Rapid Controller Prototyping of the Inverted Pendulum Using Scilab/Scicos, Comedi Drivers and Rtai-Lab

Ing. Roberto Bucher
University of Applied Sciences of Southern Switzerland
CH-6928 Manno
roberto.bucher@supsi.ch

May 15, 2005

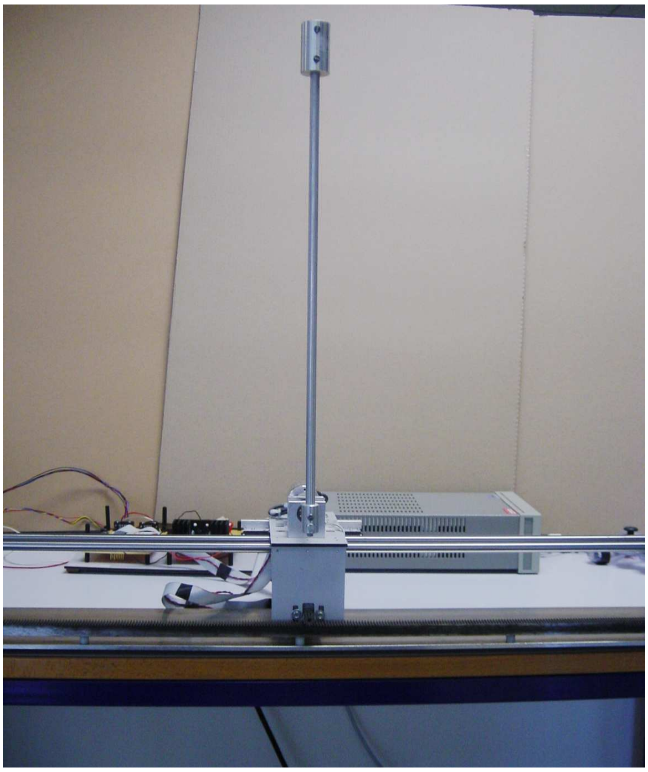

Figure 1: Running system

1 Pendulum model

1.1 Nonlinear model

The nonlinear model of the inverted pendulum can be expressed in the form

$$\begin{aligned}
\dot{\varphi} &= \omega \\
\dot{\omega} &= m \cdot r \cdot \frac{(M+m)\cdot g \cdot \sin\varphi - m \cdot r \cdot \dot{\varphi}^2 \cdot \sin\varphi \cdot \cos\varphi + (F - d\cdot v)\cdot\cos\varphi}{\Theta\cdot(M+m) - m^2\cdot r^2\cdot\cos^2\varphi} \\
\dot{x} &= v \\
\dot{v} &= \frac{m^2\cdot r^2\cdot g\cdot\sin\varphi\cdot\cos\varphi - \Theta\cdot m\cdot r\cdot\dot{\varphi}^2\cdot\sin\varphi + \Theta\cdot(F - d\cdot v)}{\Theta\cdot(M+m) - m^2\cdot r^2\cdot\cos^2\varphi}
\end{aligned} \tag{1}$$

1.2 Linear model

The linearized model of the inverted pendulum on the inverted unstable equilibrium point is described by

$$\begin{bmatrix} \delta\dot{\varphi} \\ \delta\dot{\omega} \\ \delta\dot{x} \\ \delta\dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{m\cdot r\cdot(M+m)\cdot g}{\Theta\cdot(M+m)-m^2\cdot r^2} & 0 & 0 & \frac{-m\cdot r\cdot d}{\Theta\cdot(M+m)-m^2\cdot r^2} \\ 0 & 0 & 0 & 1 \\ \frac{m^2\cdot r^2\cdot g}{\Theta\cdot(M+m)-m^2\cdot r^2} & 0 & 0 & \frac{-\Theta\cdot d}{\Theta\cdot(M+m)-m^2\cdot r^2} \end{bmatrix} \cdot \begin{bmatrix} \delta\varphi \\ \delta\omega \\ \delta x \\ \delta v \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{m\cdot r}{\Theta\cdot(M+m)-m^2\cdot r^2} \\ 0 \\ \frac{\Theta}{\Theta\cdot(M+m)-m^2\cdot r^2} \end{bmatrix} \cdot \delta F \tag{2}$$

where
M is the mass of the cart
m is the mass of the pole
r is the fictive length of the pole (distance of center of mass from the joint)
Θ is the Inertial moment of the pole
d is the viscous friction coefficient

2 Substituting the force

The force δF is given by the torque T of the motor which drives the cart with the pole.
In this case, the force δF can be calculated from

$$\delta F = \frac{\delta T}{r} = \frac{K_t \delta I_a}{r} = \frac{K_t}{r}\frac{\delta U - K_b \delta\omega}{R_a} = -\frac{K_t K_b}{R_a r^2}\delta v + \frac{K_t}{R_a r}\delta U \tag{3}$$

If we substitute equation (3) in (2) we obtain

$$\begin{bmatrix} \delta\dot{\varphi} \\ \delta\dot{\omega} \\ \delta\dot{x} \\ \delta\dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{m\cdot r\cdot(M+m)\cdot g}{\Theta\cdot(M+m)-m^2\cdot r^2} & 0 & 0 & \frac{-m\cdot r}{\Theta\cdot(M+m)-m^2\cdot r^2}\cdot(\frac{K_t K_b}{R_a r^2} + d) \\ 0 & 0 & 0 & 1 \\ \frac{m^2\cdot r^2\cdot g}{\Theta\cdot(M+m)-m^2\cdot r^2} & 0 & 0 & \frac{-\Theta}{\Theta\cdot(M+m)-m^2\cdot r^2}\cdot(\frac{K_t K_b}{R_a r^2} + d) \end{bmatrix} \cdot \begin{bmatrix} \delta\varphi \\ \delta\omega \\ \delta x \\ \delta v \end{bmatrix} +$$

$$
+ \begin{bmatrix} 0 \\ \frac{m \cdot r}{\Theta \cdot (M+m) - m^2 \cdot r^2} \frac{K_t}{R_a r} \\ 0 \\ \frac{\Theta}{\Theta \cdot (M+m) - m^2 \cdot r^2} \frac{K_t}{R_a r} \end{bmatrix} \cdot \delta U \tag{4}
$$

3 The real plant

Figure 2 shows the real inverted pendulum with the I/O modules.

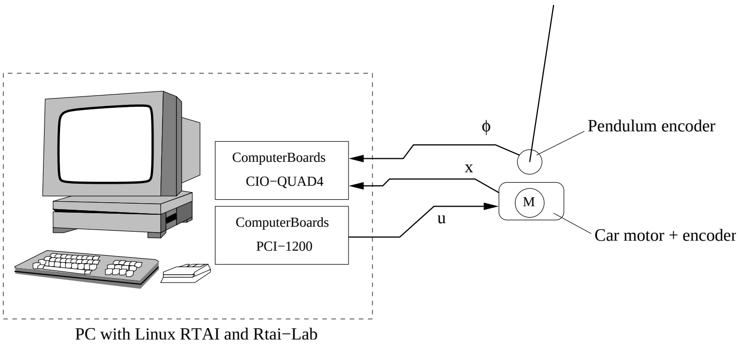

Figure 2: I/O scheme of the inverted pendulum

The PC generates the control signal and give it to the plant through a ComputerBoards DA Card (PCI-1200). The pole and cart positions are read using an encoder card (ComputerBoards CIO-QUAD4).

4 Cart model

Now we analyze only the cart and the motor without the pole. The differential equation that describes this system is

$$
M\ddot{x} = F - d\dot{x} \tag{5}
$$

where F is given by the motor and is the same found in equation (3).
We now calculate the transfer function between $U(s)$ and $X(s)$ and we find

$$
\frac{X(s)}{U(s)} = \frac{\frac{K_t}{MR_a r}}{s\left(s + \frac{K_t K_b}{MR_a r^2} + \frac{d}{M}\right)} = \frac{K}{s(s+\alpha)} \tag{6}
$$

where

$$
K = \frac{K_t}{MR_a r}
$$

and

$$
\alpha = \left(\frac{K_t K_b}{R_a r^2} + d\right)\frac{1}{M}
$$

5 Parametrical identification

The inverse Laplace Transform of $y(t)$ from 6 with an input step is

$$y(t) = -\frac{K}{\alpha^2} + \frac{K}{\alpha}t + \frac{K}{\alpha^2}e^{\alpha t} \tag{7}$$

The response becomes a straight line for $t \to \infty$

$$\lim_{t\to\infty} y(t) = -\frac{K}{\alpha^2} + \frac{K}{\alpha}t \tag{8}$$

The response data of the real cart can be acquired using the scheme of the Figure 3.

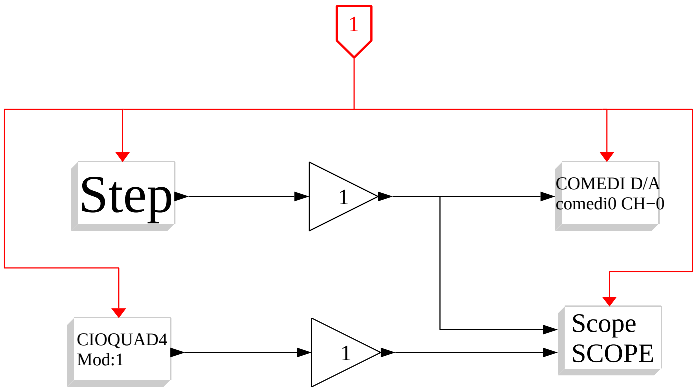

Figure 3: Superblock for the "Idntification" task

After getting a step response of the cart system (see figure 4), we can select the straight line part of this response and use it to extract the values of K and α.

```
dati2=read('Scope_45.dat',-1,2);
t=dati2(:,1);
y=dati2(:,2);
xbasc()
plot2d(t,y)
pt=locate(2);
n1=round(pt(1,1)/0.001);
n2=round(pt(1,2)/0.001);
uin=4.5;
y=y/uin;
tt=t(n1:n2,1);
yy=y(n1:n2,1);

n=size(tt,1);
u=ones(n,1);
aa=[tt,u];
x=aa\yy;
a=x(1,1);
b=x(2,1);
alfa=-a/b;
K=a*alfa;
```

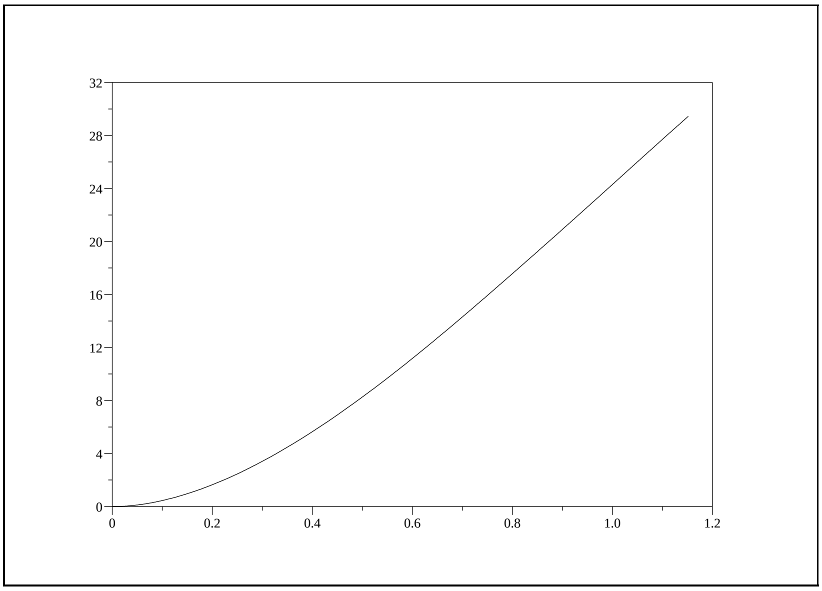

Figure 4: Step response of the cart system

6 Inserting the motor parameter in the pendulum model

Now we substitute the motor parameters with the values of K and α found before. The model of the inverted pendulum is now

$$\begin{bmatrix} \delta\dot{\varphi} \\ \delta\dot{\omega} \\ \delta\dot{x} \\ \delta\dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{m\cdot r\cdot(M+m)\cdot g}{\Theta\cdot(M+m)-m^2\cdot r^2} & 0 & 0 & \frac{-m\cdot r}{\Theta\cdot(M+m)-m^2\cdot r^2}\alpha\cdot M \\ 0 & 0 & 0 & 1 \\ \frac{m^2\cdot r^2\cdot g}{\Theta\cdot(M+m)-m^2\cdot r^2} & 0 & 0 & \frac{-\Theta}{\Theta\cdot(M+m)-m^2\cdot r^2}\alpha\cdot M \end{bmatrix} \cdot \begin{bmatrix} \delta\varphi \\ \delta\omega \\ \delta x \\ \delta v \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ \frac{m\cdot r}{\Theta\cdot(M+m)-m^2\cdot r^2}K\cdot M \\ 0 \\ \frac{\Theta}{\Theta\cdot(M+m)-m^2\cdot r^2}K\cdot M \end{bmatrix} \cdot \delta U \tag{9}$$

7 Control of the inverted pendulum

Figure 5 shows the controlled system. The state feedback gains can be calculated using a LQR approach. The states can be obtained using a reduced order observer.
Starting from the reduced order observer transfer function
The controller can be implemented as

$$u = -K_{lqr}x \tag{10}$$

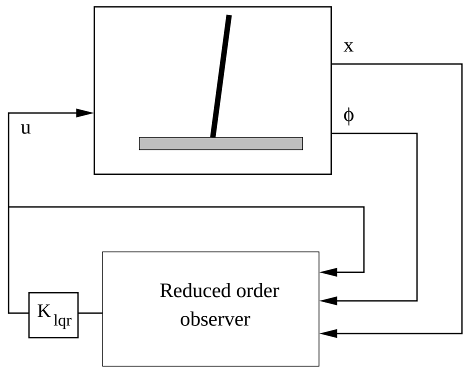

Figure 5: Control scheme of the inverted pendulum

The reduced order transfer function is given by

$$x = G_u u + G_y y \tag{11}$$

By inserting 11 into 10 we obtain

$$u = \frac{-K_{lqr} G_y}{1 + K G_u} y \tag{12}$$

We calculate the controller of the inverted pendulum with the following scilab script

```
// Inverted pendulum!

Ts=1e-3;

// Variabili del sistema

M=1.316;            // cart mass
g=9.81;
m=0.10632+0.08555;  // Pole mass
r=331.33e-3;        // Pole length
J=m*r^2;            // Inertial
alfa=3.656;         // alfa=kt*kb/(M*ra*r^2)+d/M
kangle=1;           // angle constant (conversion factor)
kx=123.6374/0.75;   // position constant [rad/m] (onversion factor)
u0=1.5;
k=122.317/kx;       // k=kt/(ra*rr*M)

// Linearized system

den=J*(M+m)-m^2*r^2;

a=[0,1,0,0;(m*r*(m+M)*g)/den,0,0,-m*r/den*alfa*M;0,0,0,1;(m^2*r^2*g)/den,0,0,-J/den*alfa*M];
b=[0;m*r/den*k*M;0;J/den*k*M];
c=[kangle,0,0,0;0,0,kx,0];
d=[0;0];

sys=syslin('c',a,b,c,d);

sysd=dscr(sys,Ts);
[ad,bd,cd,dd]=abcd(sysd);

// State feedback controller [LQR]

// weights

Q=diag([10000,1,10000,1]);                      // 4 by 4
R=[1];                                          // 1 by 1
```

```
k_lqr=bb_dlqr(ad,bd,Q,R);

E=spec(ad);

plqr=abs(real(log(E)/Ts))';
pmax=max(plqr);

k_lqr=-k_lqr;

preg=spec(a);

// Reduced order observer

poli_oss=exp([real(preg(3)),real(preg(4))]*10*Ts);
T=[0,0,0,1;0,1,0,0];
[ao,bo,co,do]=redobs(ad,bd,cd,dd,T,poli_oss);

Greg=comp_form(ao,bo,co,do,Ts,k_lqr);

[g1n,g1d]=tfdata(Greg(:,2));
[g2n,g2d]=tfdata(Greg(:,3));

function [A_redobs,B_redobs,C_redobs,D_redobs]=redobs(A,B,C,D,T,poles)
P=[C;T]
invP=inv([C;T])

AA=P*A*invP

ny=size(C,1)
nx=size(A,1)
nu=size(B,2)

A11=AA(1:ny,1:ny)
A12=AA(1:ny,ny+1:nx)
A21=AA(ny+1:nx,1:ny)
A22=AA(ny+1:nx,ny+1:nx)

L1=ppol(A22',A12',poles)';

nn=nx-ny;

A_redobs=[-L1 eye(nn,nn)]*P*A*invP*[zeros(ny,nn); eye(nn,nn)];
B_redobs=[-L1 eye(nn,nn)]*[P*B P*A*invP*[eye(ny,ny);L1]]*[eye(nu,nu) zeros(nu,ny); -D, eye(ny,ny)];
C_redobs=invP*[zeros(ny,nx-ny);eye(nn,nn)];
D_redobs=invP*[zeros(ny,nu) eye(ny,ny);zeros(nx-ny,nu) L1]*[eye(nu,nu) zeros(nu,ny); -D, eye(ny,ny)];

function [Gu,Gy]=get_gu_gy(G)
Gu=G('num')(:,1)./G('den')(:,1);
Gy=G('num')(:,2:$)./G('den')(:,2:$);

function [Greg]=comp_form(A,B,C,D,Ts,K)

ss_sys=syslin('d',A,B,C,D);
ss_sys(7)=Ts;
g_sys=ss2tf(ss_sys);

[gu,gy]=get_gu_gy(g_sys);

Greg=[1/(1+K*gu),-K*gy/(1+K*gu)];

function [num,den]=tfdata(G)
num=G('num');
den=G('den');
```

8 Implementation

Figure 6 shows the superblock of the controller.

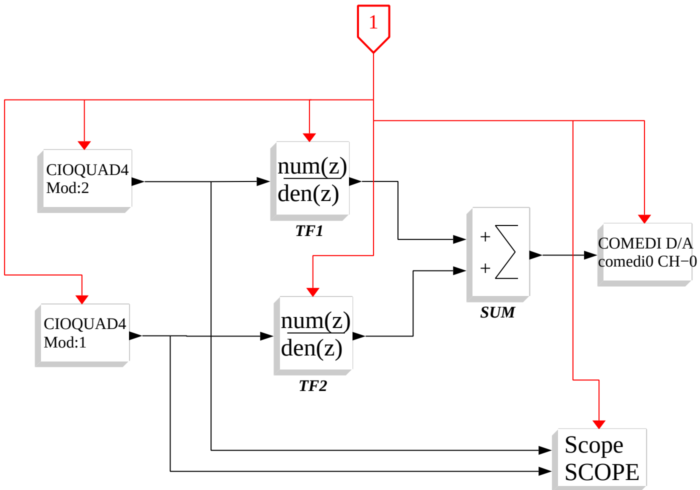

Figure 6: Super block of the controller