State Feedback and Observers

Real-Time Systems, Lecture 9

Karl-Erik Årzén

February 2017

Lund University, Department of Automatic Control

Lecture 9

[IFAC PB Chapter 8]

- State feedback
- Observers
- Integral action and disturbance estimation

Control Design

Many factors to consider, including:

- Attenuation of load disturbances
- Reduction of the effect of measurement noise
- Command signal following
- Variations and uncertainties in process behavior

Two Classes of Control Problems

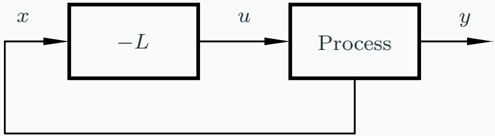
Regulation problems: compromise between rejection of load disturbances and injection of measurement noise

- Feedback
- Lecture 9

Servo problems: make the output respond to command signals in the desired way

- Feedforward
- Lecture 10

State Feedback: Problem Formulation



- Discrete-time process model

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$

- Linear feedback from all state variables

$$u(k) = -Lx(k)$$

- Disturbances modelled by nonzero initial state $x(0) = x_0$
- Goal: Control the state to the origin, using a reasonable control signal

Closed-Loop System

The state equation

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$

with the control law

$$u(k) = -Lx(k)$$

gives the closed-loop system

$$x(k+1) = (\Phi - \Gamma L)\, x(k)$$

Pole placement design: Choose L to obtain the desired characteristic equation

$$\det(zI - \Phi + \Gamma L) = 0$$

(Matlab: `place` or `acker`)

State feedback only modifies the poles, not the zeros

Example – Double Integrator

$$x(k+1) = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix} x(k) + \begin{pmatrix} h^2/2 \\ h \end{pmatrix} u(k)$$

Linear state-feedback controller

$$u(k) = -Lx(k) = -l_1 x_1(k) - l_2 x_2(k)$$

The closed-loop system becomes

$$\begin{aligned} x(k+1) &= (\Phi - \Gamma L)x(k) \\ &= \begin{pmatrix} 1 - l_1 h^2/2 & h - l_2 h^2/2 \\ -l_1 h & 1 - l_2 h \end{pmatrix} x(k) \end{aligned}$$

Characteristic equation

$$z^2 + \left(\frac{l_1 h^2}{2} + l_2 h - 2\right) z + \left(\frac{l_1 h^2}{2} - l_2 h + 1\right) = 0$$

Example Cont'd

Characteristic equation

$$z^2 + \left(\frac{l_1 h^2}{2} + l_2 h - 2\right) z + \left(\frac{l_1 h^2}{2} - l_2 h + 1\right) = 0$$

Assume desired characteristic equation $z^2 + a_1 z + a_2 = 0$.

Linear equations for l_1 and l_2

$$\frac{l_1 h^2}{2} + l_2 h - 2 = a_1 \qquad \frac{l_1 h^2}{2} - l_2 h + 1 = a_2$$

Solution:

$$l_1 = \frac{1}{h^2}\left(1 + a_1 + a_2\right)$$
$$l_2 = \frac{1}{2h}\left(3 + a_1 - a_2\right)$$

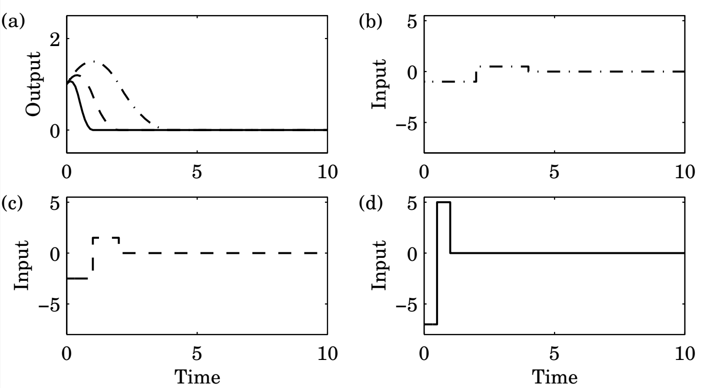
- L depends on h

Deadbeat Control — Only in Discrete Time

Choose $P(z) = z^n \Rightarrow\ h$ only remaining design parameter

Drives all states to zero in at most n steps after an impulse disturbance in the states (can be very aggressive for small h!)

Example: Double integrator, $x_0^T = [\,1 \quad 1\,]$, (b) $h = 2$ (dash-dotted), (c) $h = 1$ (dashed), (d) $h = 0.5$ (solid)



Reachability

The eigenvalues of $\Phi - \Gamma L$ can be placed arbitrarily if and only if the system is *reachable*, i.e. if the reachability matrix

$$W_c = \begin{pmatrix} \Gamma & \Phi\Gamma & \dots & \Phi^{n-1}\Gamma \end{pmatrix}$$

has full rank.

If $\text{rank}(W_C) = n_c < n$ then the system has an unreachable subsystem and its $n - n_c$ eigenvalues cannot be changed by state feedback. However, if the system is stabilizable the closed loop system will still be stable

In practice, also when the system is reachable, moving some eigenvalues could require high gain and lead to bad controllers.

State Feedback in Reachable Form

Convert the system to reachable canonical form:

$$x(k+1) = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_n \\ 1 & 0 & \dots & 0 \\ & \ddots & & \vdots \\ & & 1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(k)$$

In this case, application of the state feedback

$$u = -l_1 x_1 - \dots - l_n x_n$$

changes the coefficients $a_1, \dots, a_n$ to $a_1 + l_1, \dots, a_n + l_n$, so the characteristic polynomial changes to

$$z^n + (a_1 + l_1) z^{n-1} + \dots + (a_{n-1} + l_{n-1}) z + a_n + l_n$$

Design method: Transform to reachable canonical form, apply state feedback, transform the controller back again – Ackermann's formula (see IFAC PB)

State Feedback with Integral Action

Integral action can be introduced by augmenting the plant model with an extra state variable, x_i, that integrates the plant output:

$$x_i(k+1) = x_i(k) + y(k) = x_i(k) + Cx(k)$$

The augmented open-loop system becomes

$$\begin{pmatrix} x(k+1) \\ x_i(k+1) \end{pmatrix} = \begin{pmatrix} \Phi & 0 \\ C & I \end{pmatrix} \begin{pmatrix} x(k) \\ x_i(k) \end{pmatrix} + \begin{pmatrix} \Gamma \\ 0 \end{pmatrix} u(k)$$

We can then design a state feedback controller

$$u(k) = -\begin{pmatrix} L & L_i \end{pmatrix} \begin{pmatrix} x(k) \\ x_i(k) \end{pmatrix}$$

using the same techniques as before

(Integral action can also be introduced using a disturbance observer, as we will see later)

Reconstruction

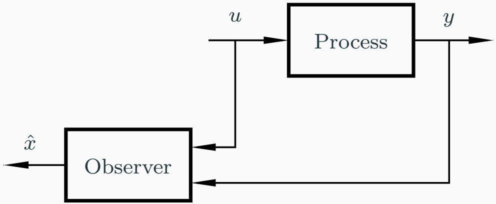
What to do if we cannot measure the full state vector or if we have noisy measurements?

Reconstruction Through Direct Calculations

Basic idea: Reconstruct the state vector $x(k)$ through direct calculations using the output and input sequences $y(k)$, $y(k-1)$, $\dots$, $u(k-1)$, $u(k-2)$, $\dots$ together with the state-space model of the plant.

See IFAB PB p. 57 for details.

Reconstruction Using An Observer



Simulate the system ("predict" the state):

$$\hat{x}(k+1) = \Phi\hat{x}(k) + \Gamma u(k)$$
$$\hat{y}(k) = C\hat{x}(k)$$

Introduce "feedback" from measured $y(k)$ in order to adjust ("correct") the state estimate if the simulated output disagrees with the measured output

$$\hat{x}(k+1) = \Phi\hat{x}(k) + \Gamma u(k) + K\Big(y(k) - C\hat{x}(k)\Big)$$

Reconstruction Using An Observer

Form the estimation error $\widetilde{x} = x - \hat{x}$

$$\begin{aligned} \widetilde{x}(k+1) &= \Phi\widetilde{x}(k) - KC\widetilde{x}(k) \\ &= [\Phi - KC]\widetilde{x}(k) \end{aligned}$$

- Any observer pole placement possible, provided the observability matrix
$$W_o = \begin{pmatrix} C \\ \vdots \\ C\Phi^{n-1} \end{pmatrix}$$
has full rank
- Also when the system is not observable the estimation error will still go to zero as long as the system is detectable
- Choose K to get good convergence but not too much amplification of measurement noise

Deadbeat Observer

A *deadbeat observer* is obtained if the observer gain K is chosen so that the matrix $\Phi - KC$ has all eigenvalues zero.

The observer error goes to zero in finite time (in at most n steps, where n is the order of the system)

Noise sensitive (fast observer dynamics)

Equivalent to reconstruction using direct calculations.

Observer for the Double Integrator

$$\Phi - KC = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 - k_1 & h \\ -k_2 & 1 \end{pmatrix}$$

Characteristic equation

$$z^2 - (2 - k_1)z + 1 - k_1 + k_2 h = 0$$

Desired characteristic equation:

$$z^2 + p_1 z + p_2 = 0$$

Gives:

$$\begin{aligned} 2 - k_1 &= -p_1 \\ 1 - k_1 + k_2 h &= p_2 \end{aligned}$$

Observer for the Double Integrator cont'd

Solution:

$$\begin{aligned} k_1 &= 2 + p_1 \\ k_2 &= (1 + p_1 + p_2)/h \end{aligned}$$

Assume deadbeat observer $\left(p_1 = p_2 = 0\right)$

$$\begin{aligned} k_1 &= 2 \\ k_2 &= 1/h \end{aligned}$$

Resulting observer

$$\begin{aligned} \hat{x}_1(k+1) &= \hat{x}_1(k) + h\hat{x}_2(k) + \frac{h^2}{2}u(k) + 2\Big(y(k) - \hat{x}_1(k)\Big) \\ \hat{x}_2(k+1) &= \hat{x}_2(k) + hu(k) + \frac{1}{h}\Big(y(k) - \hat{x}_1(k)\Big) \end{aligned}$$

An Alternative Observer

The observer presented so far has a one sample delay:
$\hat{x}(k \mid k-1)$ depends only on measurements up to time $k-1$.

Alternative observer with direct term:

$$\begin{aligned}\hat{x}(k \mid k) &= \Phi\hat{x}(k-1 \mid k-1) + \Gamma u(k-1) \\ &\qquad + K\Big[y(k) - C\Big(\Phi\hat{x}(k-1 \mid k-1) + \Gamma u(k-1)\Big)\Big] \\ &= (I - KC)\Big(\Phi\hat{x}(k-1 \mid k-1) + \Gamma u(k-1)\Big) + Ky(k)\end{aligned}$$

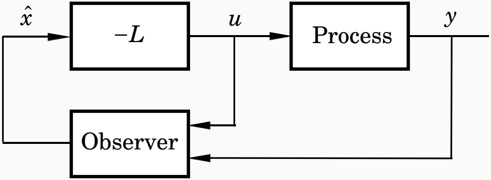
Reconstruction error:

$$\widetilde{x}(k \mid k) = x(k) - \hat{x}(k \mid k) = (\Phi - KC\Phi)\,\widetilde{x}(k-1 \mid k-1)$$

- $\Phi - KC\Phi$ can be given arbitrary eigenvalues if $\Phi - KC$ can
- K may be chosen so that some of the states will be observed directly through $y \Rightarrow$ the order of the observer can be reduced
    - Reduced order observer or *Luenberger observer*

Output Feedback

State feedback from observed state:



Controller:

$$\begin{aligned} \hat{x}(k+1) &= \Phi\hat{x}(k) + \Gamma u(k) + K(y(k) - C\hat{x}(k)) \\ u(k) &= -L\hat{x}(k) \end{aligned}$$

Controller transfer function (from y to u):

$$H_c(z) = -L(zI - \Phi + \Gamma L + KC)^{-1}K$$

Analysis of the Closed-Loop System

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ \widetilde{x}(k+1) &= (\Phi - KC)\widetilde{x}(k) \\ u(k) &= -L\hat{x}(k) = -L(x(k) - \widetilde{x}(k)) \end{aligned}$$

Eliminate $u(k)$

$$\begin{pmatrix} x(k+1) \\ \widetilde{x}(k+1) \end{pmatrix} = \begin{pmatrix} \Phi - \Gamma L & \Gamma L \\ 0 & \Phi - KC \end{pmatrix} \begin{pmatrix} x(k) \\ \widetilde{x}(k) \end{pmatrix}$$

Separation

Control poles: $A_c(z) = \det(zI - \Phi + \Gamma L)$

Observer poles: $A_o(z) = \det(zI - \Phi + KC)$

The observer poles should normally be considerably faster than the controller poles

Disturbance Estimation

How to handle disturbances that can not be modeled as impulse disturbances in the process state?

Assume that the process is described by

$$\begin{aligned}\frac{dx}{dt} &= Ax + Bu + v\\ y &= Cx\end{aligned}$$

where v is a disturbance modeled as

$$\begin{aligned}\frac{dw}{dt} &= A_w w\\ v &= C_w w\end{aligned}$$

Since disturbances typically have most of their energy at low frequencies, A_w often has eigenvalues in the origin (constant disturbance) or on the imaginary axis (sinusoidal disturbance)

Disturbance Estimation

Augment the state vector: $\begin{pmatrix} x \\ w \end{pmatrix}$

Gives the augmented system

$$\frac{d}{dt}\begin{pmatrix} x \\ w \end{pmatrix} = \begin{pmatrix} A & C_w \\ 0 & A_w \end{pmatrix} \begin{pmatrix} x \\ w \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u$$

$$y = \begin{pmatrix} C & 0 \end{pmatrix} \begin{pmatrix} x \\ w \end{pmatrix}$$

Sample this using ZOH:

$$\begin{pmatrix} x(k+1) \\ w(k+1) \end{pmatrix} = \begin{pmatrix} \Phi & \Phi_{xw} \\ 0 & \Phi_w \end{pmatrix} \begin{pmatrix} x(k) \\ w(k) \end{pmatrix} + \begin{pmatrix} \Gamma \\ 0 \end{pmatrix} u(k)$$

$$y = \begin{pmatrix} C & 0 \end{pmatrix} \begin{pmatrix} x(k) \\ w(k) \end{pmatrix}$$

Augmented Observer and State Feedback

Augmented observer:

$$\begin{pmatrix} \hat{x}(k+1) \\ \hat{w}(k+1) \end{pmatrix} = \begin{pmatrix} \Phi & \Phi_{xw} \\ 0 & \Phi_w \end{pmatrix} \begin{pmatrix} \hat{x}(k) \\ \hat{w}(k) \end{pmatrix} + \begin{pmatrix} \Gamma \\ 0 \end{pmatrix} u(k) + \begin{pmatrix} K \\ K_w \end{pmatrix} \epsilon(k)$$

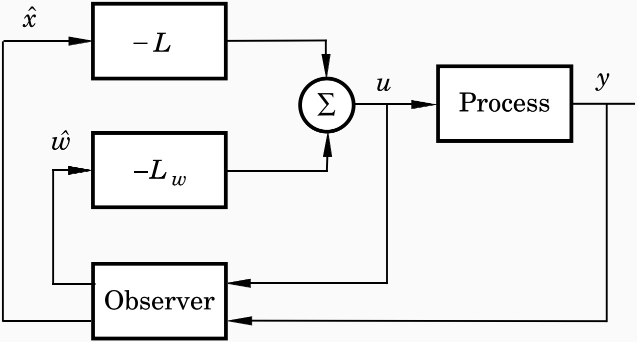
with $\epsilon(k) = y(k) - C\hat{x}(k)$

Augmented state feedback control law:

$$u(k) = -L\hat{x}(k) - L_w\hat{w}(k)$$

If possible, select L_w such that $\Phi_{xw} - \Gamma L_w = 0$

Disturbance Estimation: Block Diagram



Disturbance Estimation: Closed-Loop System

The closed-loop system can be written

$$
\begin{aligned}
x(k+1) &= (\Phi - \Gamma L)x(k) + (\Phi_{xw} - \Gamma L_w)w + \Gamma L\widetilde{x}(k) + \Gamma L_w\widetilde{w} \\
w(k+1) &= \Phi_w w(k) \\
\widetilde{x}(k+1) &= (\Phi - KC)\widetilde{x}(k) + \Phi_{xw}\widetilde{w}(k) \\
\widetilde{w}(k+1) &= \Phi_w \widetilde{w}(k) - K_w C\widetilde{x}(k)
\end{aligned}
$$

- L ensures that x goes to zero at the desired rate after a disturbance.
- The gain L_w reduces the effect of the disturbance v on the system by feedforward from the estimated disturbance $\hat{w}$.
- K and K_w influence the rate at which the estimation errors go to zero.

Special Case: Constant Input Disturbance

Assume constant disturbance acting on the plant input:

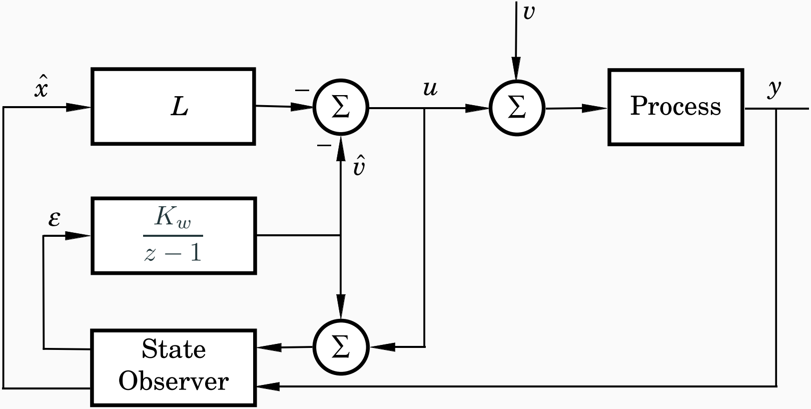
- $v = w$
- $\Phi_w = 1$
- $\Phi_{xw} = \Gamma$

If we choose $L_w = 1$ we will have perfect cancellation of the load disturbance

New controller + estimator

$$\begin{aligned} u(k) &= -L\hat{x}(k) - \hat{v}(k) \\ \hat{x}(k+1) &= \Phi\hat{x}(k) + \Gamma\Big(\hat{v}(k) + u(k)\Big) + K\epsilon(k) \\ \hat{v}(k+1) &= \hat{v}(k) + K_w\epsilon(k) \qquad \text{(integrator)} \\ \epsilon(k) &= y(k) - C\hat{x}(k) \end{aligned}$$

Special Case: Block Diagram



Example – Design

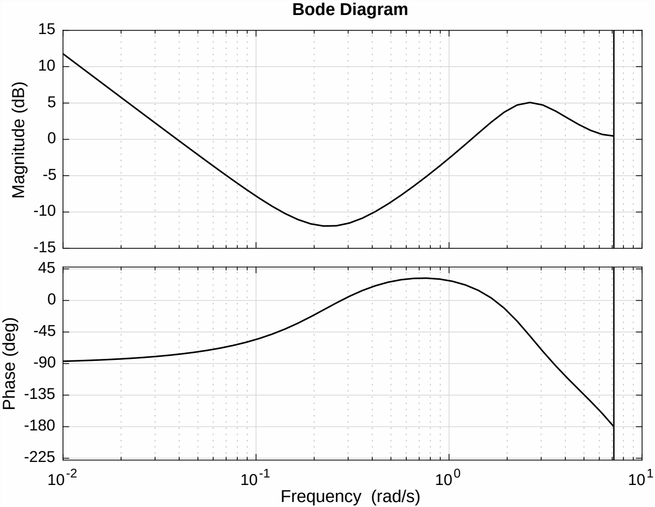
- Control of double integrator

$$\frac{dx}{dt} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$
$$y = \begin{pmatrix} 1 & 0 \end{pmatrix} x$$

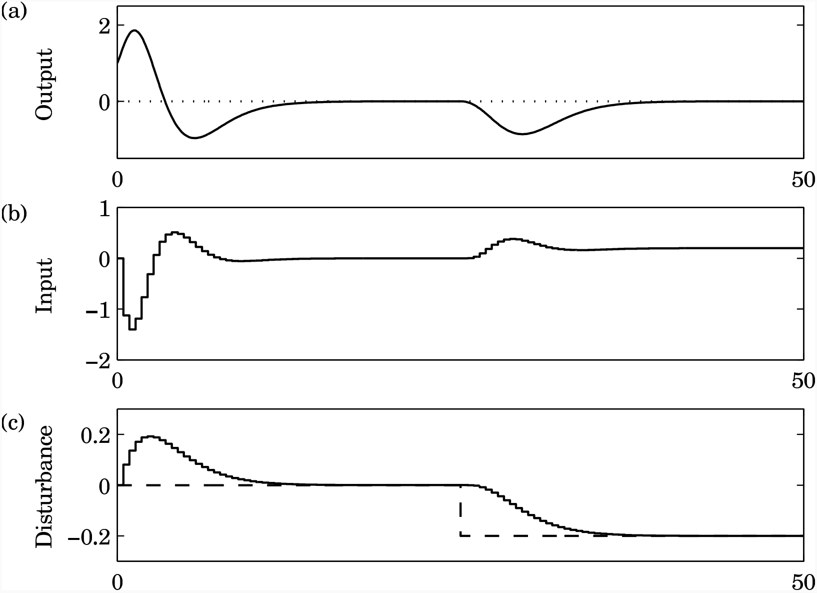
- Sample with $h = 0.44$
- Discrete state feedback designed based on continuous-time specification $\omega = 1$, $\zeta = 0.7$
  - Gives $L = [0.73 \;\; 1.21]$
- Extended observer assuming constant input disturbance to obtain integral action; all three poles placed in $z = 0.75$.
  - Gives $K^T = [0.75 \;\; 0.41]$, $K_w = 0.08$

Example – Design

Bode diagram of resulting controller:



Example – Simulation



Pseudo-Code Structure

State feedback using observer with one sample delay

```
y = ReadInput();
u = -L*x_hat;
WriteOutput(u);
x_hat = Phi*x_hat + Gamma*u + K*(y - C*x_hat);
```

Pseudo-Code Structure

State feedback using observer with direct term

```
y = ReadInput();
x_hat = Phi*x_hat + Gamma*u + K*(y - C(Phi*x_hat + Gamma*u)):
u = -L*x_hat;
WriteOutput(u);
```

Can be simplified further in order to minimize the computation time

```
y = ReadInput();
x_hat = x_temp + K*y;
u = -L*x_hat;
WriteOutput(u);
x_temp = (I-K*C)*(Phi*x_hat + Gamma*u);
```

Optimization-Based Design

Pole placement used to calculate L and K in this course

In the course Multivariable Control (Flervariabel Reglering), L and K are instead derived through optimization

- LQ (Linear Quadratic) and LQG (Linear Quadratic Gaussian) control
- Short overview in Chapter 11 of IFAC PB
- Not part of this course

Example in Matlab

```
% Define continuous-time process
A = [0 1; 0 0];
B = [0; 1];
C = [1 0];

% Sample the process using ZOH
h = 0.44;
[Phi,Gamma] = c2d(A,B,h)
Hp = ss(Phi,Gamma,C,0,h)

% Specify desired poles in continuous time
omega = 1; zeta = 0.7;
pc = roots([1 2*zeta*omega omega^2])
% Calculate corresponding desired discrete-time poles
pd = exp(pc*h)

% Design state feedback
L = place(Phi,Gamma,pd)

% Define augmented system and design observer
Gammae = [Gamma; 0]
Phie = [Phi Gamma; zeros(1,2) 1];
Ce = [C 0];
Ke = acker(Phie',Ce',[0.75 0.75 0.75])'
Le = [L 1]

% Form complete controller
Hc = ss(Phie-Gammae*Le-Ke*Ce,Ke,Le,0,h)
bode(Hc)         % draw controller Bode plot
margin(Hp*Hc)    % check stability margins
```