adquirit del procés de manera que la informació que expressi cada conjunt representi el comportament del sistema.

#### 4.4. Avaluació de regles

És l'etapa del controlador a on s'avalua el pes de cada regla obtenint el grau de veracitat per cada una d'aquestes.

Per l'elaboració de les regles cal tenir en compte cada situació possible. El resultat de la inferència difusa, es a dir la capacitat de deduir una cosa a partir d'un altre, serà una variable lingüística que tindrà una forma donada per la relació que tingui aquesta amb el conjunt associat a la seva resposta.

Per la creació d'aquesta etapa cal tenir en compte:

- La definició de cada variable lingüística de sortida (variable controlada) determinant les característiques dels seus conjunts de la mateixa forma que es va fer amb els conjunts d'entrada.
- Crear una base de coneixements que determini les accions adequades del controlador en forma de regles.
- Crear un conjunt de regles que manifestaran el comportament del sistema. Aquestes regles, si analitzem la regla de l'exemple d'abans, tindran la següent forma:

*Si la temperatura és alta i la pressió te tendència a baixar. **Aleshores** tanca la vàlvula una mica*

## 4.6. Defuzzyfication

S'anomena defuzzification al procés que transforma un conjunt o valors difusos en un conjunt o valors no difusos, es a dir en termes numèrics. És el pas invers a la defuzzification.

Per generar una defuzzification es poden utilitzar diferents mètodes com:

- Per màxims de centres
- Mitjana de centres
- Per centre d'àrea

## 4.7. Lògica difusa sobre el sistema

Per dur a terme la lògica difusa per aixecar el pèndul s'han proposat varies formes. La més interessant era tenir en compte tres variables d'entrada: la posició del carro, la del angle i la de la derivada de l'angle. Aquesta semblava la millor opció però la maqueta de la que disposem a l'EUPMT té un problema que feia impossible l'aplicació d'aquesta idea.

El problema de la maqueta és que a mesura que passa el temps i el driver està engegat el motor cada vegada exerceix menys força sobre el carro. Això dificultava la configuració de les regles i els seus camps d'actuació ja que variaven a mesura que passava el temps.

Així es va agafar una idea d'un article que només tenia en compte la posició de carro i del angle [8]. Les regles que proposava l'article es basaven en la idea d'aixecar el pèndol amb força, en qüestió de



tres o quatre empentes. Amb el problema de la maqueta és la millor opció ja que d'altre manera era difícil treballar.

Per poder editar tota la informació i introduir-la al Matlab es fa amb un editor que incorpora el entorn de treball que apareix si a la línia de comandaments es tecleja fuzzy.

#### 4.7.1. Fuzzyfication de les variables d'entrada

Com a variables d'entrada es defineixen  $x$ , que serà la posició del carro i  $\theta$  que serà la posició del angle.

El univers del discurs per la posició estarà comprès entre el interval  $[-0.3 \ 0.3]$  expressat en metres. Cal no excedir-se ja que una distància massa llarga farà que el carro arribi fins al final de carrera i es talli la simulació.

Current Variable	
Name	$x$
Type	input
Range	<input type="text" value="[-0.3 0.3]"/>
Display Range	<input type="text" value="[-0.3 0.3]"/>

Fig. 4-2: univers del discurs de la posició

Per la posició del angle es defineix un univers del discurs que oscil·larà entre el interval  $[0 \ 6.284]$  expressat en radians, ja que els sensors treballen amb aquesta magnitud.

Current Variable	
Name	$\theta$
Type	input
Range	<input type="text" value="[0 6.284]"/>
Display Range	<input type="text" value="[0 6.284]"/>

Fig. 4-3: univers del discurs de l'angle

#### 4.7.2. Inferència, defuzzification i defuzzification sobre el sistema

La tria de les regles han estat les del article [8]. Per l'elaboració de les regles em de delimitar les zones de cada univers del discurs. En el cas de l'entrada  $\theta$  (angle) per situar les àrees que recorre la variable farem servir un eix dividit en quadrants:

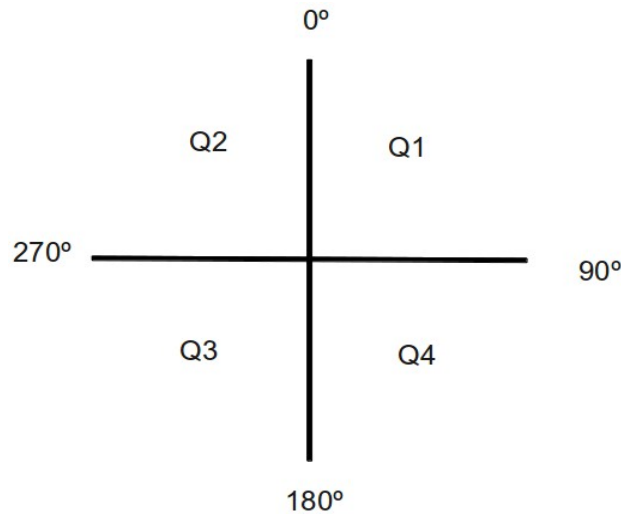


Fig. 4-4: divisió del espai recorregut pel pèndol en quadrants

Les etiquetes lingüístiques per la variable  $\theta$  seran Q1, Q2, Q3 i Q4. Als punts intermedis entre quadrants també se'ls assignaran etiquetes però al no estar clarament definits no s'utilitzaran; aquests punts són: Q1-Q4, Q4-Q3, Q3-Q2 i Q2-Q1.

Per la variable d'entrada  $x$  les etiquetes lingüístiques seran ZO (zero) NB (negative big) i PB (positive big).

La variable de sortida és la força que s'aplica al sistema en funció de l'estat en que aquest es trobi. Les etiquetes lingüístiques per aquesta variable que anomenem  $F$  seran: ZO, NB, NM, PM, PB

a on tant per  $F$  com per  $x$  :

- ZO és zero
- NB és negative big
- NM és negative medium
- PB és positive big
- PM és positive big

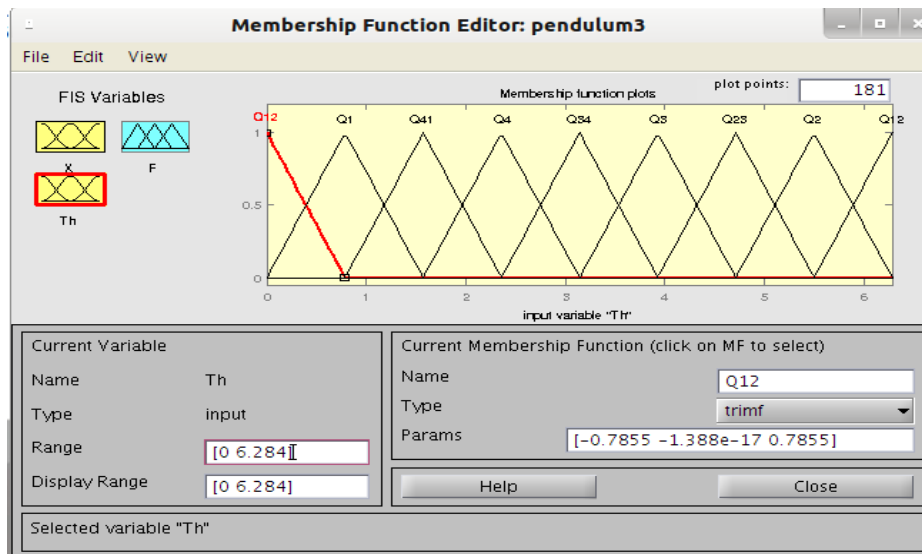


Fig. 4-5: definició d'etiquetes per la variable theta amb l'editor de funcions de pertinença de Matlab

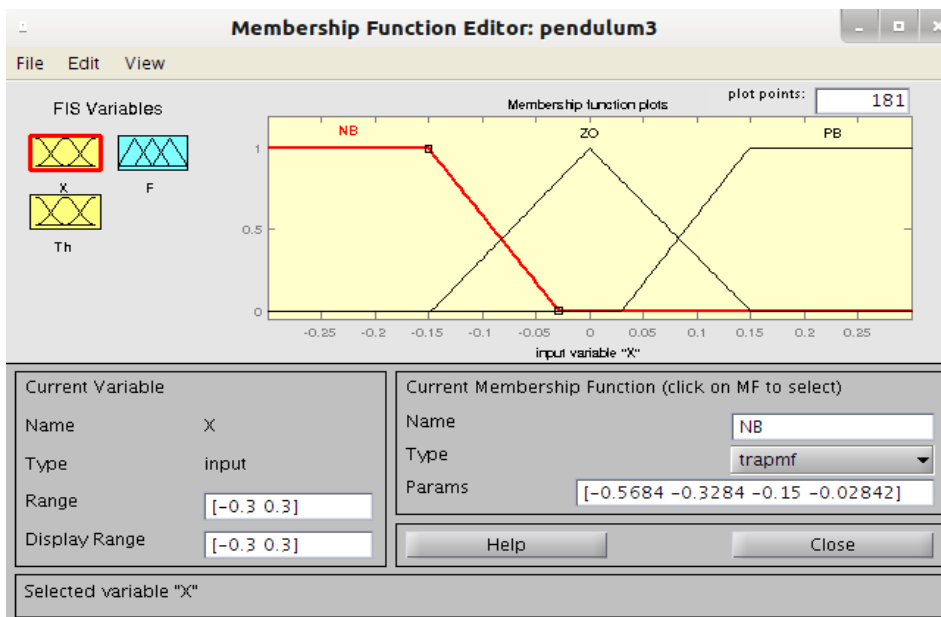


Fig. 4-6: definició d'etiquetes per la variable x amb l'editor de funcions de pertinença de Matlab

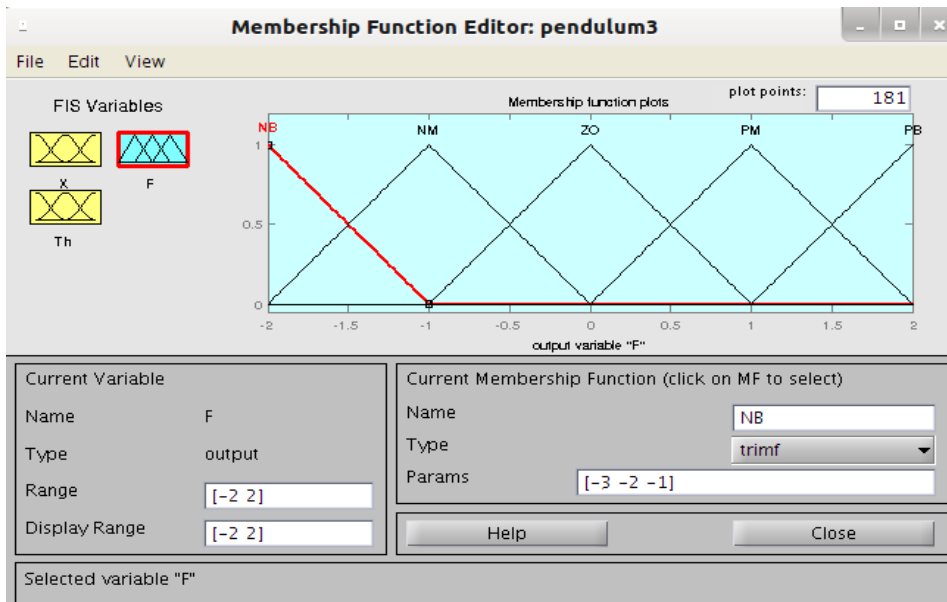


Fig. 4-7: definició d'etiquetes per la variable  $F$  amb l'editor de funcions de pertinença de Matlab

A la figura 4-5, 4-6, 4-7 s'observa la definició de regles, la forma de les funcions de pertinença i el univers de discurs de cada variable:  $x$ ,  $\theta$  i  $F$  respectivament.

Ara cal construir una taula que relacioni les etiquetes lingüístiques de les dues variables d'entrada amb la variable de sortida. A la primera columna es posa les etiquetes de  $x$  i a la primera fila les de  $\theta$ . La relació entre ambdues són les etiquetes lingüístiques de  $F$ .

	Q4	Q1	Q2	Q3
NB	ZO	PB	ZO	PB
ZO	NM	PB	NB	PM
PB	NB	ZO	NB	ZO

Taula 4-1: definició de regles

```

1. if (x is NB) and (Th is Q1) then (F is PB) (1)
2. if (x is NB) and (Th is Q4) then (F is ZO) (1)
3. if (x is NB) and (Th is Q3) then (F is PB) (1)
4. if (x is NB) and (Th is Q2) then (F is ZO) (1)
5. if (x is ZO) and (Th is Q2) then (F is PM) (1)
6. if (x is ZO) and (Th is Q3) then (F is NB) (1)
7. if (x is ZO) and (Th is Q4) then (F is PB) (1)
8. if (x is ZO) and (Th is Q1) then (F is NM) (1)
9. if (x is PB) and (Th is Q1) then (F is ZO) (1)
10. if (x is PB) and (Th is Q4) then (F is NB) (1)
11. if (x is PB) and (Th is Q3) then (F is ZO) (1)
12. if (x is PB) and (Th is Q2) then (F is NB) (1)
    
```

Fig. 4-8: Regles editades amb l'editor de regles de Matlab

Finalment s'introdueix aquesta taula al sistema amb el editor de regles de Matlab, figura 4-6.

### 4.7.3. Simulació del sistema

#### 4.7.3.1. Simulació sobre sistema no lineal

La simulació del sistema es farà sobre el sistema no lineal, el muntatge del diagrama de blocs al Matlab és el següent:

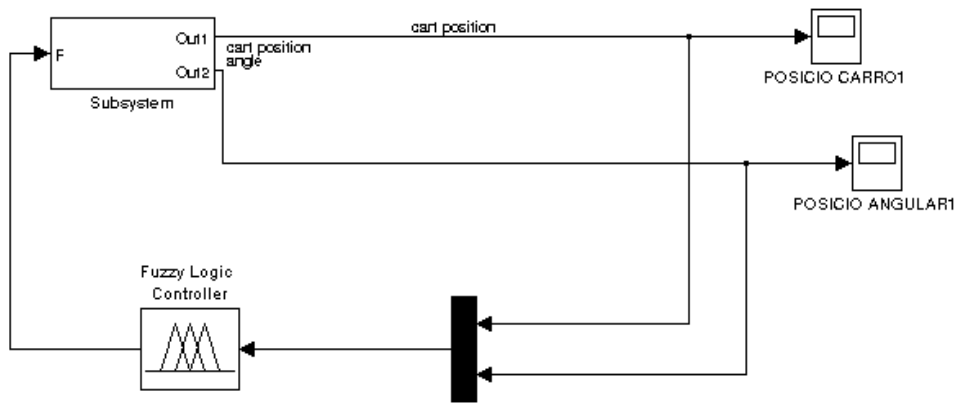


Fig. 4-9: diagrama de blocs muntatge simulació lògica difusa

Resultat de la simulació:

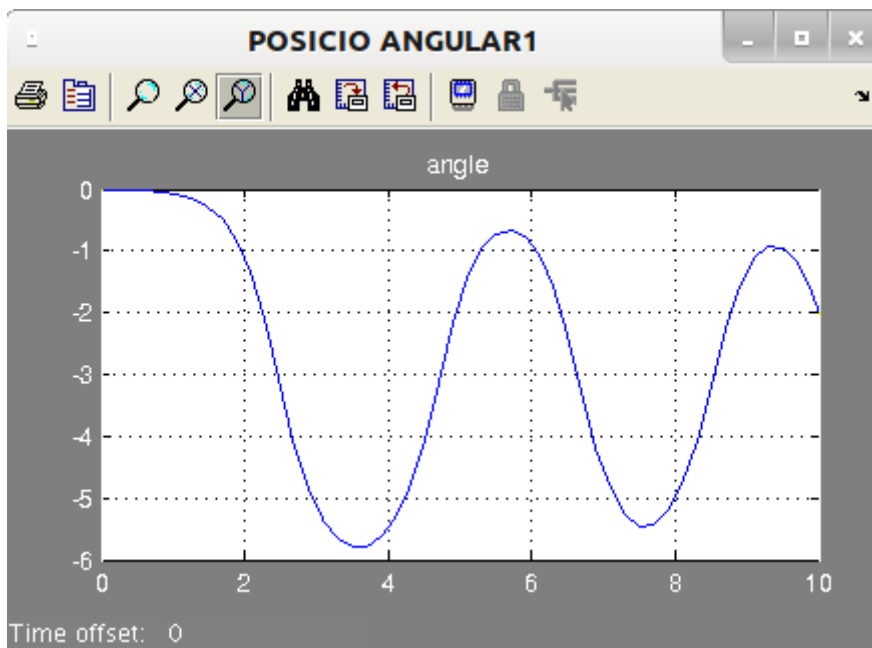


Fig. 4-10: resposta del la variable theta (angle)

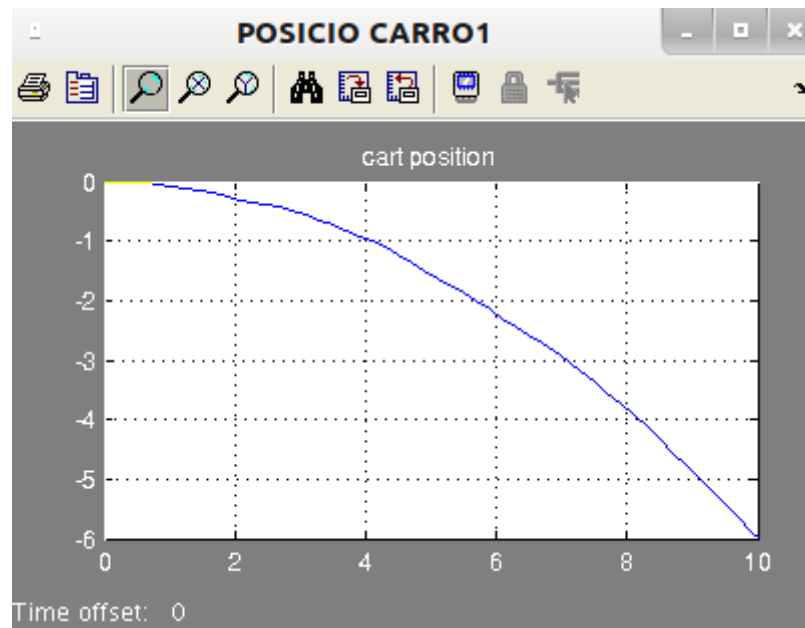


Fig. 4-11: resposta del la variable  $x$  (posició)

A les figures 4-10 i 4-11 es veuen les gràfiques de les respostes de les dos variables,  $\theta$  i  $x$ . La senyal del angle té un comportament desitjat ja que l'angle oscil·la i arriba ràpidament als 6,28 radians.

La posició surt cap a un costat i se suposa que hauria de oscil·lar, però com s'ha dit abans el comportament de la simulació en aquest sentit pot no ser del tot fiable. A més s'ha d'observar que la variable  $\theta$  arriba ràpidament al punt d'equilibri i per tant pot donar temps a fer el control de posició sense que el carro arribi al final de carrera de la maqueta.

#### 4.7.3.2. Simulació sobre sistema discret

Per a simular el sistema discretitzat s'ha triat el bloc que Feedback té per fer les seves simulacions ja que el discretitzat utilitzat fins ara per aquesta simulació tenia un comportament no fiable:

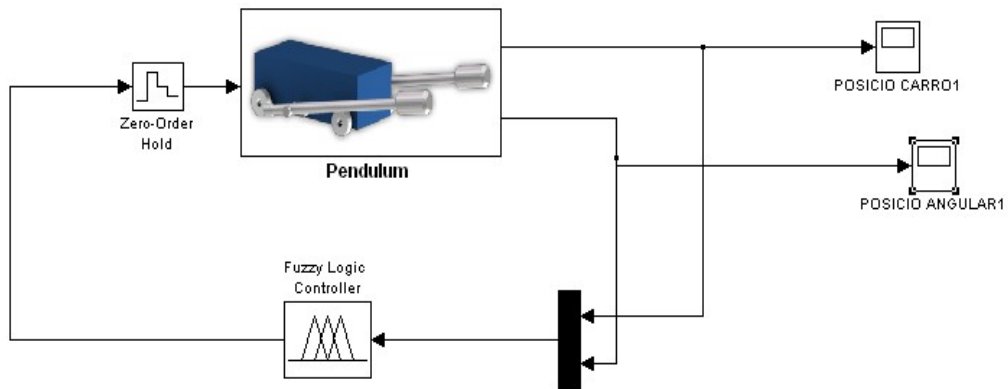


Fig. 4-12: Diagrama de blocs simulació sistema discretitzat amb el bloc de Feedback per al pèndol

Resposta de la simulació:

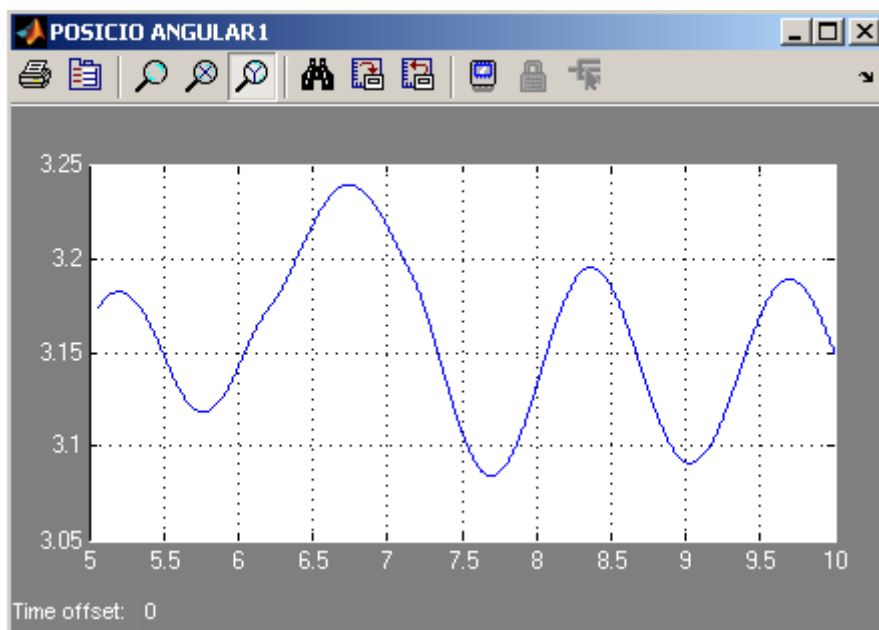


Fig. 4-13: resposta del la variable theta (angle)

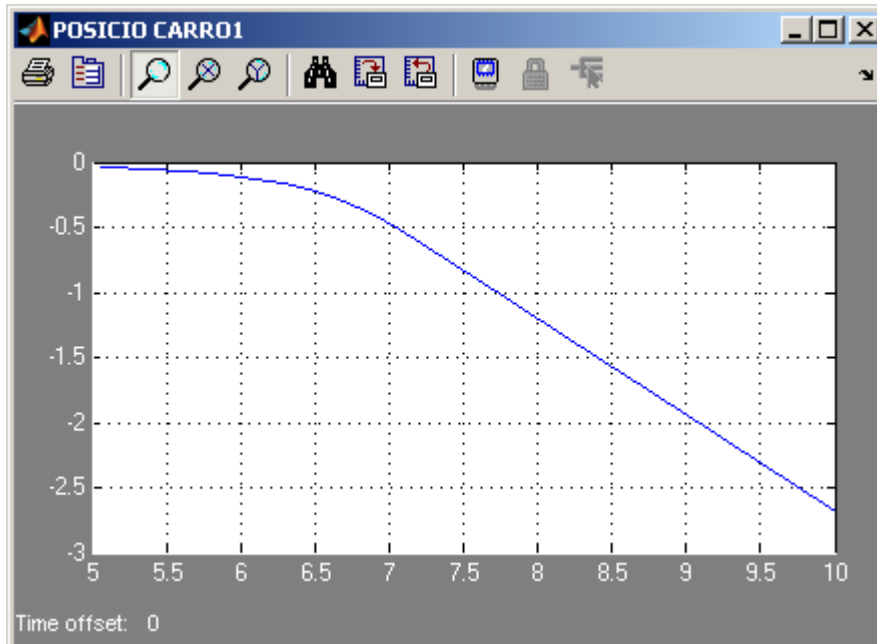


Fig. 4-14: resposta del la variable  $x$  (posició)

A les figures 4-13 i 4-14 es veuen les gràfiques de les respostes de les dos variables,  $\theta$  i  $x$  al sistema discretitzat. La senyal del angle en aquesta simulació no arriba als 6,28 radians, de fet es queda molt curt.

La posició sí que es comporta similarment, surt cap a un costat.



## 5. Assajos dels controladors sobre el sistema real

En aquesta part del projecte s'aplicaran tots els càlculs i simulacions fets fins ara al sistema real. Però primer hem de conèixer algun punts del sistema per poder-ho fer amb efectivitat.

Als manuals del pèndol invertit de Feedback hi han certes coses de les que no es comenta res i cal esbrinar aquestes qüestions. Com per exemple la relació entre la tensió amb la força. Els manuals ens diuen que per una entrada de  $\pm 2,5V$  tenim una força aplicada de  $\pm 20N$  com que no es parla d'una constant fa suposar que aquesta sigui 1, però calia comprovar-ho.

Per altre banda tenim que la linealització s'ha fet entorn a zero graus però la lectura del sensors sobre el pèndol al girar en sentit horari quan arriba als 0 graus es de  $360^\circ$  (6,28 radians).

També s'ha desenvoluparà en aquest capítol el disseny del switch, per canviar del control que serveix per aixecar el pèndol a la resta de controls dissenyats: retorns d'estat, observador, control integral. Aquesta part es va desenvolupar directament sobre la maqueta, sense simular.

Per aixecar el pèndol es faran servir dos mètodes un serà el fuzzy del capítol 4 i l'altre serà un dels mètodes que té Feedback d'exemple. La raó és per evitar la comentada pèrdua de força progressiva que té la maqueta ja que sí que és veritat que als exercicis de Feedback també ocorre aquest problema també s'ha comprovat que un d'ells està dissenyat de tal forma que evita el problema.

Per aquestes raons el primers apartats d'aquest capítol es tractaran aquestes qüestions al camp experimental i a continuació es plasmaran els resultats de cada control, estudiat i simulat que apareixen en aquest treball, al sistema real.

### 5.1. Relació entre tensió i força

Per fer les simulacions el més reals possibles ens cal saber què relació hi ha entre la força i la tensió. Per esbrinar si la relació era 1 o algun altre valor es va deixar el sistema real en llaç obert ficant una tensió a l'entrada juntament amb un dinamòmetre. Aquest muntatge permet introduir una tensió i mesurar la força aplicada al carro.

El primer que es va fer és desmuntar el pèndol del carro i es va fixar el dinamòmetre entre el carro i uns forats que hi han a prop dels finals de carrera. Després es va muntar el següent diagrama de blocs al Matlab:

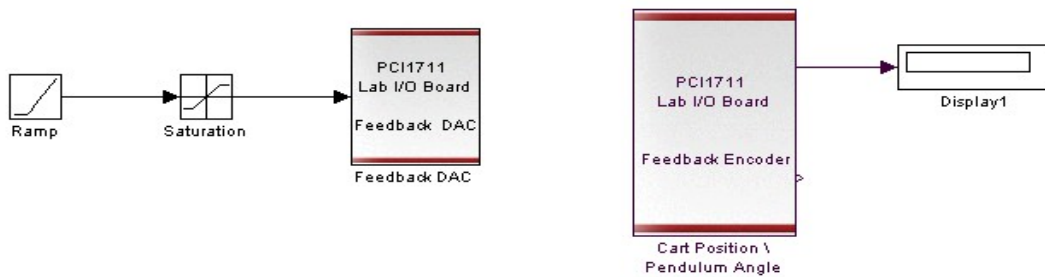


Fig. 5-1: diagrama de blocs de Matlab en enllaç obert

El bloc *Ramp* ens permet anar ficant diferents tensions que augmenten progressivament i amb el bloc *Saturation* limiten els valors màxims i mínims.

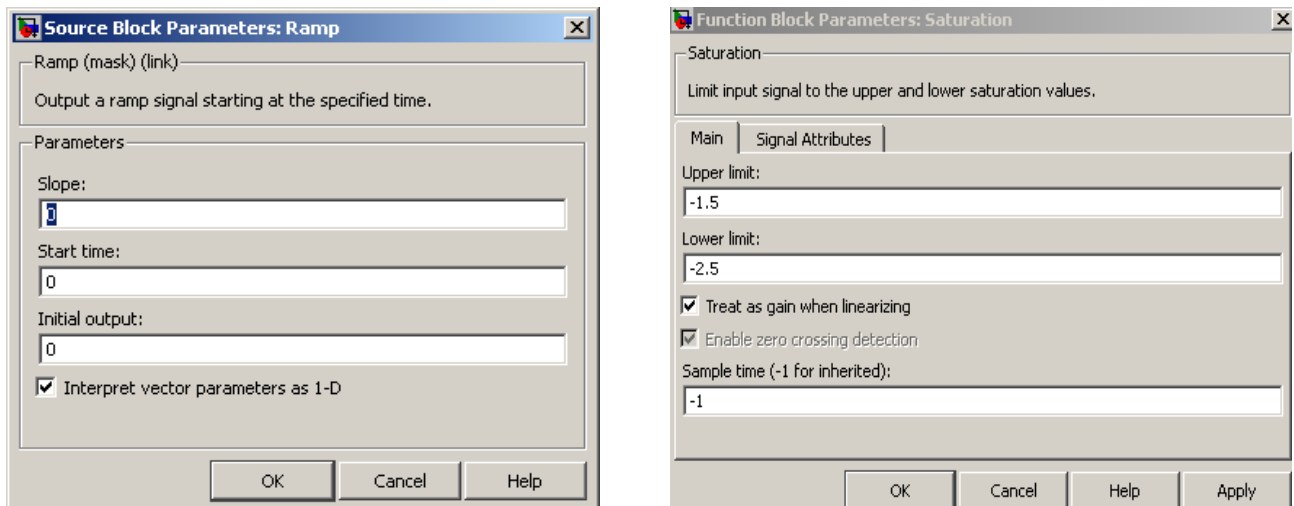


Fig. 5-2: a l'esquerra configuració bloc Ram; a la dreta configuració bloc Saturation

Com s'ha comentat la maqueta a mesura que passa el temps dona diferents valors de sortida (força) per una mateixa tensió, cada vegada menys intensos, per això es va construir una taula al LibreOffice Calc en la que es van introduir diferents valors per a cada tensió i es va calcular la mitjana.

1	Tensió	Newton	Newton	Newton	Mitjanes
2	0,3	2,3	1,6	1,4	1,766666667
3	0,5	2,8	2,8	2,8	2,8
4	0,7	3,2	3,1	3,5	3,266666667
5	0,9	4,7	4,7	4,8	4,733333333
6	1,1	5,2	5,4	5,4	5,333333333
7	1,3	6,5	6,5	6,5	6,5
8	1,5	6,8	6,6	6,7	6,7
9	1,7	7,3	7	7,4	7,233333333
10	1,9	7,7	7,7	7,7	7,7
11	2,1	7,7	7,6	6,6	7,3
12	2,3	6,4	6,2	6,6	6,4
13	-0,3	-2,6	-2,9	-2,8	-2,766666667
14	-0,5	-3,4	-3,5	-3,5	-3,466666667
15	-0,7	-4,4	-4,9	-4,4	-4,566666667
16	-0,9	-5,2	-4,8	-5,1	-5,033333333
17	-1,1	-6,2	-6	-6	-6,066666667
18	-1,3	-6,3	-6,2	-6,1	-6,2
19	-1,5	-6,4	-6,4	-6,4	-6,4
20	-1,7	-6,8	-6,7	-6,8	-6,766666667
21	-1,9	-6,6	-6,6	-6,4	-6,533333333
22	-2,1	-6	-6	-6	-6
23	-2,3	-7,2	-7,7	-8	-7,633333333

Fig. 5-3: taula que relaciona tensió d'entrada i força

Ara ja es tenen els valors que relacionen la tensió amb la força però per tenir una relació fiable lo millor no és calcular-la si no utilitzar un bloc del Matlab que s'encarrega de fer-ho ell i a més ho fa amb la proporcionalitat justa en cada relació de tensió i força que li indiquem.

Aquest bloc s'anomena *Look Table*:

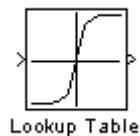


Fig. 5-4: bloc Look Table

Totes les tensions aplicades i els seus corresponents valors de la variable força s'han de configurar dins del bloc:

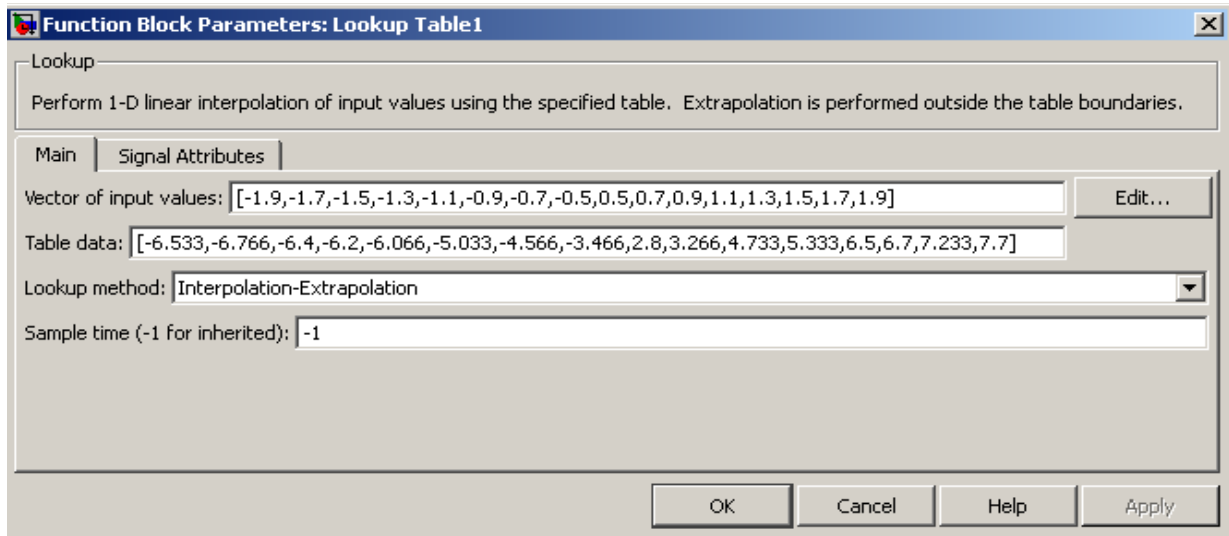


Fig. 5-5: configuració bloc Look Table

Aquest bloc va incorporat en el diagrama de blocs del sistema no lineal i s'ha de utilitzar per linealitzar les equacions amb la instrucció *linmod* i obtenir unes matrius A, B, C, D que representin el sistema de forma més fidedigne. Es pot veure el detall del muntatge a la figura 2-10 del present treball.

## 5.2. Linealització 360°

El segon aspecte a resoldre és el fet de que si el angle gira en sentit horari els valors que dona el sensor son positius i cada més alts sense passar per zero, es a dir que si arriben al punt de equilibri inestable no dona una lectura de 0<sup>a</sup> si no de 360°. Aquí es pretén aconseguir que sigui en sentit antihorari o horari el valor que llegeixi el sensor, si el pèndol es troba a la posició desitjada, sigui zero.

Per aconseguir aquest propòsit es va utilitzar un bloc de Matlab que permet al usuari fer petits programes. Aquest bloc s'anomena *Embedded MATLAB Function*:

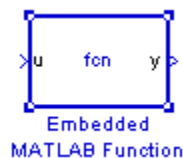
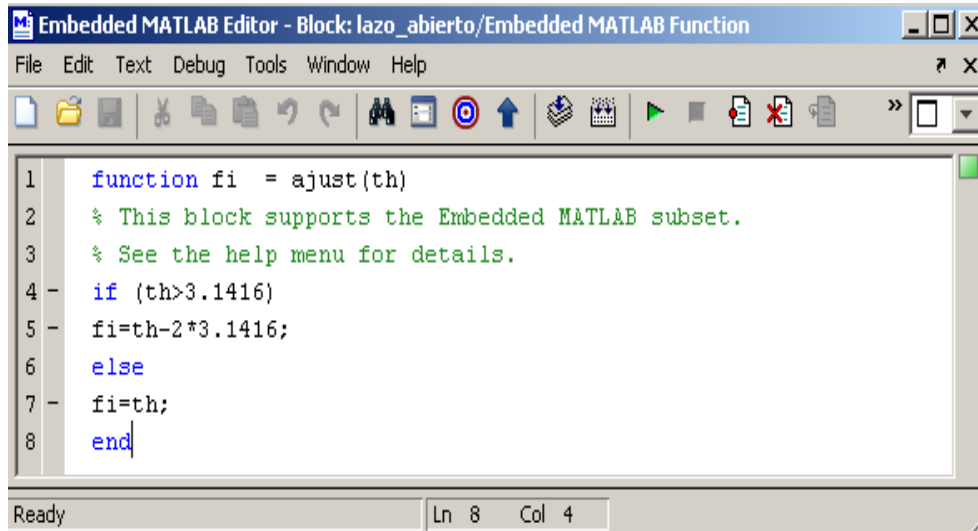


Fig. 5-6: bloc Embedded MATLAB Function

Si es fa doble clic sobre el bloc apareix el editor de Matlab que ens permet escriure el programa que vulguem dintre, aquest funciona amb sentències del tipus *if*, *relop*, *else*, *elseif*, *end*, *for*, *while*.

El programa que s'ha escrit dins perquè el sensor ens doni valor zero quan es trobi el pèndol en la posició de control que ens interessa és el següent:



```

Embedded MATLAB Editor - Block: lazo_abierto/Embedded MATLAB Function
File Edit Text Debug Tools Window Help
1 function fi = ajust(th)
2 % This block supports the Embedded MATLAB subset.
3 % See the help menu for details.
4 - if (th>3.1416)
5 - fi=th-2*3.1416;
6 else
7 - fi=th;
8 end
Ready Ln 8 Col 4

```

Fig. 5-7: programació bloc Embedded MATLAB Function

A on:

- $fi$  és l'entrada del bloc
- $th$  és la sortida del bloc

A l'entrada  $th$  es posarà un bloc anomenat *Abs* que farà que la variable que arribi ho faci sempre amb valors absoluts.

Si la sortida és més gran que  $\pi$  llavors l'entrada prendrà el valor de la sortida menys dos multiplicat per  $\pi$ . Si no es compleix aquesta condició llavors l'entrada i la sortida del bloc seran iguals.

Aquest programa situat a la sortida del bloc de Feedback que correspon al angle soluciona el problema, es pot veure amb detall més endavant, a la figura 5-13.

### 5.3. Aixecament de Feedback

Per aixecar el pèndol s'ha utilitzat el exercici de Feedback Swing Hold Pendulum Extra. La raó es perquè quan es va detectar que la maqueta tenia un problema de pèrdua de força es van testear tots els exercicis de Feedback i a tots els hi afectava aquest problema a excepció del exercici comentat.

Del exercici de Feedback només es va escollir, de totes les parts que el formen, la part que ens interessa:

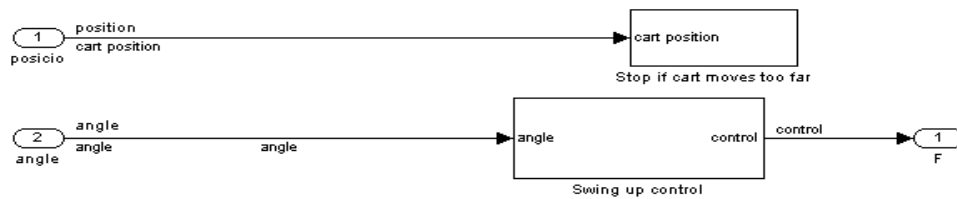


Fig. 5-8: blocs del exercici Swing Hold Pendulum Extra de Feedback

Els blocs que apareix a la figura 5-8 es van introduir dins d'un bloc del tipus *Subsystem* per poder-lo utilitzar en els següents apartats.

## 5.4. Switch

Per commutar el pas de aixecar el pèndol a mantenir-lo aixecat es va utilitzar el bloc de Matlab anomenat *switch*. Aquest bloc té tres entrades i una sortida:

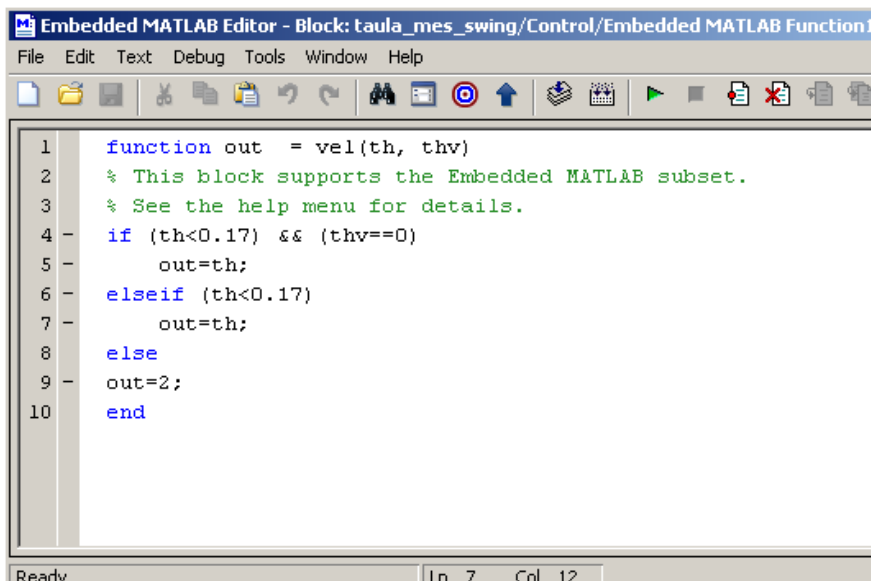


Fig. 5-9: blocs Swieth

L'entrada del mig (a l'esquerra del bloc) serveix per fer arribar un valor i si aquest coincideix amb el valor que es posa a la configuració del bloc, i segons les condicions que se li doni (si és major que o menor que), commutarà entre els contactes dels extrems del bloc.

La commutació, en el cas que ens ocupa, s'havia de fer en les condicions més favorables i aquestes són dues: que el angle es trobi per sota dels deu graus (0,17 radians) i que la velocitat angular sigui zero.

Per aconseguir aquest propòsit es va programar un bloc *Embedded MATLAB Function* com el de la figura 5-6. El programa que es va escriure amb l'editor es aquest:



```

1  function out = vel(th, thv)
2  % This block supports the Embedded MATLAB subset.
3  % See the help menu for details.
4  if (th<0.17) && (thv==0)
5      out=th;
6  elseif (th<0.17)
7      out=th;
8  else
9      out=2;
10 end

```

Fig. 5-10: programació bloc Embedded MATLAB Function

A on:

- $th$  és una entrada que dóna el valor de  $theta$  (situació del angle)
- $thv$  és una entrada que dóna el valor de la velocitat angular de  $theta$
- $out$  és la sortida del bloc

Si l'entrada  $th$  és menor que 0,17 radians i l'entrada  $thv$  és igual a zero llavors la sortida  $out$  serà igual a l'entrada  $th$ . Es a dir prendrà el valor real del angle.

Però si no es compleix la condició d'abans i la  $th$  és menor que zero llavors també la sortida  $out$  serà igual a l'entrada  $th$ . En cas contrari la sortida  $out$  serà igual a dos.

La sortida del bloc programat es connecta a la entrada del mig del switch i serà la consigna que el faci commutar.

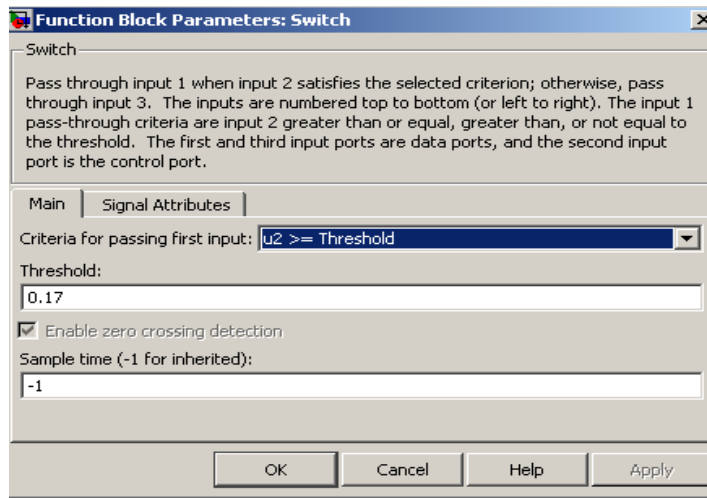


Fig. 5-11: configuració del bloc switch

Aquesta configuració farà que la sortida del switch estigui connectada a l'entrada del extrem superior si el angle té valors més alts que 0,17 radians i en el moment que l'angle prengui valors més petits que 0,17 i la velocitat angular sigui zero llavors commutarà a l'entrada del extrem inferior.

A l'entrada superior es connectarà la primera etapa del exercici, aixecar el pèndol, i a la inferior el control perquè el pèndol es mantingui als zero radians.

## 5.5. Implementació del retorn d'estat

Primer cal comentar que les primeres proves al sistema real es feien simplement amb la part de control per retorn d'estat per mantenir el pèndol aixecat; per aixecar-lo es feia a mà ja que aquesta part va ser l'ultima en dissenyar-se. Per fer-ho a mà es posava en marxa el diagrama de blocs muntat al Matlab però amb el botó verd del driver apagat. S'aixecava el pèndol a mà i després de tenir el controlador iniciat al Matlab es polsava el botó verd del driver.

Les gràfiques que sortiran a aquesta i les següents seccions seran el resultat del swing més el retorns d'estat, observador i control integral segons sigui el cas.



El diagrama de blocs muntat al Matlab per implementar el disseny és aquest:

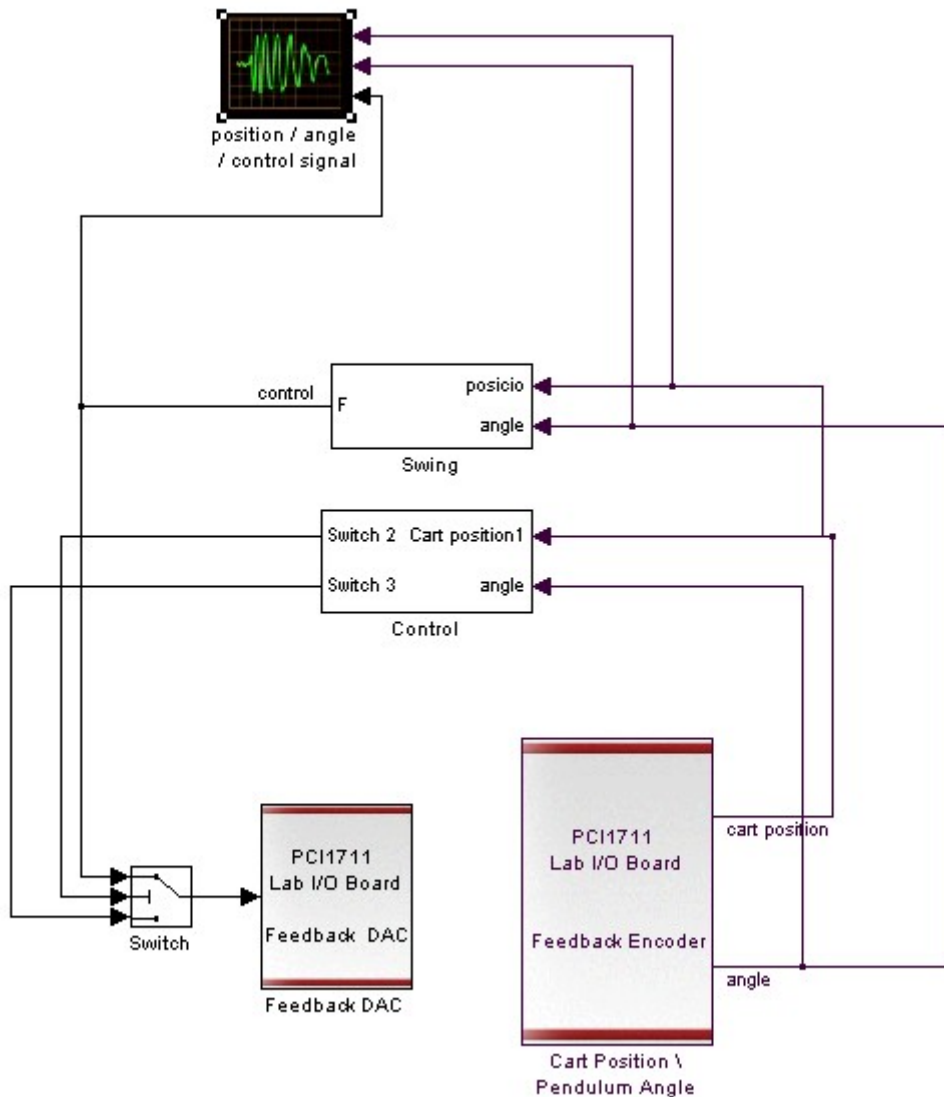


Fig. 5-12: diagrama de blocs del control per retorn d'estat

Els blocs Swing i Control són blocs de subsistemes. El primer conté dintre el diagrama de blocs de la figura 5-8 que serveix per aixecar el pèndol i el segon el de la figura 5-13 que mantindrà controlat el pèndol als zero radians.

El bloc *Feedback DAC* és l'entrada del sistema (a on s'aplica la senyal de control) i el bloc *Cart Position \ Pendulum Angle* són les sortides del sistema: angle del pèndol i posició del carro; les derivades d'aquestes dues variables s'aconsegueixen col·locant blocs derivatius del Matlab, es pot veure amb detall a la figura 5-13.

El bloc *position \ angle \ control signal* és un oscil·loscopi.

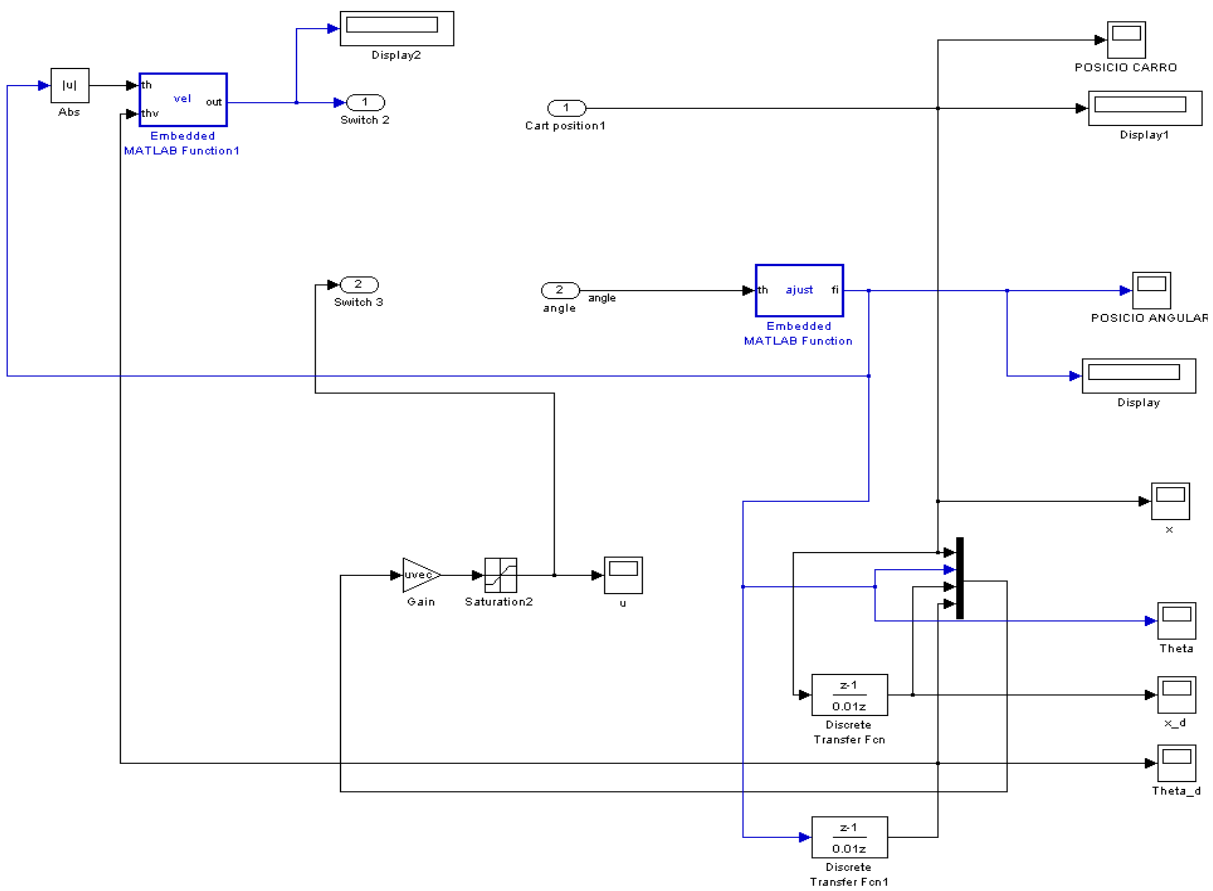


Fig. 5-13: diagrama de blocs intern del bloc Control

Resultats de l'aplicació de la matriu de realimentació  $Kd$  (bloc Gain figura 5-13) sobre el sistema real:

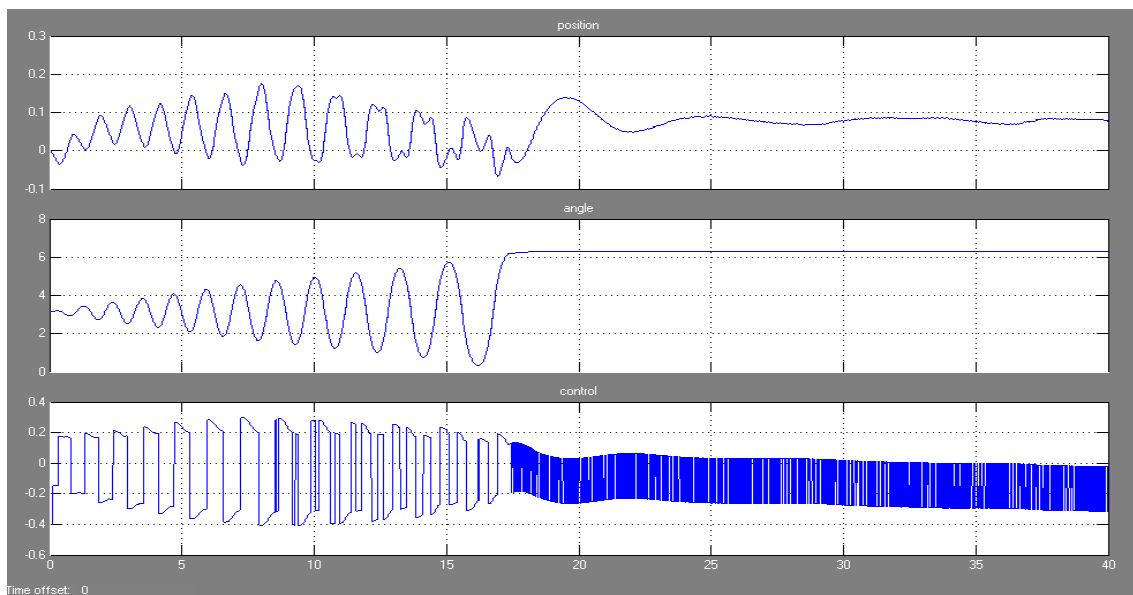


Fig. 5-14: gràfiques resultat de l'aplicació del disseny de  $Kd$  sobre el sistema real

A la figura 5-14 s'observa que els resultats són els desitjats. Pel que fa a la posició, primera gràfica, es mou entre els valors -0,1 i 0,15, no es desplaça gaire, recordar que aquest swing és el de

Feedback plc.

El pèndol, segona gràfica de la figura 5-14, comença a oscil·lar fins arribar a prop dels 6,28 radians, segon 17,5 aproximadament, llavors hi ha la commutació i la seva posició s'estabilitza al voltant dels zero radians. En aquesta gràfica es pot comprovar l'acció desenvolupada al apartat 5.2: Linealització 360° ja que el pèndol s'aixeca finalment en sentit horari.

A la tercera gràfica de la figura és veu l'acció de control que des de el segon 0 al 17 té uns valors més alts ja que la força per aixecar el pèndol es major que la de control al voltant dels zero radians, des de els 17 fins als 40 segons.

A la següent figura es veu el mateix exercici però amb pertorbacions: s'empenta el pèndol amb la mà.

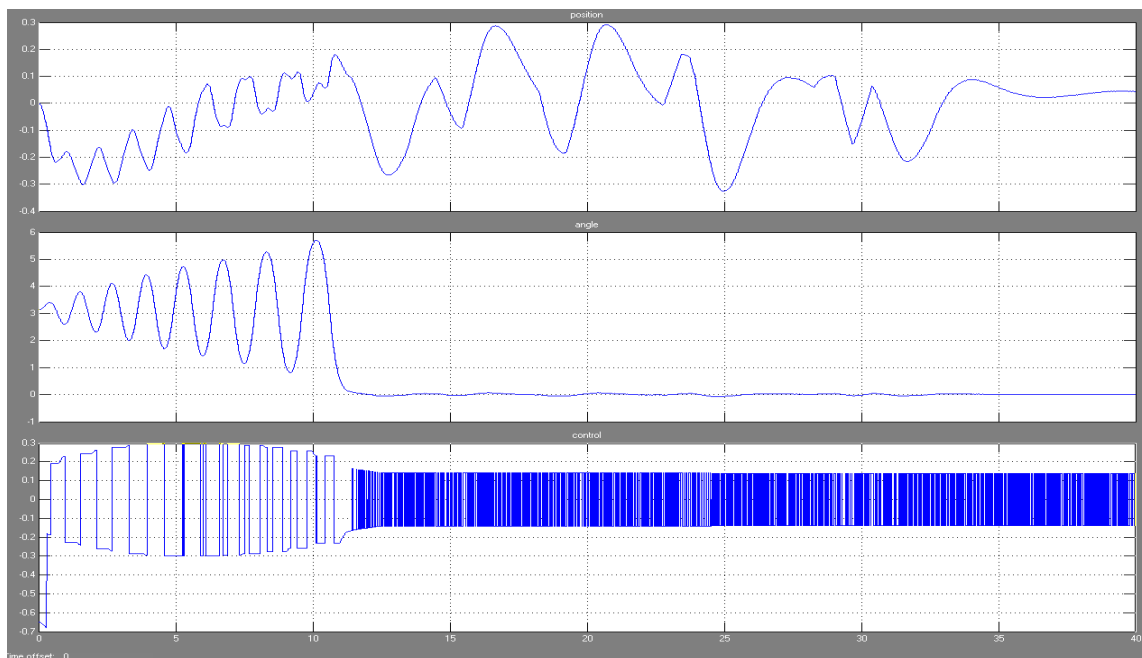


Fig. 5-15: gràfiques resultat de l'aplicació del disseny de  $Kd$  sobre el sistema real amb pertorbacions

Aquesta vegada el control per retorn d'estat s'aplica quan el pèndol arriba als zero radians, no als 6,28 com el cas de la figura anterior.

Els resultats de la figura 5-15 mostren el exercici amb pertorbacions (cops amb la mà). A la segona gràfica de la figura s'observa la oscil·lació del pèndol fins als 12 segons, moment en que es produeix el *switch*, i és a partir d'aquí fins al segon 35 que se li sotmet a pertorbacions.

A la primera gràfica es veu com la posició del carro varia per aconseguir que el pèndol no caigui davant la dificultat introduïda.

L'acció de control, última gràfica de la figura 5-15, és oscil·lant però, en els moments de

pertorbacions, es manté a una valors de  $\pm 0,15V$ .

Finalment a la figura 5-16 es mostra el exercici per retorn d'estat però amb el control difús per al *swing*.

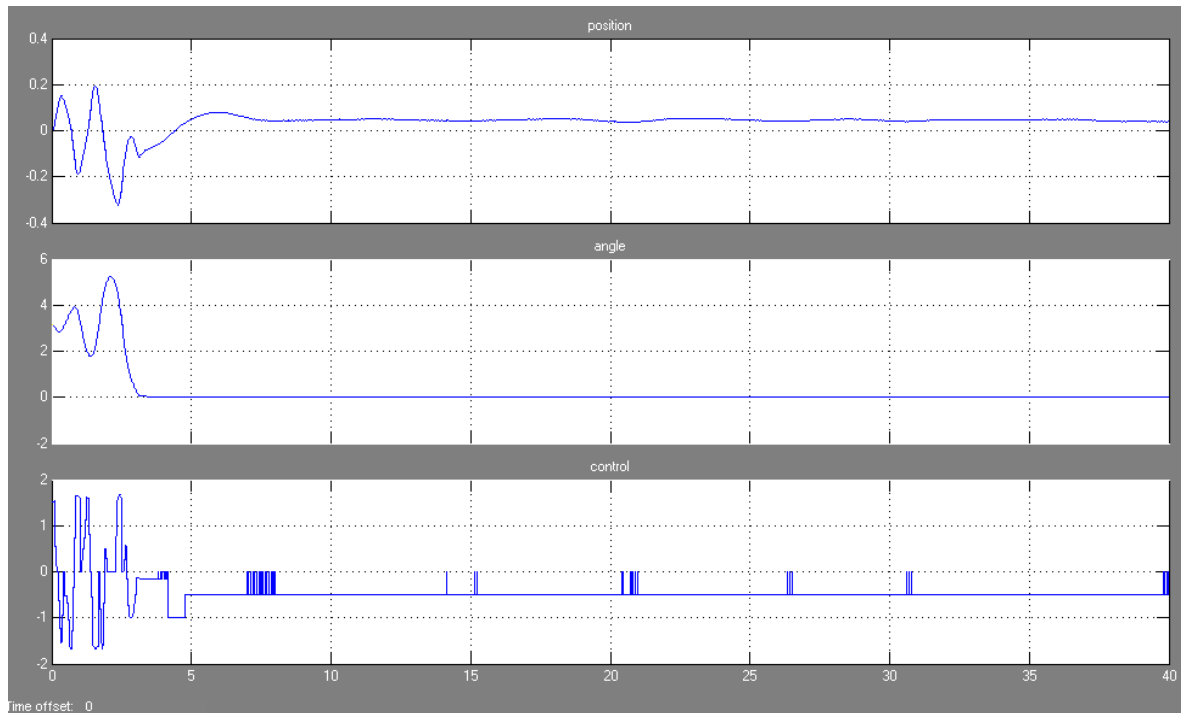


Fig. 5-16: gràfiques resultat de l'aplicació del disseny de  $Kd$  sobre el sistema real amb control difús al *swing*

La figura 5-16 mostra les tres gràfiques novament amb  $x$ ,  $\theta$  i l'acció de control. Destacar la ràpida acció del control difús (s'havia dissenyat amb aquest propòsit) que fa arribar al pèndol en 3,5 segons aproximadament a la posició zero radians (segona gràfica).

L'acció de control amb control difús al *swing*, una vegada aixecat el pèndol, es manté gaire bé constant als  $-0,5V$ . Aquest fet no s'explica, sembla l'acció d'un filtre però el muntatge és el mateix que l'anterior, a excepció del comentat bloc *Swing*, de la figura 9-11.

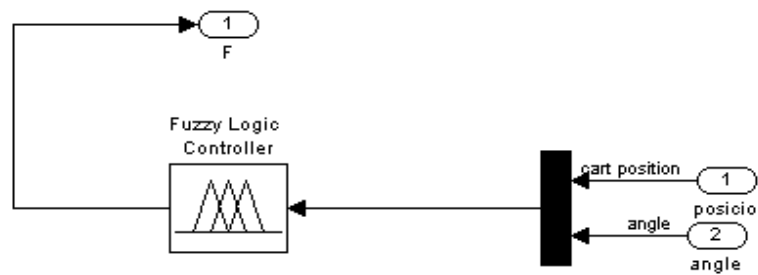


Fig. 5-17: blocs interns del bloc Swing amb controlador difús

Els resultats dels controladors, tant el difús com el retorn d'estats es poden considerar òptims, per tant es donen per bons els càlculs efectuats amb un temps de mostreig de 0,01 segons.

## 5.7. implementació del observador d'ordre complet més retorn d'estat

Exteriorment el diagrama de blocs de Matlab al exercici del observador o estimador d'estat és el mateix que el de la figura 5-12 a excepció del diagrama de blocs intern del bloc Control, que està dissenyat per actuar com un observador.

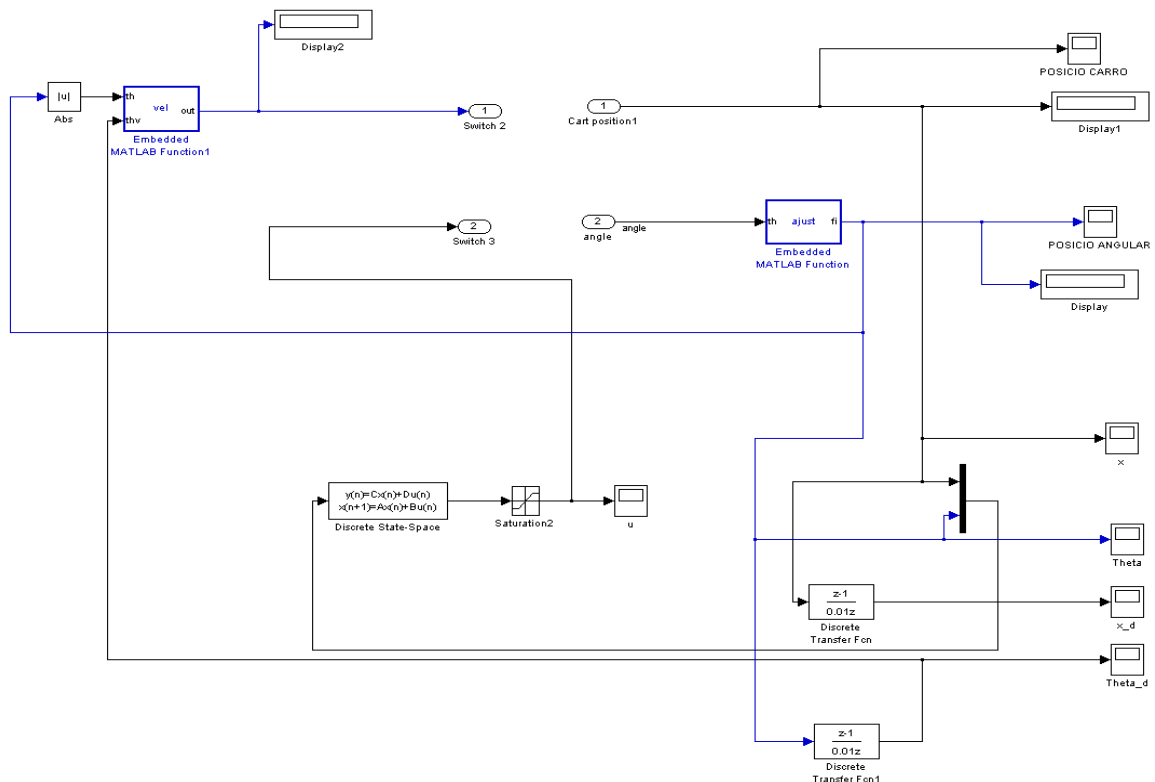


Fig. 5-18: diagrama de blocs intern del bloc Control per a un observador d'ordre complet.

La diferència es troba en el bloc Discret Estate Space que està configurat amb les matrius  $a_{con}$ ,  $b_{con}$ ,  $c_{con}$ ,  $d_{con}$  del apartat 3.3.1 del capítol 3. La configuració del bloc és la mateixa que la de la figura 3-19.

Els resultats de l'execució del exercici al sistema real s'observen a la figura 5-19.

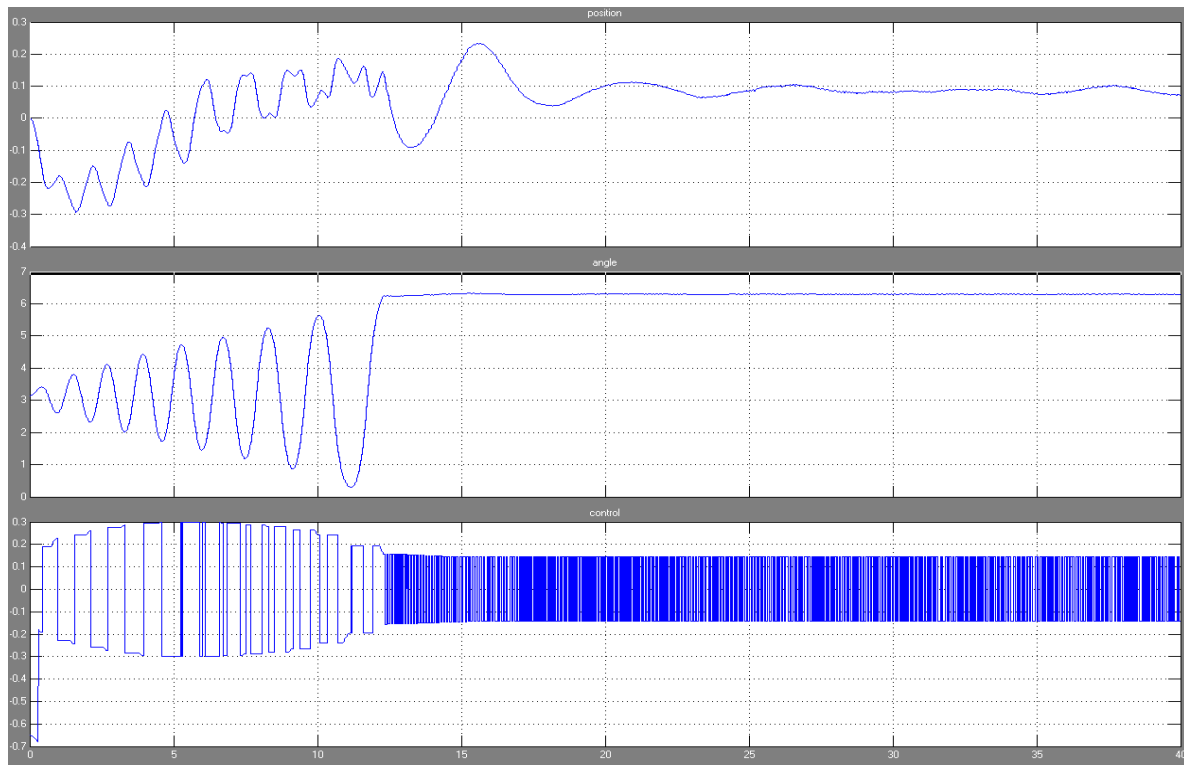


Fig. 5-19: gràfiques resultat de l'aplicació del disseny del observador sobre el sistema real

La primera gràfica és la posició i es veu amb petites oscil·lacions per aixecar el pèndol, al segon 13 aproximadament. A partir d'aquest moment fa uns moviments bruscos que es deuen a l'acció del *switch* ja que a la segona gràfica es comprova que el pèndol ha arribat a la posició de commutació en aquest instant.

A la segona gràfica de la figura 5-19 la resposta és l'esperada: oscil·lacions fins a arribar al voltant dels 0,17 radians i després es manté controlada als zero radians.

La acció de control, tercera gràfica de la figura 5-19, aplica una tensió més alta per aixecar el pèndol, al igual que les proves anteriors, i finalment es manté oscil·lant entre  $\pm 0,15V$  regularment.

El comportament del sistema amb un observador d'ordre complet és bo des de punt de vista del control. El sistema, amb l'estimació de les derivades de la posició de carro i del pèndol, es manté controlat durant tot els temps de l'execució del exercici.

## 5.8. implementació control integral més retorn d'estat

La implementació del control integral té una petita variació respecte als exercicis anteriors en quant al diagrama de blocs.

El cas és que com què té com a entrada un senyal extern per poder fer el switch s'havia de tenir en compte que la posició del carro fos igual que el del senyal d'entrada en el moment de la commutació, ja que el sistema a de seguir aquest senyal. A més de complir les condicions anteriors: que el angle del pèndol es trobés en una posició entre  $\pm 0,17$  radians i la velocitat angular fos zero.

Per poder dur a terme aquest exercici es va pensar en una triple acció: swing, control per retorn d'estat, fins que l'entrada fos igual a la posició del carro, i control integral més retorn d'estat. Per tant es necessita un diagrama de blocs amb dos blocs *switch*.

Diagrama de blocs que es va construir és el de la següent figura.

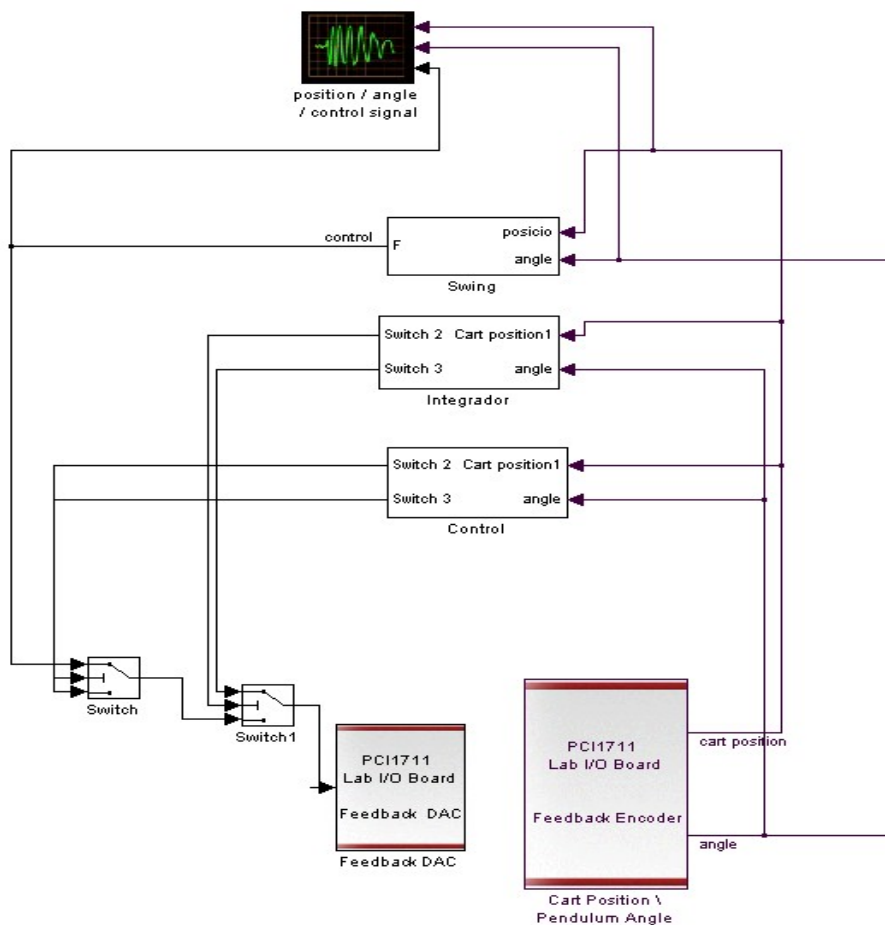


Fig. 5-20: diagrama de blocs del control per retorn d'estat més integrador

Al bloc Swing és el mateix utilitzat fins ara, el de Feedback, és també un bloc *subsystem* amb el control per retorn d'estat del apartat 5.5 i el bloc *integrador* conté dintre el diagrama de blocs del control per retorn d'estat més integrador.

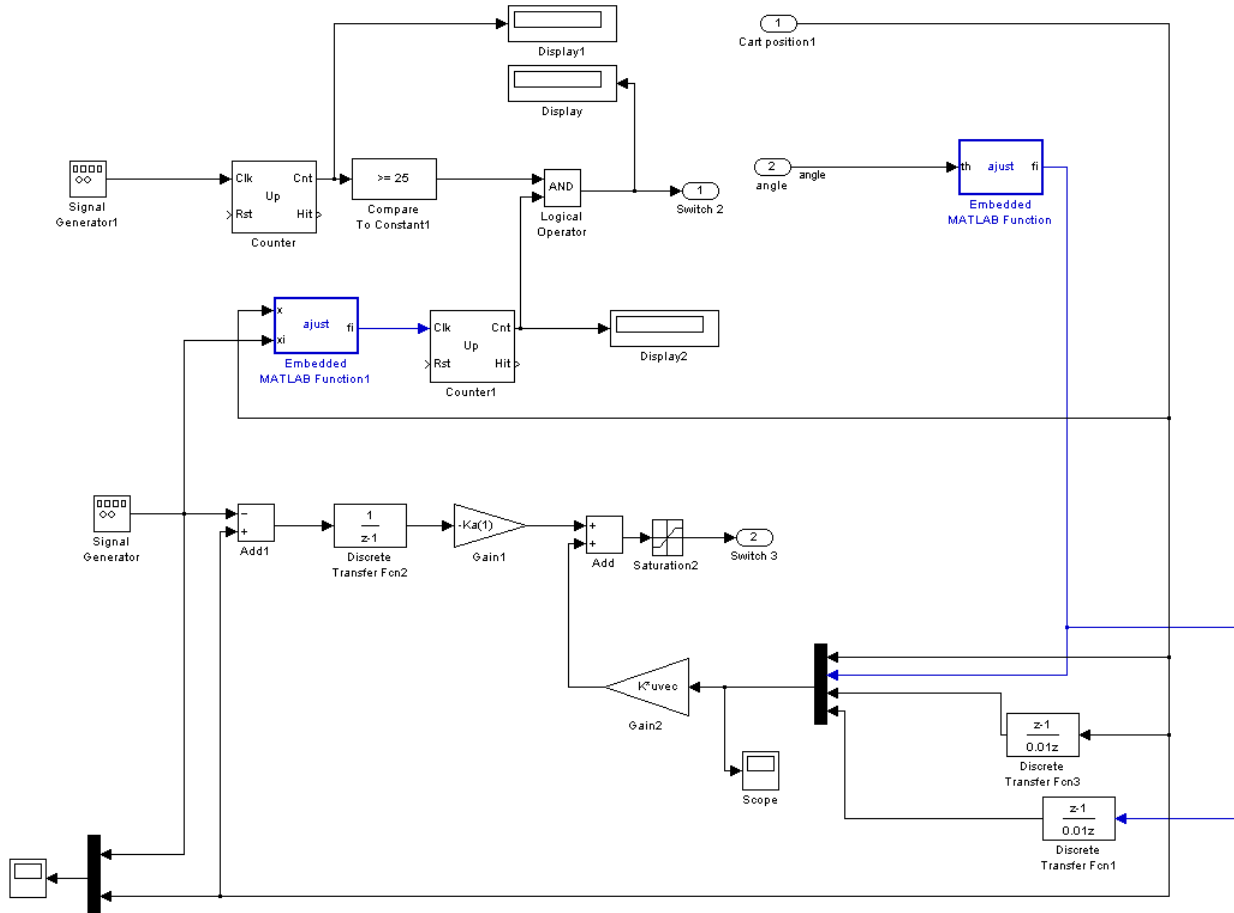


Fig. 5-21: diagrama de blocs intern del bloc integrador

Si ens fixem en la figura 5-21 tenim dos blocs del tipus switch: *switch* i *switch1*. El primer dels dos té la mateixa configuració del de l'apartat 5.5, de fet és el mateix.

El *switch1* té al connector de la entrada inferior el senyal del bloc *switch* i està programat perquè commuti de forma inversa al primer *switch*, es a dir del connector de l'entrada inferior al superior.

Com a senyal per commutar (connector de l'entrada central del *switch1*) té configurat un valor de 1. Aquest valor de 1 és un valor lògic que li arribarà de la sortida 2 (*switch2*) del bloc Integrador de la figura 5-21 i que donarà pas, en cas de que es compleixin certes condicions, al control integral.

Aquestes condicions són les següent:

- Que hagin passat 25 segons des de el inici del exercici a la maqueta.
- Que la posició del carro  $x$  sigui igual a la senyal d'entrada, sigui aquesta un senyal quadrat, triangular, etc.



Perquè es compleixin aquestes dues condicions en el moment de la commutació s'ha afegit un operador lògic AND, el qual donarà un valor de 1 si es compleixen aquestes condicions.

Perquè passin els 25 segons s'ha fet el que mostra la següent figura.

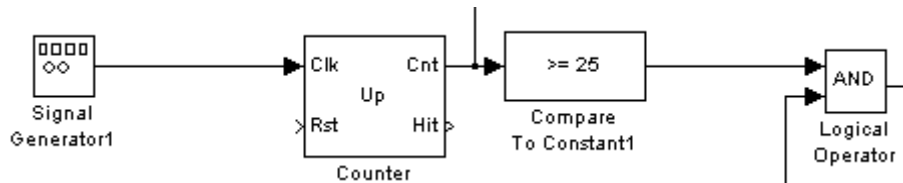


Fig. 5-22 comptador de segons

El generador de senyals està configurat amb un senyal quadrat amb una amplitud de 1 i una freqüencial de 1 d'aquesta manera, el comptador, actua com un cronòmetre. La sortida Cnt del comptador arriba a un comparador i aquest deixa passar el valor d'entrada si supera els 25 segons, provocant un 1 al operador AND.

La altre condició es compleix sota les premisses d'un petit programa escrit en un bloc *Embedded MATLAB Function*. El programa és aquest:

```

1  function fi = ajust(x,xi)
2  % This block supports the Embedded
3  % See the help menu for details.
4  if (x==xi)
5  fi=1;
6  else
7  fi=0;
8  end

```

Fig. 5-23 comptador de segons

A on:

$x$  és una entrada del bloc amb el valor de la posició de carro,

$xi$  és una entrada del bloc amb el valor de la posició del senyal d'entrada extern al sistema

$fi$  és la sortida del bloc

La sortida  $fi$  serà igual a 1 si les entrades  $x$  i  $xi$  són iguals, si no es compleix aquesta condició llavors la sortida  $fi$  serà igual a 0.

Aquest bloc estarà connectat a un comptador que donarà a la sortida un 1 si el programa es posa a valor 1. Aquesta part del diagrama juntament amb comptador que fa de cronòmetre posaran aquesta la porta AND a 1 en el moment adequat.

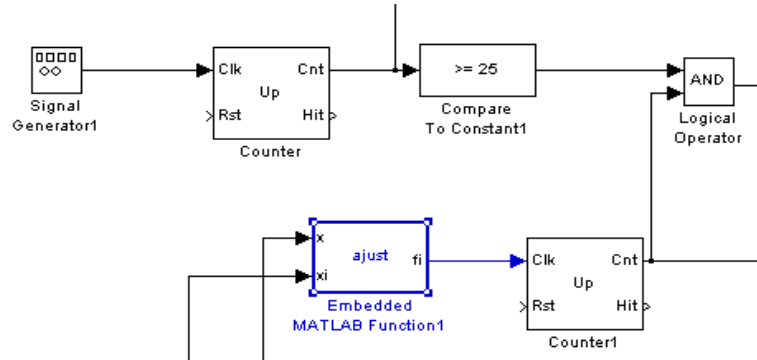


Fig 5-24: Detall del muntatge per commutar al control integral

S'ha recorregut als comptadors perquè aquests, amb el reset lliure, guarden els valors fins que acaba el exercici al Matlab, si no el bloc *Embedded MATLAB Function* es posava a zero en quant  $x \neq x_i$  i la porta deixava d'actuar sobre el *switch1*.

Primer es mostraran els resultats de l'execució del exercici amb el següent protocol: aixecar el pèndol a mà, iniciar el exercici al Matlab i finament polsar el botó dret del driver.

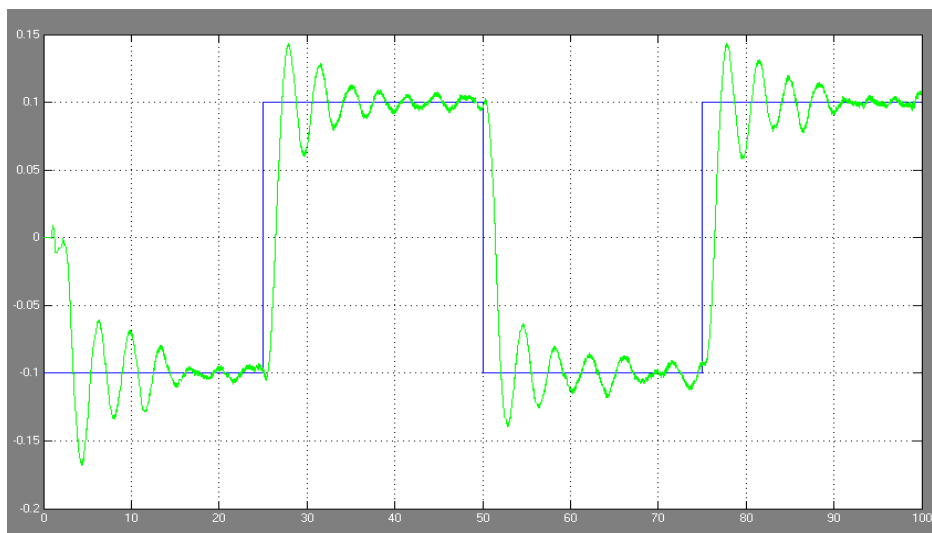


Fig. 5-25: gràfic d'un senyal quadrat d'entrada, integrador amb pols de Bessel

Al gràfic de la figura 5-25 el senyal de color verd és la posició del carro amb el pèndol aixecat i el senyal de color blau és el senyal d'entrada amb una freqüència de 0,02 hertz. La imatge mostra com el carro segueix el senyal que té el sistema d'entrada.

Aquesta part del exercici es va fer amb els pols de Bessel. Cada vegada que el senyal d'entrada

canvia el carro oscil·la al seu voltant fins que s'estabilitza a sobre.

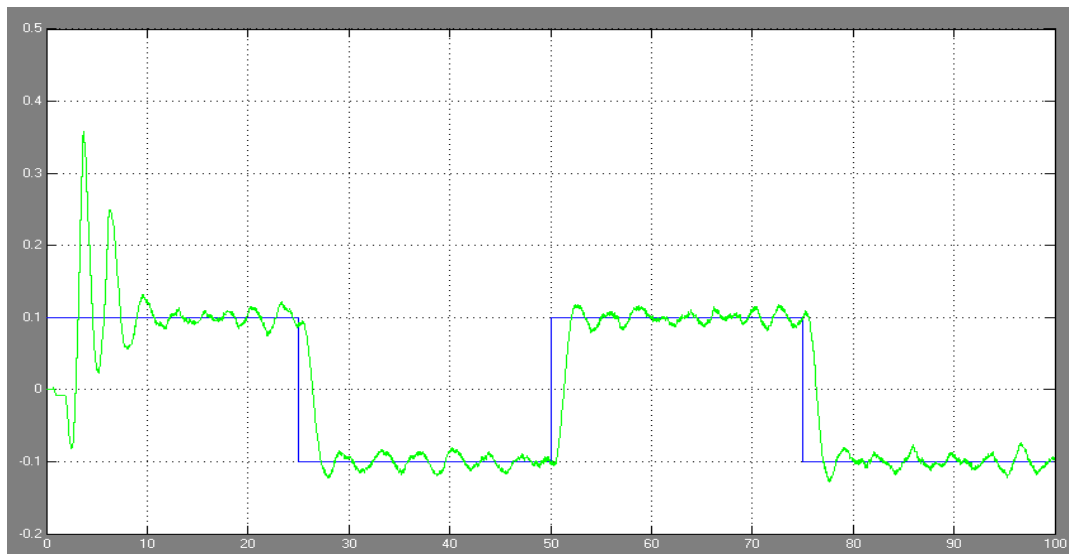


Fig. 5-26: gràfic d'un senyal quadrat d'entrada, integrador amb pols LQR

A la figura 5-26 tenim el mateix exercici però els càlculs de  $Ka$  estan fets amb el pols LQR:

$$Ka = [-0.3438 \quad -45.9324 \quad 138.1285 \quad -29.6546 \quad 53.2232]$$

Les oscil·lacions que es veuen al principi ocorren perquè el pèndol comença en una posició massa lluny perquè el control sigui efectiu, per tant el pèndol s'aguanta amb la mà fins que els valors  $x$  i  $x_i$  coincideixen.

La resposta amb el pols LQR és encara millor, davant els canvis del senyal d'entrada el carro respon molt bé, oscil·la molt poc.

S'han fet més proves amb senyals triangular amb la mateixa freqüència de 0,02 hertz, primer amb el pols de Bessel:

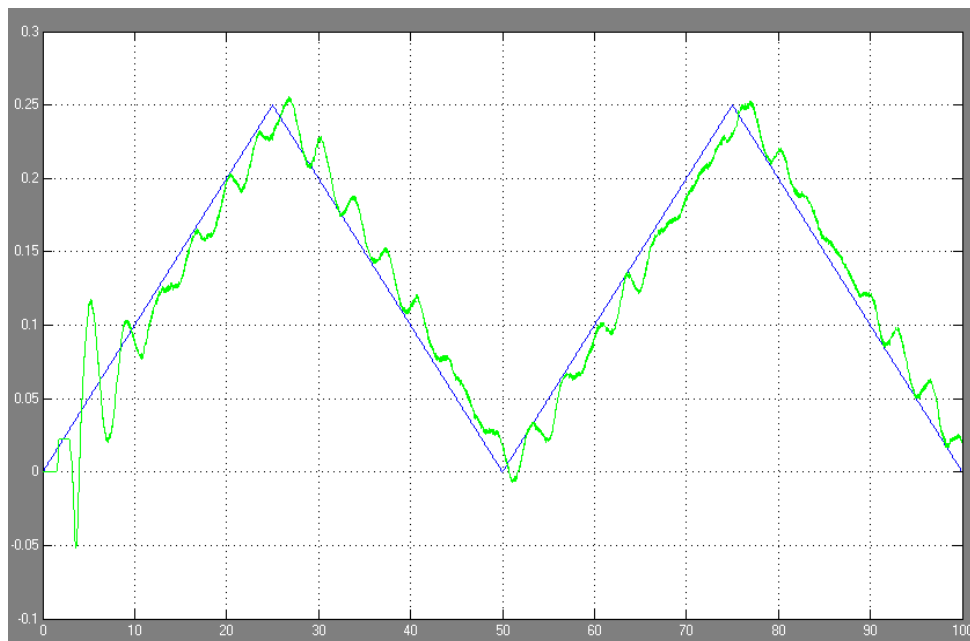


Fig. 5-27: gràfic d'un senyal triangular d'entrada, integrador amb pols de Bessel

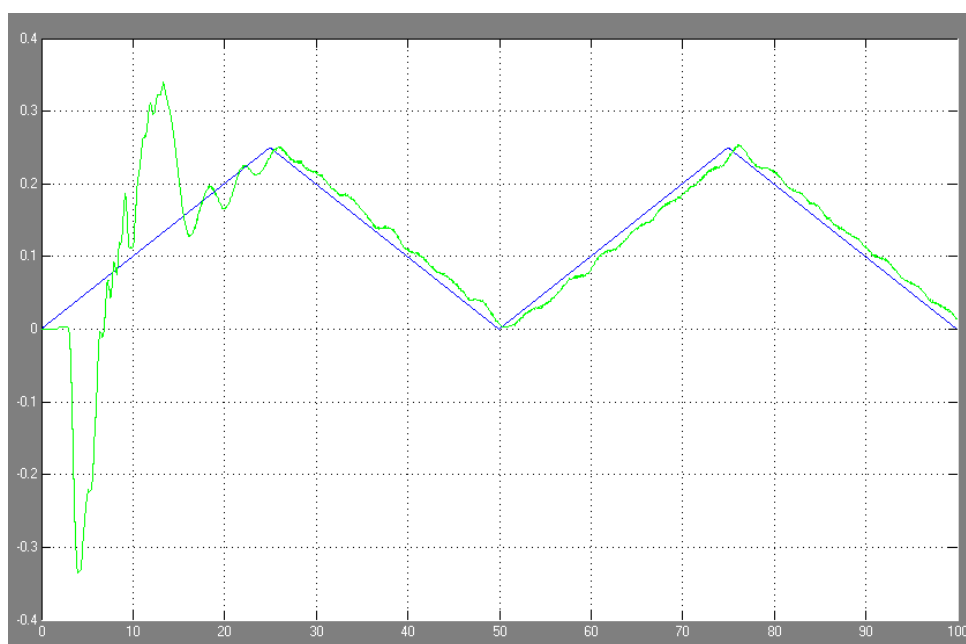


Fig. 5-28: gràfic d'un senyal triangular d'entrada, integrador amb pols LQR

Els resultats són també molt positius en els dos casos. El seguiment d'un senyal sobre l'altre és molt acceptable.

Per finalitzar es mostra una nova figura amb el procés complert: swing, control per retorn d'estat i control per retorn d'estat més integrador, amb una freqüencial de 0,1 hertz.

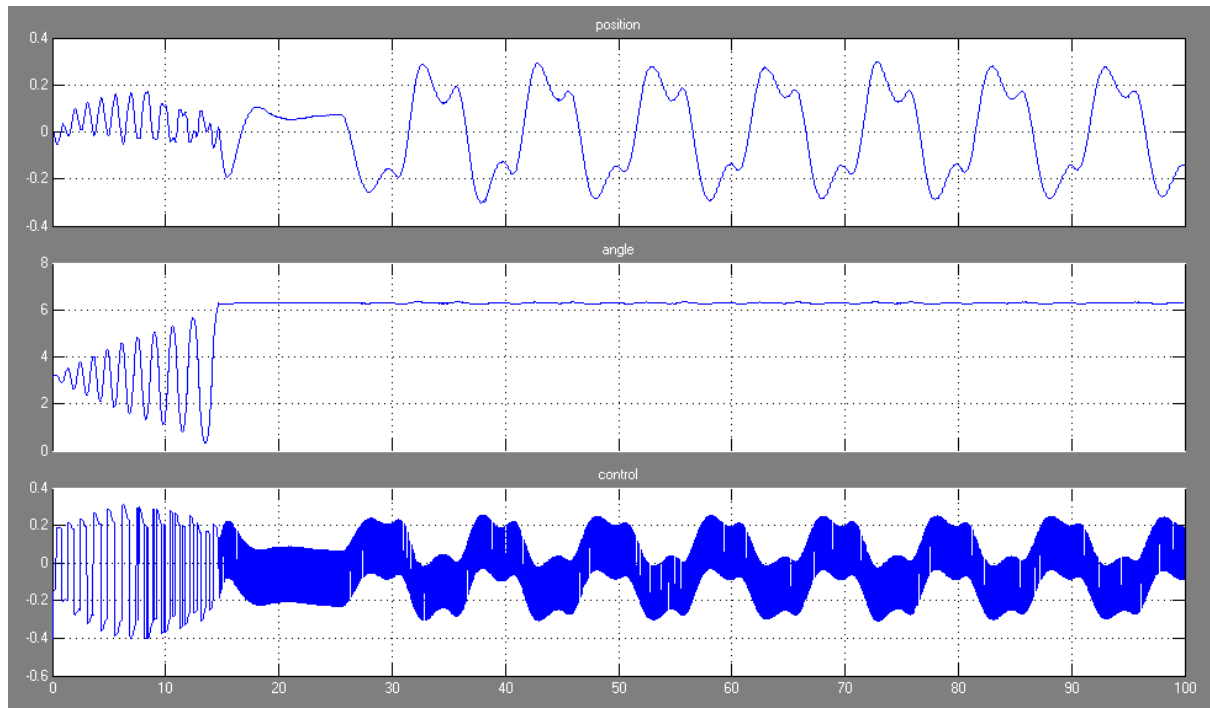


Fig. 5-29: gràfic d'un senyal quadrat d'entrada, integrador amb pols de Bessel

Aquesta última figura mostra com primer el pèndol oscil·la fins arribar a la posició 6,28 radians aproximadament sobre el segon 17; a partir d'aquí comença el control per retorn d'estat fins al segon 25; i finalment des de el segon 25 fins al 100 actua la suma del integrador i retorn d'estat seguint el senyal quadrat d'entrada.

Destacar que aquest muntatge a la pràctica només s'ha aconseguit amb èxit un 30% de les vegades que s'ha provat.

La raó és perquè el bloc *switch1* en aquest cas no és lo suficientment ràpid. Als altres casos (observador i retorn d'estat) amb el *switch* si que ho era perquè tenia el temps que transcorre des de que el pèndol passa als 0,17 radians fins als -0,17 radials o vice versa.

Amb aquest muntatge té la meitat de temps ja que el pèndol es troba als zero radians i cau cap a un costat en el moment que el *switch1* commuta.



## **6. Conclusions**

### **6.1. Sobre Matlab**

Els coneixements en matèria de control sobre Matlab des del inici d'aquest projecte fins finalitzar-lo han estat quantiosos. S'han aplicat mètodes de control com el retorn d'estat, observador i control integral desenvolupant-los en tot allò que ha estat possible amb Matlab com el cas de la linealització amb linmod, l'obtenció de pols per LQR, etc. A més s'han provat diverses formes de muntatge de diagrames de bloc amb Simulink per comprovar les amplies possibilitats que contempla aquesta eina.

Es pot concloure que el coneixement sobre Matlab i Simulink a augmentat significativament en aquest treball i que ara s'està molt més familiaritzat amb l'entorn d'aquesta excel·lent eina.

### **6.2. Sobre els algoritmes de control**

S'han dissenyat pel control que estabilitza el pèndol mètodes basats en el espai-estat amb diferents formes d'obtenció de pols. Els resultats tant en simulacions com en l'aplicació d'aquests a la maqueta s'han vist als capítols 3, 4 i 5 i els resultats de tots ells es força satisfactori. Una vegada calculat el algoritme de control no ha calgut fer cap modificació, això vol dir que la deducció del sistema amb equacions ha estat correcte i que ha facilitat la tasca en matèria de càlculs.

### **6.3. Lògica difusa**

La lògica difusa ha estat l'estratègia escollida per aixecar el pèndol. Tot i que ja ha hagut altre treball a l'EUPMT sobre aquesta maqueta no s'havia treballat sobre aquest aspecte: portar-lo al punt d'equilibri inestable.

Els experiments en aquest sentit els mostra el capítols 5 i 6, i s'observa que el propòsit s'ha assolit. Però la maqueta té la dificultat ja comentada en aquests mateixos capítols, que la força que es pot injecta al sistema disminueix progressivament, i que per recuperar el nivell òptim de treball de sistema s'ha d'anar apagant la maqueta. Aquest problema ha dificultat la millora d'aquesta part que tot i que té un bon comportament es podria millorar.

### **6.4. Estudis futurs**

Encara queden alguns aspectes per treballar i millorar que pot donar per el desenvolupament d'un TFC. El pèndol ofereix al alumne introduir mètodes de control de diferents menes.

El control fuzzy sobre el punt d'equilibri es va dur a terme al projecte anterior però es pot aprofundir aquest aspecte i millorar-lo. També s'haurien de desenvolupar controladors PID i, d'aquesta manera, contribuir a un estudi més complet sobre el sistema.

També es pot aixecar el pèndol per altres mètodes com per exemple una excitació a la freqüència natural per fer-lo entrar en ressonància o amb el disseny d'un PID.



## 7. Bibliografia

[1] C-T Chen. Linear System Theory and Design. Oxford University Press, 1999.

[2] [http://ethesis.nitrkl.ac.in/2213/1/Inverted\\_Pendulum.pdf](http://ethesis.nitrkl.ac.in/2213/1/Inverted_Pendulum.pdf)

[3] Gene F. Franklin, J. David Powell, Abbas Emami Naeini, Control de sistemas dinámicos con retroalimentación

[4] Roberto Martínez, Estudi i disseny de sistemes de control avançats: El pèndol invertit.

[5] EUPMT, Apunts de Control Avançat

[6] [http://ib.cnea.gov.ar/~instyct/Tutorial\\_Matlab\\_esp/invpen.html](http://ib.cnea.gov.ar/~instyct/Tutorial_Matlab_esp/invpen.html)

Pèndol invertit

[7] <http://redalyc.uaemex.mx/src/inicio/ArtPdfRed.jsp?iCve=85201006>

Lògica difusa

[8] [http://gb.stut.edu.tw/site/ir.lib.stut.edu.tw/bitstream/987654321/3751/2/1992IEEE\\_IE\\_A\\_Hybrid\\_Control\\_Approach\\_for\\_Pendulum-Car\\_Control.pdf](http://gb.stut.edu.tw/site/ir.lib.stut.edu.tw/bitstream/987654321/3751/2/1992IEEE_IE_A_Hybrid_Control_Approach_for_Pendulum-Car_Control.pdf)

Lògica difusa

# Escola Universitària Politécnica de Mataró

Centre adscrit a:



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA

**Enginyeria Tècnica Industrial Especialitat en Electrònica Industrial**

**Disseny de Controladors Avançats Aplicats a la Maqueta del Pèndol Invertit**

**Estudi econòmic**

**ALFREDO COLOMERA CASTELLANO  
PONENT: MIQUEL ROCA CISA**

TARDOR 2011

Escola Universitària  
Politécnica de Mataró



TecnoCampus  
Mataró-Maresme



# Índex

1. Costos d'instrumentació i material.....	1
1.1 Amortització d'equips, instrumental i software.....	1
1.2 Costos de recursos humans.....	1
1.3 Pressupost final.....	2



## 1. Costos d'instrumentació i material

Les hores empleades en la elaboració del projecte han estat 400

### 1.1 Amortització d'equips, instrumental i software.

Concepte	Hores empleades	Preu / hora (€)	Suma (€)
<i>Ordinador</i>	350	0,5	175
<i>Maqueta pèndol</i>	300	2	600
<i>Software Matlab</i>	300	2	600
<i>LibreOffice</i>	50	1	50
<i>Suma</i>	1000	4,5	1425

### 1.2 Costos de recursos humans

Concepte	Hores empleades	Preu / hora (€)	Suma (€)
<i>Estudi i documentació</i>	50	60	3000
<i>Muntatge i posta a punt de la maqueta</i>	3	30	90
<i>Disseny de controladors</i>	200	60	12000
<i>Redacció memòria final</i>	50	30	1500
<i>Suma</i>	303	180	16590

### 1.3 Pressupost final

Concepte	Preu (€)	Suma (€)
<i>Amortització d'equips, instrumental i software</i>	1375	1425
<i>Costos de recursos humans</i>	16590	16590
<i>Total</i>		18015

# Escola Universitària Politécnica de Mataró

Centre adscrit a:



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA

**Enginyeria Tècnica Industrial Especialitat en Electrònica Industrial**

**Disseny de Controladors Avançats Aplicats a la Maqueta del Pèndol Invertit**

**Annexos**

**ALFREDO COLOMERA CASTELLANO  
PONENT: MIQUEL ROCA CISA**

TARDOR 2011



**TecnoCampus  
Mataró-Maresme**





# Índex

Annex I. Programa de control per retorn d'estat.....	1
Annex II. Programa de control per retorn d'estat més observador (continu).....	2
Annex III. Programa de control per retorn d'estat més observador (discret).....	3
Annex IV. Programa de control per retorn d'estat més integrador (continu).....	4
Annex V. Programa de control per retorn d'estat més integrador (discret).....	5



## Annex I. Programa de control per retorn d'estat

```
M=2.4;
I=0.099;
b=0.05;
L=0.4;
d=0.005;
m=0.23;
g=9.81;
Ts=0.01;
T=Ts
[A, B, C, D]=linmod('modelnolin_taula',[0;0;0;0],0) %linealització sistema
sys=ss(A,B,C,D) % es construeix sistema
Pc=[-3.1811+3.0446i, -3.1811-3.0446i, -2.4446+0.4400i, -2.4446-0.4400i] % Pols
a partir de Bessel de 4 ordre *2
K=acker(A,B,Pc) % matriu K
sysd=c2d(sys,Ts) %es discretitza el sistema
[Ad Bd Cd Dd]=ssdata (sysd) %s'extreuen matrius discretes
%[Ad,Bd,Cd,Dd]=c2dm (A,B,C,D,Ts,'zoh')%discretitzat del sistema.
Pd=[0.975827253+0.0043i, 0.975827253-0.0043i, 0.968241+0.029i, 0.968241-0.029i]
%Pols per al sistema discret en el pla Z
Kd=place(Ad, Bd, Pd) %K per sistema discret
co = ctrb (Ad,Bd) %controlabilidad del sistema:
Controlabilitat = rank (co)
ob = obsv (Ad,Cd) % Observabilidad del sistema:
Observabilitat = rank (ob)
```

## Annex II. Programa de control per retorn d'estat més observador (continu)

```
M=2.4;
I=0.099;
b=0.05;
L=0.4;
d=0.005;
m=0.23;
g=9.81;
Ts=0.01;
[A, B, C, D]=linmod('modelnolin_taula',[0;0;0;0],0) %linialització sistema
sys=ss(A,B,C,D) % es construeix sistema
Pc=[-3.1811+3.0446i, -3.1811-3.0446i, -2.4446+0.4400i, -2.4446-0.4400i]
K=acker(A,B,Pc) % matriu K
co = ctrb (A,B) %controlabilidad del sistema:
Controlabilitat = rank (co)
ob = obsv (A,C) % Observabilidad del sistema:
Observabilitat = rank (ob)
Po=Pc*3 %Pols per al observador dues vegades mes ràpids
Ld=(place(A', C',Po))' %Matriu L per observador
acon = [A-B*K-Ld*C]
bcon=Ld
ccon=-K
dcon=[0 0]
syscon=ss(acon,bcon,ccon,dcon,0.01)
```

## Annex III. Programa de control per retorn d'estat més observador (discret)

```

M=2.4;
I=0.099;
b=0.05;
L=0.4;
d=0.005;
m=0.23;
g=9.81;
Ts=0.01;
[A, B, C, D]=linmod('modelnolin_taula',[0;0;0;0],0) %linealització sistema
sys=ss(A,B,C,D) % es construeix sistema
Pc=[-3.1811+3.0446i, -3.1811-3.0446i, -2.4446+0.4400i, -2.4446-0.4400i] % Pols
a partir de Bessel de 4 ordre *2
K=acker(A,B,Pc) % matriu K
sysd=c2d(sys,Ts) %es discretitza el sistema
[Ad Bd Cd Dd]=ssdata (sysd) %s'extreuen matrius discretes
%[Ad,Bd,Cd,Dd]=c2dm (A,B,C,D,Ts,'zoh')%discretitzat del sistema.
Pd=[0.975827253+0.0043i, 0.975827253-0.0043i, 0.968241+0.029i, 0.968241-0.029i]
%Pols per al sistema discret en el pla Z
Kd=acker(Ad, Bd, Pd) %K per sistema discret
co = ctrb (Ad,Bd) %controlabilidad del sistema:
Controlabilitat = rank (co)
ob = obsv (Ad,Cd) % Observabilidad del sistema:
Observabilitat = rank (ob)
Po=Pd/15 %Pols per al observador dues vegades mes ràpids
Ld=(place(Ad', Cd',Po))' %Matriu L per observador
acon = [Ad-Bd*Kd-Ld*Cd]
bcon=Ld
ccon=-Kd
dcon=[0 0]
syscon=ss(acon,bcon,ccon,dcon,0.01)

```

## Annex IV. Programa de control per retorn d'estat més integrador (continu)

```
M=2.4;
I=0.099;
b=0.05;
L=0.4;
d=0.005;
m=0.23;
g=9.81;
Ts=0.01;
[A, B, C, D]=linmod('modelnolin_taula',[0;0;0;0],0) %linealització sistema
sys=ss(A,B,C,D) % es construeix sistema
Pc=[-2.7792, -1.7718+2.7216i, -1.7718-2.7216i, -2.5548+1.3281i, -2.5548-1.3281i]%
Pols a partir de Bessel de 5*3
Ada=[0 1 0 0 0;zeros(4,1),A]
Bda=[0;B]
Cda=[0 1 0 0 0]
Dda=[0]
Ka=acker(Ada, Bda, Pc) %K per sistema discret
```

## Annex V. Programa de control per retorn d'estat més integrador (discret).

```
M=2.4;
I=0.099;
b=0.05;
L=0.4;
d=0.005;
m=0.23;
g=9.81;
Ts=0.01;
[A, B, C, D]=linmod('modelnolin_taula',[0;0;0;0],0) %linealització sistema
sys=ss(A,B,C,D) % es construeix sistema
Pc=[-2.7792,-1.7718+2.7216i,-1.7718-2.7216i,-2.5548+1.3281i,-2.5548-1.3281i]%
Pols a partir de Bessel de 5*3
sysd=c2d(sys,Ts) %es discretitza el sistema
[Ad Bd Cd Dd]=ssdata (sysd) %s'extreuen matrius discretes
Ada=[1 1 0 0 0;zeros(4,1),Ad]
Bda=[0;Bd]
Cda=[0 1 0 0 0]
Dda=[0]
Pa=[0.9726,0.9821+0.02673i,0.9821-0.02673i,0.97477+0.0129i,0.97477-0.0129i]
%Pols per al sistema discret en el pla Z
Ka=acker(Ada, Bda, Pa) %K per sistema discret
```