

Tecnologia de Control

Control d'un doble integrador

Alumne: Lluís Salord Quetglas

Professor: Manel Velasco

Octubre 2015

Resum

[illegible]

Índex

Resum	1
1 Introducció	9
1.1 Planta a controlar	9
1.2 Metodologia	11
2 Control de la planta	13
2.1 Control continu	13
2.2 Control discret	15
2.3 Control amb observador	16
2.4 Control amb refús de pertorbació	16
2.4.1 Pertorbació constant	16
2.4.2 Pertorbació sinusoïdal	16
2.4.3 Pertorbació constant i sinusoïdal	16
Annexos	17
A Matlab	19
A.1 Control continu	19
A.2 Control discret	19
A.3 Control amb observador	20
A.4 Control amb refús de pertorbació	21
A.4.1 Pertorbació constant	21
A.4.2 Pertorbació sinusoïdal	21
A.4.3 Pertorbació constant i sinusoïdal	21
A.5 Funció per plotejar amb format \LaTeX	21
B Arduino	23
B.1 Control continu	23
B.2 Control discret	23
B.3 Control amb observador	23

B.4	Control amb refús de pertorbació	23
B.4.1	Pertorbació constant	23
B.4.2	Pertorbació sinusoïdal	23
B.4.3	Pertorbació constant i sinusoïdal	23

Índex de figures

1.1	Esquemàtic del doble integrador instal·lat a la placa	10
2.1	Esquematic del sistema amb control continu	13
2.2	Dades de la simulació amb control continu	15

Índex de taules

1.1	Taula de connexions entre el doble integrador i ports de l'Arduino	10
-----	--	----

Capítol 1

Introducció

En l'àmbit d'aquesta assignatura es pretén aplicar els diferents coneixements de control de plantes que s'han adquirit al llarg de la carrera. Com a primera planta s'ha plantejat la d'un *DI* (doble integrador). En aquest cas, pel fet de ser un doble integrador té dos pols en el zero i per tant és per ella mateixa una planta inestable.

En aquest informe es pretén explicar els procediments que s'han seguit per dur a terme el control d'aquesta planta amb els diferents mètodes utilitzats, argumentant quines implicacions té cadascun d'aquests. El control és du a terme a través d'una placa Arduino que es connecta directament a la placa a controlar.

Els diferents mètodes que s'han utilitzat per fer el control primer s'han simulat i fet els càlculs amb *Matlab* i la *toolbox Simulink*. Tot seguit, s'ha connectat l'Arduino a l'ordinador per introduir-hi i executar el codi de control per finalment connectar la placa a controlar amb l'Arduino.

1.1 Planta a controlar

En aquest cas la planta a controlar és un placa que conté un doble integrador. El fet de tenir dos pols en el zero provoca que sigui inestable. L'esquemàtic del doble integrador és el que s'il·lustra a la figura 1.1.

Aquesta placa té diverses connexions per tal de comunicar-se amb l'exterior i que sigui controlable de diverses formes. Aquestes són, per una banda, les entrades de 5 V i l'acció de control u ; i per altra banda, els punts V_1 , V_2 i el *GND* (0 V del circuit). La relació de les connexions entre el doble integrador i l'Arduino és la que es mostra en la taula 1.1. En aquestes connexions sobretot s'ha de tenir en compte que les entrades pel *DI*, són sortides per l'Arduino i a l'inrevés. Per tant, *D6* s'ha de configurar com a sortida, a més de tipus *PWM*, i per altra banda, *A5* i *A6* com a entrades en l'Arduino.

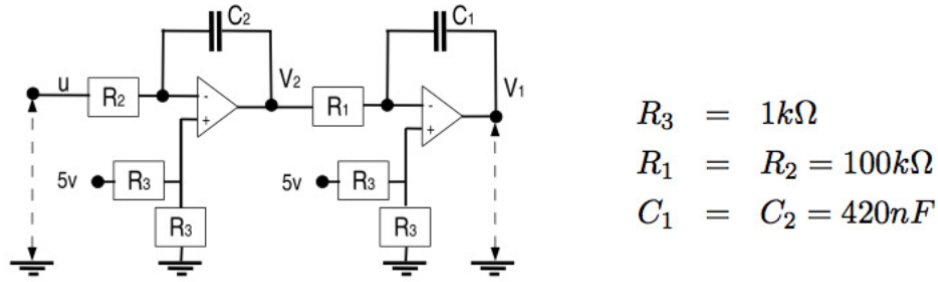


Figura 1.1: Esquemàtic del doble integrador instal·lat a la placa

Arduino	Doble Integrador
5v	5v (LM358N power)
IOREF	5v (voltage divider input)
GND	GND
A5	V_1 DI output voltage
A4	V_2 DI output voltage
D6	u (PWM)

Taula 1.1: Taula de connexions entre el doble integrador i ports de l'Arduino

Finalment, sobre el *DI* només queda comentar el model matemàtic d'aquest. Desenvolupant les equacions de la dinàmica del circuit electrònic de la figura 1.1 s'arriba a les equacions (1.1) i (1.2). A partir d'aquestes equacions i dels valors numèrics de la figura 1.1 s'arriba al model matemàtic del doble integrador que s'utilitza, expressat com (1.3), (1.4) i amb els valors de (1.5) per les matrius del sistema de la planta .

$$\dot{V}_1 = -\frac{u}{R_2 C_2} \quad (1.1)$$

$$\dot{V}_2 = -\frac{V_2}{R_1 C_1} \quad (1.2)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = A_g \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + B_g u \quad (1.3)$$

$$y = C_g \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + D_g \quad (1.4)$$

$$A_g = \begin{bmatrix} 0 & -23.8095 \\ 0 & 0 \end{bmatrix} \quad B_g = \begin{bmatrix} 0 \\ -23.8095 \end{bmatrix} \quad C_g = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad D_g = 0 \quad (1.5)$$

1.2 Metodologia

En aquesta assignatura es treballa directament amb la planta a controlar i l'Arduino que és la que porta el control d'aquesta. Ara bé, abans d'implementar el control primer s'han de fer les simulacions pertinents. A més, en haver-se incorporat el codi de control en l'Arduino, s'ha de comprovar que aquest s'està duent a terme de forma correcta. Aquesta metodologia de treball és du a terme en cadascun dels tipus de control que es duen a terme en el treball. Tot seguit s'explica en detall quines accions i programari s'utilitza per fer-ho.

El primer és dur a terme la simulació del conjunt planta-controlador. Aquesta es fa amb la *toolbox* **Simulink** havent fet els càlculs previs amb **Matlab**, tals com matrius d'alguns sistemes o valors dels controlador. En les simulacions s'extreuen les dades que es creguin convenientes per tal d'esser analitzades, tant per comprovar si el sistema es controlat teòricament amb precisió, com per esser contrastades després amb les dades reals.

En tenir el sistema simulat i amb un controlador adient s'ha de procedir a implementar aquest a través de l'Arduino. Per fer-ho s'han de transformar el conjunt de blocs de simulació a equacions i codi. Aquesta tasca és més complicada del que sembla, aquí es demostra com de clars es tenen els coneixements i és molt fàcil equivocar-se en algun punt, tot i ser un simple detall.

Finalment s'ha de comprovar el bon funcionament del control. Per dur a terme aquesta fase el primer és tenir alguna sortida de dades del sistema real. En aquest cas, s'utilitza el **Serial Port** de l'Arduino on a través de les funcions **Serial.print** i **Serial.println** esdevenen sortides del que es desitgi. Per l'estudi en qüestió s'envien les variables que intervenen en el sistema al **Serial Port** com són: les variables d'estat V_1 i V_2 , la referència, l'acció de control u i altres variables que poden ser importants segons el control que es faci. Al treure les dades pel **Serial Port**, si s'envien seguint cert patrons, poden ser llegides i plotejades *online* amb el programa **KST** per exemple, o també poder ser emmagatzemades en un fitxer per després ser analitzades amb **Matlab**. Amb les dades plotejades es pot comprovar la precisió del control o comparar-les amb les dades extretes de les simulacions.

Capítol 2

Control de la planta

La planta d'estudi es pot arribar a controlar de diverses formes, portant totes elles a l'estabilitat. Algunes porten l'estat a la referència amb més imprecisió que d'altres, com es comprova al llarg del treball, però totes elles provoquen que la planta sigui estable, que és l'objectiu del control. En aquesta part del treball és pretén explicar com s'han duit a terme els diferents controls i les característiques de cadascun d'ells.

2.1 Control continu

El control més senzill que es pot arribar a fer és considerar que el conjunt del sistema, planta-controlador, és en temps continu. En aquest cas, el sistema tan sols requereix del bloc de la planta, que utilitza els valors de les matrius de l'equació (1.5), i una realimentació d'estat K , tal que els pols de llaç tancat siguin de part real negativa o nul·la (només si es un pol simple). Aquest tipus de sistema és simula amb l'esquemàtic de la figura 2.1.

A banda del que s'acaba d'explicar, en la figura 2.1 també apareixen una referència que es pretén seguir i el bloc N_x . Aquest permet que el sistema segueixi la referència donada. Per altra banda, també s'hi hauria d'incloure un bloc N_u que multiplicaria la

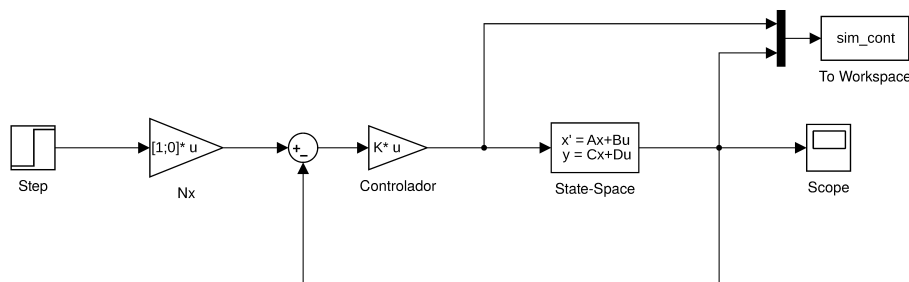


Figura 2.1: Esquemàtic del sistema amb control continu

referència i es sumaria a l'acció de control u a l'entrada de la planta, però com es veu en els càlculs de les equacions (2.2), N_u és igual a zero. El sistema (2.2) surt de desenvolupar els requisits (2.1) per tal que es segueixi la referència.

$$\begin{cases} N_x r = x_r = x_s s \\ C_r X_s s = y_r = r \end{cases} \quad (2.1)$$

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} A_g & B_g \\ C_g & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{cases} N_x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ N_u = 0 \end{cases} \quad (2.2)$$

Un cop es té el sistema montat de tal manera que aquest pugui seguir una referència, s'ha de buscar el controlador K tal que porti el sistema a error zero. Per aconseguir-ho, existeixen diverses formes depenent de si es pensen imposar uns pols de llaç tancat determinats (per assignació de pols o amb la formula d'Ackermann) o si es pretén fer un control òptim a partir d'una funció de cost determinada (control LQR). En el cas d'estudi, s'utilitza la formula d'Ackermann imposant un pols de llaç tancat amb part real negativa o nul·la (només si és un pol simple).

Els pols que s'han d'imposar com s'ha dit abans han de ser uns que siguin de part real negativa o nul·la. Ara bé, en aquest sistema s'han d'escollir dos pols, per tant, aquests no poden ser de part real nul·la ja que farien el sistema inestable. Finalment, s'han escollit els pols $-5 \pm 20i$ que compleixen el requisit i que ha sigut proposat pel professor.

Un cop escollits els pols tan sols queda aplicar la formula d'Ackermann. Aquesta dona el valor K del controlador donats els pols desitjats i les matrius A i B del sistema. Al cap i a la fi, dona el mateix resultat que l'assignació de pols on s'iguala el polinomi característic desitjat amb $|z\mathbf{I} - (\mathbf{A}_g - \mathbf{B}_g\mathbf{K})|$. La formula consisteix en l'equació (2.3) on $p_d(A_g)$ és el polinomi característic desitjat amb el valor de la matriu A_g . En el **Matlab** la funció **acker** s'encarrega de fer aquests càlculs. Si es desitja llegir el codi **Matlab** utilitzat per fer tots els càlculs es pot trobar en l'Annex A.1.

$$K = (0 \quad 0 \quad 0 \quad \dots \quad 1)(\mathbf{A}_g \quad \mathbf{A}_g\mathbf{B}_g \quad \mathbf{A}_g^2\mathbf{B}_g \quad \dots \quad \mathbf{A}_g^{n-1}\mathbf{B}_g)^{-1}p_d(A_g) \quad (2.3)$$

Al tenir tots els valors es procedeix a dur a terme les simulacions. Les dades plotejades es mostren en la figura 2.2. Aquí es pot veure com, en la simulació, s'aconsegueix estabilitzar la planta, tot i venir d'unes condicions inicials diferents a zero. Ara bé, aquesta simulació no s'adapta a la realitat, ja que l'Arduino envia l'acció de control de forma discreta i per tant el control no es correcte. Tot i que en aquest cas s'ha comprovat experimentalment que igualment es controla, tot i que no de la forma desitjada. Per tant, d'aquest tipus de control no se'n fa l'anàlisi experimental amb l'Arduino.

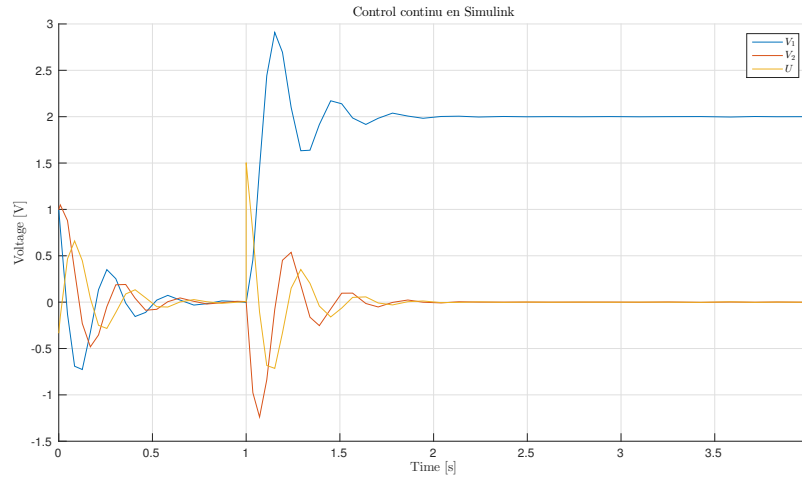


Figura 2.2: Dades de la simulació amb control continu

D'aquest cas no se'n fa un anàlisi en profunditat ja que no té especialment sentit físic. Només destacar, com es veu en la figura 2.2, el fet que es demostra que una planta inestable, amb el controlador adequat por permetre l'estabilització del sistema.

2.2 Control discret

Com s'ha explicat en el cas del control continu, aquest no reflecteix la realitat, ja que el controlador (l'Arduino) funciona de forma discreta. Aquest llegeix les dades i fa els càlculs un cop cada cert temps, i envia el mateix valor del senyal durant aquest període.

El fet que es consideri el controlador discret no s'ha de confondre en que la planta funcioni de forma discreta. La planta és continua, l'únic que passa és que l'entrada, almenys la part que prové del controlador, és discreta i per tant el controlador s'ha d'adaptar a aquest tipus de funcionament.

El primer que s'ha de fer és saber a quina freqüència treballa el controlador. En aquest cas, s'ha estipulat en l'enunciat un funcionament a $20Hz$ (període de mostreig de $0.05s$). Posteriorment, s'han de plantejar els pols de llaç tancat del sistema. Per simplificar, s'han escollit els mateixos que en el control continu, però discretitzats, que és fa aplicant l'equació (2.4) als pols continus, sent T_s el període de mostreig del controlador. Aplicant (2.4) als pols continus, $-5 \pm 20i$, de l'apartat anterior, en resulten els pols discrets $0.4208 \pm 0.6553i$. El mòdul d'aquests pols és 0.7788 es troben dins el cercle unitari, per tant, són estables.

$$P_{dis} = e^{T_s P_{cont}} \quad (2.4)$$

Per últim càlcul queda la K del controlador, que es designa al llarg de treball com

K_{dis} . Per tal d'aconseguir-ho també es pretén utilitzar la formula d'Ackermann (2.3), però no s'utilitzen les matrius A_g i B_g , sinó les seves transformacions al sistema discret amb el període de mostreig T_s . Això es pot fer a través de la funció `c2d` del `Matlab`, on donades les matrius del sistema continu i el període de mostreig T_s dona les matrius Φ i Γ del sistema discret. Finalment, s'introdueixen les matrius Φ i Γ i el període de mostreig a la funció `acker` i es té com a resultat el controlador K_{dis} .

2.3 Control amb observador

2.4 Control amb refús de pertorbació

2.4.1 Pertorbació constant

2.4.2 Pertorbació sinusoidal

2.4.3 Pertorbació constant i sinusoidal

2.5 Ús de pols reals de l'observador

Annexos

Apèndix A

Matlab

A.1 Control continu

```
%parametres model
A= [0 -23.8095;0 0];
B= [0;-23.8095];
C= [1 0];
D=[0;0];

%buscar controlador continu(realimentacio)
P=[-5+20i,-5-20i];
K=acker(A,B,P);

%busco parametres Nu i Nx
W=[A B;C 0];
N=inv(W)*[0;0;1];
Nx=N(1:2);
Nu=N(3);
```

A.2 Control discret

```
%parametres model
A= [0 -23.8095;0 0];
B= [0;-23.8095];
C= [1 0];
D=[0;0];
```

```
%pols de llaç tancat en continu
P=[-5+20i,-5-20i];

%buscar controlador discret
h=5E-2;
[phi, gam] = c2d(A,B,h);
P_dis=[exp(P(1)*h), exp(P(2)*h)];
K_dis=acker(phi,gam,P_dis);

%busco parametres Nu i Nx
W=[phi-eye(2) gam;C 0];
N=inv(W)*[0;0;1];
Nx=N(1:2);
Nu=N(3);
```

A.3 Control amb observador

```
%parametres model
A= [0 -23.8095;0 0];
B= [0;-23.8095];
C= [1 0];
D=[0;0];

%pols de llaç tancat en continu
P=[-5+20i,-5-20i];

%buscar controlador discret
h=5E-2;
[phi, gam] = c2d(A,B,h);
P_dis=[exp(P(1)*h), exp(P(2)*h)];
K_dis=acker(phi,gam,P_dis);

%busco parametres Nu i Nx
W=[phi-eye(2) gam;C 0];
N=inv(W)*[0;0;1];
Nx=N(1:2);
Nu=N(3);
```

```
%buscar controlador de l'observador amb pols més ràpids que la planta
P_obs=[-10+20i,-10-20i];
P_obs_dis=exp(P_obs.*h);
K_obs_dis=acker(phi', C', P_obs_dis);
L=K_obs_dis';
C_obs=eye(2);
```

A.4 Control amb refús de pertorbació

A.4.1 Pertorbació constant

```
%Control amb refus de perturbacions (Cas perturbacions constants)
%parametres model
A= [0 -23.8095;0 0];
B= [0;-23.8095];
C= eye(2);
D=[0;0];

%buscar controlador continu(realimentacio)
P=[-5+20i,-5-20i];
h=5E-2;
[phi, gam] = c2d(A,B,h);
P_dis=[exp(P(1)*h), exp(P(2)*h)];
K_dis=acker(phi,gam,P_dis);

phi_pert=[phi gam; 0 0 1];
gam_pert=[gam; 0];
C_pert=eye(3);

P_obs=[-10+20i,-10-20i];
P_obs_dis_pert=[exp(P_obs.*h), 0.9];
K_obs_dis_pert=acker(phi_pert', [1 0 0]', P_obs_dis_pert);
L_pert=K_obs_dis_pert';
```

A.4.2 Pertorbació sinusoidal

```
%Control amb refus de perturbacions (Cas perturbacions sinusoidals)
%parametres model
```

```

A= [0 -23.8095;0 0];
B= [0;-23.8095];
C= eye(2);
D=[0;0];
%buscar controlador discret
P=[-5+20i,-5-20i];
h=5E-2;
[phi, gam] = c2d(A,B,h);
P_dis=[exp(P(1)*h), exp(P(2)*h)];
K_dis=acker(phi,gam,P_dis);

w=2.4*2*pi();
phi_sin=c2d([0 1;-w*w 0],[1 0]',h);
phi_pert=[phi gam [0;0]; zeros(2) phi_sin];
gam_pert=[gam; 0; 0];
C_pert=eye(4);

P_obs=[-10+20i,-10-20i];
P_obs_dis_pert=[exp(P_obs.*h), 0.3, 0.3];
K_obs_dis_pert=acker(phi_pert', [1 0 0 0]', P_obs_dis_pert);
L_pert=K_obs_dis_pert';

```

A.4.3 Pertorbació constant i sinusoidal

```

%Control amb refus de perturbacions (Cas perturbacions sinusoidals i constants)
%parametres model
A= [0 -23.8095;0 0];
B= [0;-23.8095];
C= eye(2);
D=[0;0];

%buscar controlador discret
P=[-5+20i,-5-20i];
h=5E-2;
[phi, gam] = c2d(A,B,h);
P_dis=[exp(P(1)*h), exp(P(2)*h)];
K_dis=acker(phi,gam,P_dis);

w=2.4*2*pi();

```



```

phi_sin=c2d([0 1;-w*w 0],[1 0]',h);
phi_pert=[phi gam gam [0;0]; zeros(3,2) [1;0;0] [[0 0]; phi_sin]];
gam_pert=[gam; 0; 0; 0];
C_pert=eye(5);

P_obs=[-10+20i,-10-20i];
P_obs_dis_pert=[exp(P_obs.*h), 0.4, 0.4, 0.9];
K_obs_dis_pert=acker(phi_pert', [1 0 0 0 0]', P_obs_dis_pert);
L_pert=K_obs_dis_pert';

```

A.5 Ús de pols reals de l'observador

```

%Control amb refus de perturbacions (Cas perturbacions sinusoidals i constants)
%parametres model
A= [0 -23.8095;0 0];
B= [0;-23.8095];
C= eye(2);
D=[0;0];

%buscar controlador discret
P=[-5+20i,-5-20i];
h=5E-2;
[phi, gam] = c2d(A,B,h);
P_dis=[exp(P(1)*h), exp(P(2)*h)];
K_dis=acker(phi,gam,P_dis);

w=0.5371*2*pi();
phi_sin=c2d([0 1;-w*w 0],[1 0]',h);
phi_pert=[phi gam gam [0;0]; zeros(3,2) [1;0;0] [[0 0]; phi_sin]];
gam_pert=[gam; 0; 0; 0];
C_pert=eye(5);

P_obs=[-10,-5];
P_obs_dis_pert=[exp(P_obs.*h), 0.9, 0.9, 0.9];
K_obs_dis_pert=acker(phi_pert', [1 0 0 0 0]', P_obs_dis_pert);
L_pert=K_obs_dis_pert';

```

A.6 Funció per plotejar amb format \LaTeX

```
function plot_control(X,Y,title_plot,names, x_min, x_max, y_min, y_max)

set(0,'DefaultTextInterpreter', 'latex')

fontsize = 12;
set(0,'defaultaxesfontsize',fontsize);
set(0,'defaulttextfontsize',fontsize);

figure
hold on;

for i=1:length(Y(1,:))
plot(X,Y(:,i),'DisplayName',names(i));
end

title(title_plot);
xlabel('Time [s]', 'Interpreter', 'latex');
ylabel('Voltage [V]', 'Interpreter', 'latex');

axis([x_min x_max y_min y_max]);

h=legend(names);
set(h,'Interpreter', 'latex');

grid on;

end
```

Apèndix B

Arduino

B.1 Control continu

B.2 Control discret

B.3 Control amb observador

B.4 Control amb refús de pertorbació

B.4.1 Pertorbació constant

B.4.2 Pertorbació sinusoïdal

B.4.3 Pertorbació constant i sinusoïdal