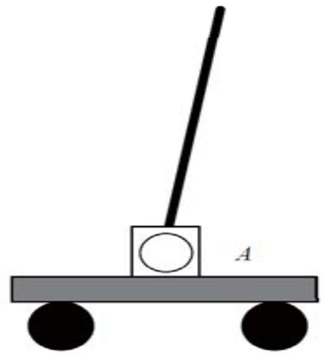
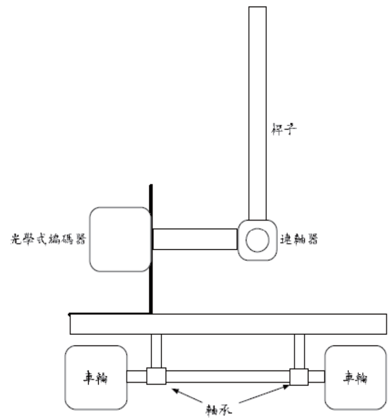
Inverted Pendulum Car

電機工程學系 鄭智湧

正面側面示意圖

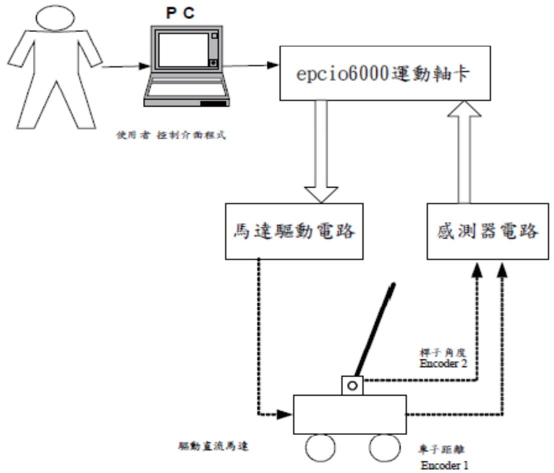




車與桿倒單擺系統機構

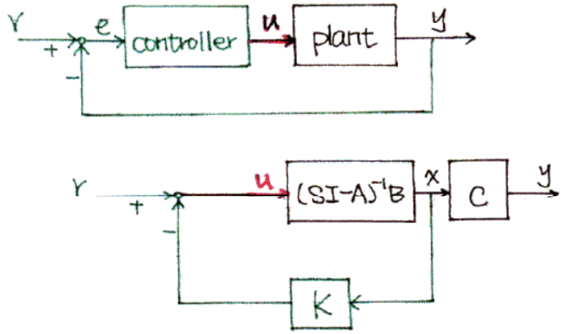
系統的機械結構由直流馬達、齒輪、車輪、編碼器組成車子的部分; 角度編碼器、桿子組合成倒單擺部份

系統結構圖



State Feedback Controller Design

State Feedback Design



$\dot{x} = Ax + Bu$

$y = Cx$

$u = -Kx + r$ state feedback control law

Eigenvalue (pole) Placement

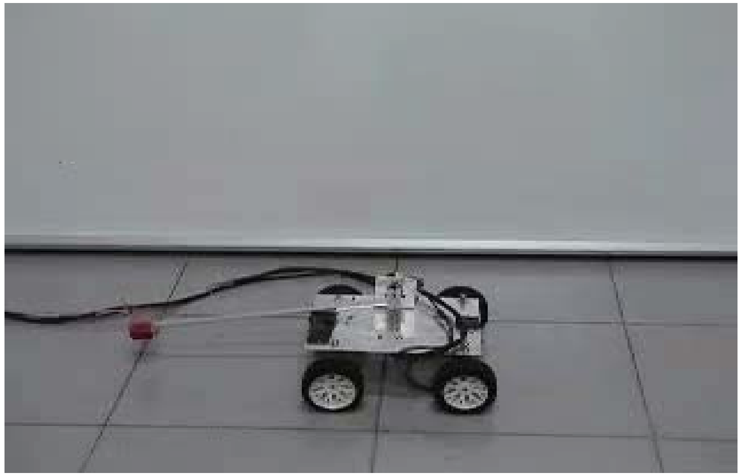
Thm :

Consider the system $\dot{x} = Ax + bu$ with the state feedback $u = -Kx$

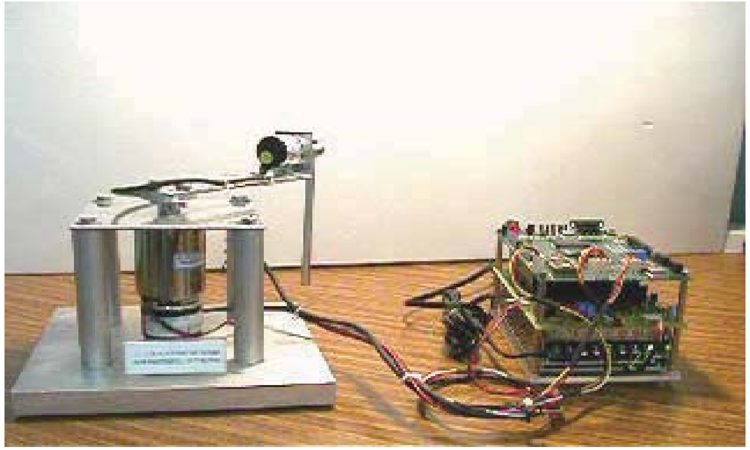
If the system is controllable then eigenvalues of $A - bk$ can be arbitrarily moved around the complex plane (with complex conjugate pairs)

```
interrupt void eva_timer1_isr(void)
{

    QEP1 = EvaRegs.T2CNT;              // counting value of pendulum encoder
    QEP2 = EvbRegs.T3CNT;              // counting value of car encoder

    arm = QEP1*0.0031416;      // theta=QEP2*(2*pi/2000) rad
        arm_dot = ((arm-arm_old)/0.001);           // sampling time=0.001s
        arm_old = arm;

    pend = QEP2*0.0031416;     // thetb=QEP3*(2*pi/2000) rad
        pend_dot = ((pend-pend_old)/0.001);        // sampling time=0.001s
        pend_old = pend;

        U = balance_LQR();



 // state feedback control law for balancing control
 float balance_LQR(void)
 {
  float law;
  law = -((k1*arm)+(k2*pend)+(k3*arm_dot)+(k4*pend_dot));
  return law;
 }
```

Video Demo 1



Video Demo 2



Simulation Study

IP Modeling

For pendulum,

$$J\ddot{\theta} = F_y l \sin\theta - F_x l \cos\theta$$

where

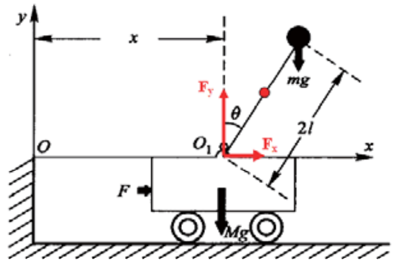
$$F_x = m\frac{d^2}{dt^2}(x + l\sin\theta)$$

and

$$F_y - mg = m\frac{d^2}{dt^2}(l\cos\theta)$$

For cart,

$$F - F_x = M\frac{d^2x}{dt^2}$$



$$(J + ml^2)\ddot{\theta} + ml\cos\theta \cdot \ddot{x} = mlg\sin\theta$$

$$(M + m)\ddot{x} + ml(\cos\theta \cdot \ddot{\theta} - \sin\theta \cdot \dot{\theta}^2) = F$$

Note: Lagrange dynamics is another approach for modeling.

$$\ddot{\theta} = \frac{ml\cos\theta \cdot F + m^2l^2\sin\theta\cos\theta \cdot \dot{\theta}^2 - (M+m)mlg\sin\theta}{m^2l^2\cos^2\theta - (J+ml^2)(M+m)}$$

$$\ddot{x} = \frac{(J+ml^2)F + lm(J+ml^2)\sin\theta \cdot \dot{\theta}^2 - m^2l^2g\sin\theta\cos\theta}{(J+ml^2)(M+m) - m^2l^2\cos^2\theta}$$

Linearization $\Longrightarrow$ $\dot{\theta}^2 \approx 0, \sin\theta \approx \theta, \cos\theta \approx 1$

$$\ddot{\theta} = \frac{(M+m)mlg\theta - mlF}{J(M+m) + mMl^2}$$

$$\ddot{x} = \frac{(J+ml^2)F - m^2l^2g\theta}{J(M+m) + mMl^2}$$

Assume $M = 1,\ m = 1,\ 2l = 0.6,\ g = 10,\ J = \dfrac{ml^2}{3} = 0.03$

Model (NL)

$$\ddot{\theta} = \frac{0.3\cos\theta \cdot F + 0.09\sin\theta \cdot \cos\theta \cdot \dot{\theta}^2 - 6\sin\theta}{0.09\cos^2\theta - 0.24}$$

$$\ddot{x} = \frac{0.12F + 0.036\sin\theta \cdot \dot{\theta}^2 - 0.9\sin\theta \cdot \cos\theta}{0.24 - 0.09\cos^2\theta}$$

Model (L)

$$\ddot{\theta} = 40\theta - 2F$$

$$\ddot{x} = -6\theta + 0.8F$$

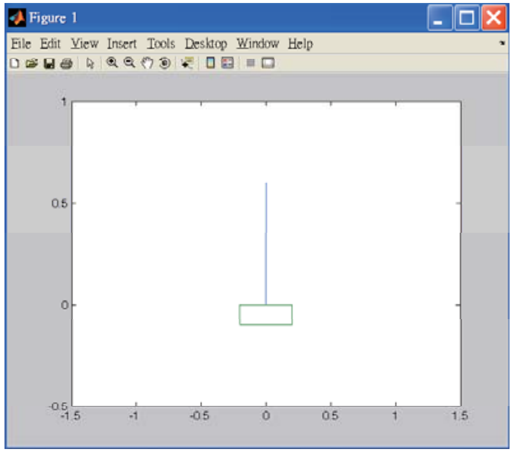
State space model

$$x_1 = \theta,\ x_2 = \dot{\theta},\ x_3 = x,\ x_4 = \dot{x}$$

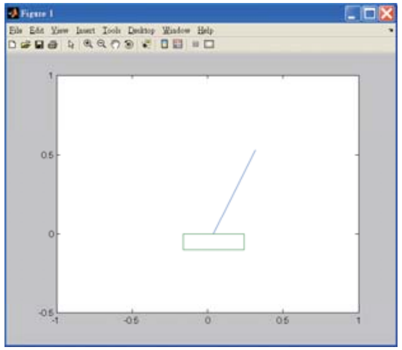
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 40 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -6 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -2 \\ 0 \\ 0.8 \end{bmatrix} F$$

Check controllability (under-actuated or non-holonomic)

IP graphics test



test_IP_gra.m




```
■ length=0.6; len=0.2; width=0.1;
■ x_pos=0; angle=pi/3;
■ N=100;
■ for i=1:N
■     x_pos=x_pos+1/N;
■     angle=angle+1/N;
■     stick_x=[x_pos, x_pos+length*cos(angle)];
■     stick_y=[  0  ,    length*sin(angle)   ];
■     car_x=[x_pos-len x_pos+len x_pos+len x_pos-len x_pos-len];
■     car_y=[    0         0       -width    -width       0   ];
■     plot(stick_x,stick_y,car_x,car_y); axis([-1 1 -0.5 1]);
■     MA(:,i)=getframe;
■ end
```

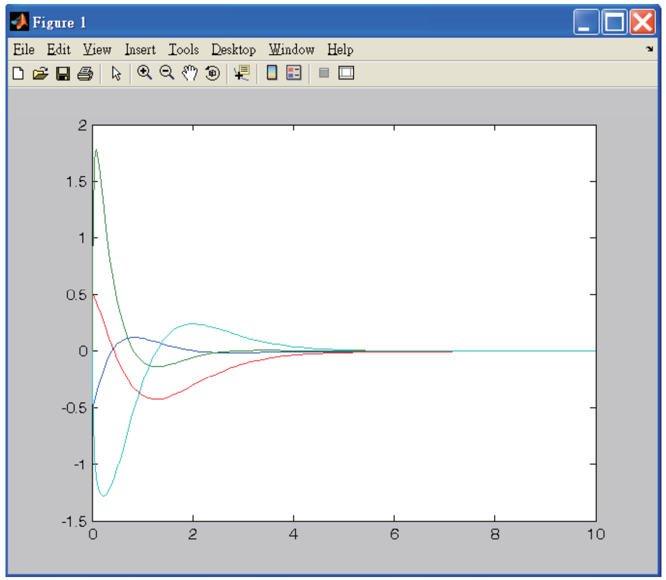
ODE simulation

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 40 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -6 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ -2 \\ 0 \\ 0.8 \end{bmatrix} u$$

```
function dx=IP(t,x)

% state feedback design
u =-[-95.4861 -22.3475 -7.0711 -12.5201]*x;
dx= [ 0 1 0 0
      40 0 0 0
       0 0 0 1
      -6 0 0 0]*x+[0 -2 0 0.8]'*u;
```

IPcar.m

```
■ length=0.6; len=0.2; width=0.1;
■ [t,x]=ode23(@IP,[0 10],[-0.5 0 0.5 0]);
■ N=max(size(x));
■ for i=1:N
■     angle=pi/2-x(i,1);
■     x_pos=x(i,3);
■     stick_x=[x_pos, x_pos+length*cos(angle)];
■     stick_y=[  0  ,    length*sin(angle)   ];
■     car_x=[x_pos-len x_pos+len x_pos+len x_pos-len x_pos-len];
■     car_y=[    0         0       -width    -width       0   ];
■     plot(stick_x,stick_y,car_x,car_y); axis([-1.5 1.5 -0.5 1]);
■     MA(:,i)=getframe;
■ end
```



Acker in Matlab

% Use the example discussed in class.

% The desired eigenvalues are at -2, -1, -1

```
A=[1 0 0; -1 0 2; 0 -1 1]; B=[1;0;0];
J=[-2 -1 -1];
K=acker(A,B,J)
```

% The result is K = [6 -8 0]

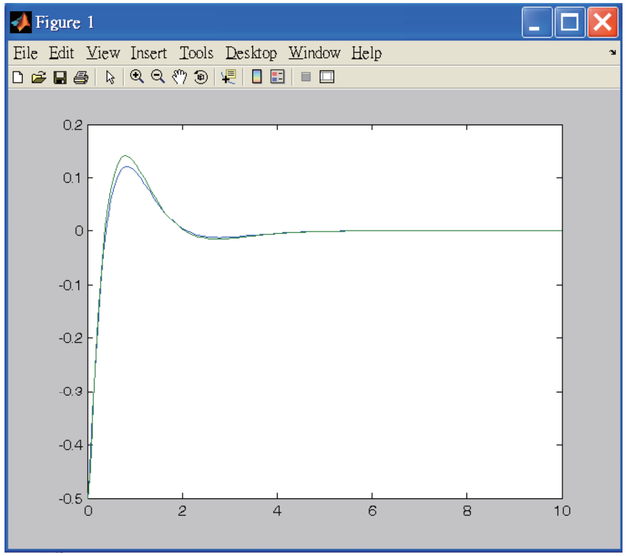
IPcar_NL and IP_NL

- function dx=IP_NL(t,x)

- % LQR design
- u=-[-95.4861 -22.3475 -7.0711 -12.5201]*x;

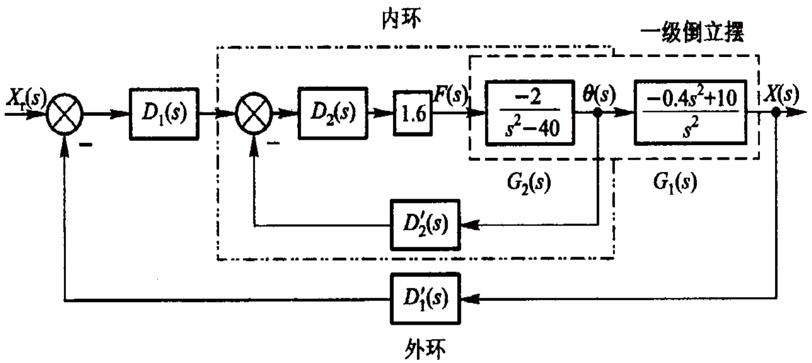
- dx=zeros(4,1);
- dx(1)=x(2);
- dx(2)=(0.3*cos(x(1))*u+0.09*sin(x(1))*cos(x(1))*x(1)^2-6*sin(x(1))) /(0.09*cos(x(1))^2-0.24);
- dx(3)=x(4);
- dx(4)=(0.12*u+0.036*sin(x(1))*x(1)^2-0.9*sin(x(1))*cos(x(1)))/(0.24-0.09*cos(x(1))^2);

Comparion (NL could diverge)



PID control design (linear simulation)

PID control of angle and position



$D_2(s) = -20$ $\qquad$ $D_1(s) = 0.12s + 0.12$

$D_2'(s) = 0.175\,s + 1.625$ $\qquad$ $D_1'(s) = 1$

IPpid.m

- G2= tf([-2],[1 0 -40]);
- D2= tf([-32],[1]);
- D2p= tf([0.175 1.625],[1]);
- sys_inner= feedback(series(D2,G2),D2p);

- G1= tf([-0.4 0 10],[1 0 0]);
- D1= tf([0.12 0.12],[1]);
- D1p= tf([1],[1]);
- sys_outer= feedback(series(series(D1,sys_inner),G1),D1p);
- sys_angle= feedback(series(D1,sys_inner),series(G1,D1p));

```
t=0:0.1:10;
pos=0.1*impulse(sys_outer,t);
the=0.1*impulse(sys_angle,t);
% plot(t,the,t,pos)

length=0.6; len=0.2; width=0.1;
N=max(size(the));
for i=1:N
    angle=pi/2-the(i);
    x_pos=pos(i);
    stick_x=[x_pos, x_pos+length*cos(angle)];
    stick_y=[  0  ,    length*sin(angle)  ];
    car_x=[x_pos-len x_pos+len x_pos+len x_pos-len x_pos-len];
    car_y=[    0         0       -width    -width       0   ];
    plot(stick_x,stick_y,car_x,car_y); axis([-1.5 1.5 -0.5 1]);
    MA(:,i)=getframe;
end
```

Pendulum angle and cart position