INTRODUCTION AND REVIEW OF MULTIVARIABLE CONTROL

GOALS OF FEEDBACK CONTROL: Change dynamic response of a system to have desired properties.

- System stabilized, has good transient and steady-state response.
- Output of system tracks reference input.
- Disturbances are rejected.

MULTIVARIABLE, STATE-SPACE CONTROL:

- Use primarily time-domain matrix representations of systems.
- Very powerful. Can often place poles *anywhere we want!*
- Same methods work for single-input, single-output (SISO) or multi-input, multi-output (MIMO or multivariable) systems.
- Both regulator poles and estimator poles need to be placed—Where should they go?
- This course covers design of *optimal* $\mathscr{H}_2$ and $\mathscr{H}_\infty$ linear controllers.

1.1: State-space dynamic systems (continuous-time)

- Representation of nth-order system as first-order differential equation in an n-vector called the state ➠ n first-order equations.
- Fundamental form of linear-time-invariant state-space model:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t),$$

where $u(t)$ is the input, $y(t)$ is the output, $x(t)$ is the "state", A, B, C, D are constant matrices.

DEFINITION: The state of a system at time t_0 is the minimum amount of information at t_0 that, together with the input $u(t)$, $t \geq t_0$, uniquely determines the behavior of the system for all $t \geq t_0$.

- Contrast with impulse-response (convolution) representation which requires all past history of $u(t)$

$$y(t) = \int_0^t h(\tau)u(t-\tau)\,\mathrm{d}\tau.$$

State-space versus transfer functions

- In ECE5520, we learned how to convert differential equations into state-space canonical forms.
- Can be transformed into any other equivalent state-space form via transformation. If $x(t) = Tz(t)$ and T is an invertible (similarity) transformation matrix,

$$\dot{z}(t) = \underbrace{T^{-1}AT}_{\bar{A}}\, z(t) + \underbrace{T^{-1}B}_{\bar{B}}\, u(t)$$

$$y(t) = \underbrace{\;CT\;}_{\bar{C}}\, z(t) + \underbrace{\;D\;}_{\bar{D}}\, u(t).$$

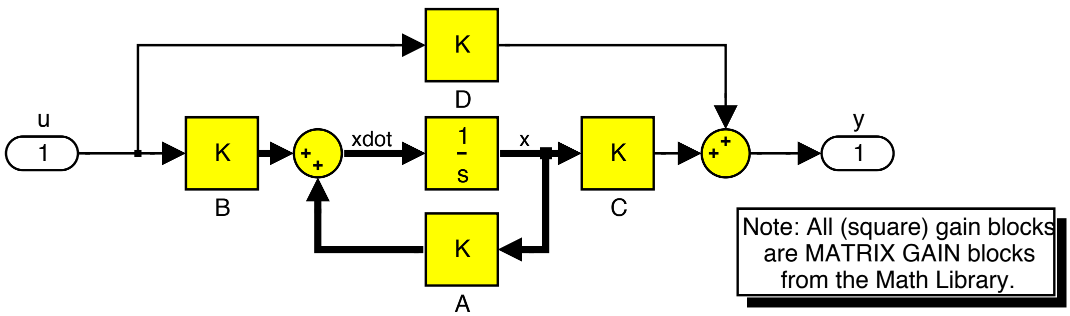
- A state-space system may be converted to a transfer function (input-output relationship) via

$$Y(s) = \underbrace{[C(sI-A)^{-1}B + D]}_{\text{transfer function of system}}\, U(s) + \underbrace{C(sI-A)^{-1}x(0)}_{\text{response to initial conditions}}\;.$$

- Characteristic equation for the system is $\chi(s) = \det(sI - A) = 0$.
- Poles of system are roots of $\det(sI - A) = 0$, the eigenvalues.

- In transfer function matrix form, $G(s) = C(sI - A)^{-1}B + D$, a pole of *any* entry in $G(s)$ is a pole of the system.

SIMULATING SYSTEMS IN SIMULINK: The following method is a direct implementation of the transfer function above, and the initial state may be set by setting the initial integrator values.



Time (dynamic) state response

- Homogeneous response: $x(t) = e^{At}x(0)$.
- Forced response:

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)\,\mathrm{d}\tau$$

$$y(t) = \underbrace{Ce^{At}x(0)}_{\text{initial resp.}} + \underbrace{\int_0^t Ce^{A(t-\tau)}Bu(\tau)\,\mathrm{d}\tau}_{\text{convolution}} + \underbrace{Du(t)}_{\text{feedthrough}}.$$

- Easiest to solve if A is diagonalized; then, $e^{At} = Ve^{\Lambda t}V^{-1}$, and

$$e^{\Lambda t} = \begin{bmatrix} e^{\lambda_1 t} & & & 0 \\ & e^{\lambda_2 t} & & \\ & & \ddots & \\ 0 & & & e^{\lambda_n t} \end{bmatrix},$$

where V is the matrix of eigenvectors; the λ_i are the eigenvalues.

- If A cannot be diagonalized, it can be put into Jordan form, *i.e.,*

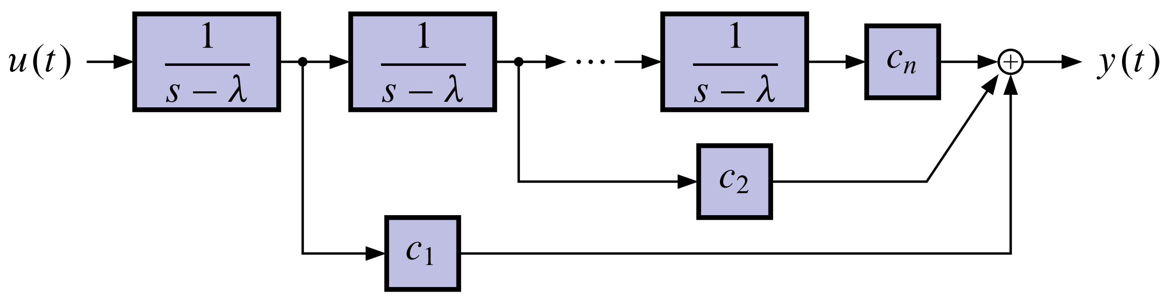
$$T^{-1}AT = J = \begin{bmatrix} J_1 & & 0 \\ & \ddots & \\ 0 & & J_q \end{bmatrix}$$

where

$$J_i = \begin{bmatrix} \lambda_i & 1 & & 0 \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda_i \end{bmatrix} \in \mathbb{R}^{n_i \times n_i}$$

is called a *Jordan block* of size n_i with eigenvalue λ_i (so $n = \sum_{i=1}^{q} n_i$).

- System decomposed into independent Jordan chains $\dot{\tilde{x}}_i(t) = J_i \tilde{x}_i(t)$



- In the time domain

$$(sI - J_\lambda)^{-1} = \begin{bmatrix} s-\lambda & -1 & & 0 \\ & s-\lambda & \ddots & \\ & & \ddots & -1 \\ 0 & & & s-\lambda \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} (s-\lambda)^{-1} & (s-\lambda)^{-2} & \cdots & (s-\lambda)^{-k} \\ & (s-\lambda)^{-1} & \cdots & (s-\lambda)^{-k+1} \\ & & \ddots & \vdots \\ 0 & & & (s-\lambda)^{-1} \end{bmatrix}$$

$$= (s-\lambda)^{-1}I + (s-\lambda)^{-2}F_1 + \cdots + (s-\lambda)^{-k}F_k$$

where F_k is the matrix with ones on the kth upper diagonal.

- Hence, the matrix exponential is

$$e^{J_\lambda t} = e^{\lambda t}\begin{bmatrix} 1 & t & \cdots & t^{k-1}/(k-1)! \\ & 1 & \cdots & t^{k-2}/(k-2)! \\ & & \ddots & \vdots \\ 0 & & & 1 \end{bmatrix}$$

$$= e^{\lambda t}(I + tF_1 + \cdots + t^{k-1}/(k-1)!F_k).$$

- Thus, Jordan blocks yield repeated poles and terms of the form $t^p e^{\lambda t}$ in e^{At}.

Blocking Zero

- A blocking zero is a value s_0 for which $G(s_0)$ is identically zero.
- Put in $u(t) = u_0 e^{s_0 t}$ and you get zero output (except for output due to initial conditions).
- Not considered a very useful definition of MIMO zero.

Transmission Zero

- Put in $u(t) = u_0 e^{z_i t}$ and you get a zero output *at "frequency"* $e^{z_i t}$.
- State space: Have input and state contributions (consider first the SISO case)

$$u(t) = u_0 e^{z_i t}, \qquad x(t) = x_0 e^{z_i t} \quad \ldots \quad y(t) = 0.$$

$$\dot{x}(t) = Ax(t) + Bu(t) \Rightarrow z_i e^{z_i t} x_0 = Ax_0 e^{z_i t} + Bu_0 e^{z_i t}$$

$$\Rightarrow \left[\begin{array}{c:c} z_i I - A & B \end{array}\right] \left[\begin{array}{c} x_0 \\ \hdashline -u_0 \end{array}\right] = 0$$

$$y(t) = Cx(t) + Du(t) \Rightarrow Cx_0 e^{z_i t} + Du_0 e^{z_i t} = 0$$

$$\Rightarrow \left[\begin{array}{c:c} -C & D \end{array}\right] \left[\begin{array}{c} x_0 \\ \hdashline -u_0 \end{array}\right] = 0.$$

- Put the two together

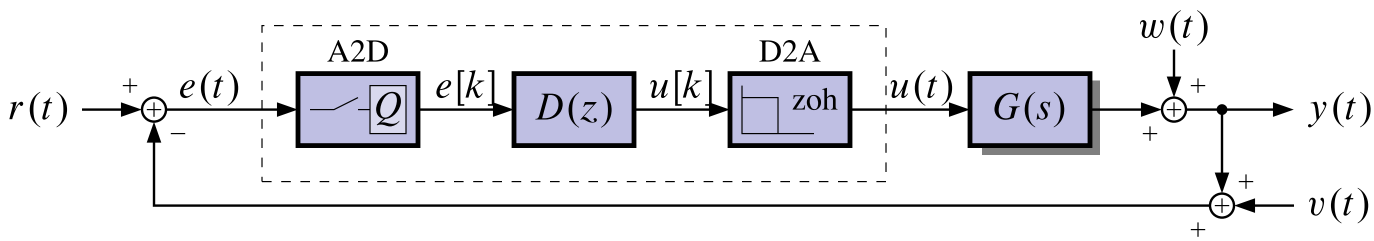
$$\left[\begin{array}{c:c} z_i I - A & B \\ \hdashline -C & D \end{array}\right] \left[\begin{array}{c} x_0 \\ \hdashline -u_0 \end{array}\right] = 0.$$

- Zero at frequency z_i if

$$\operatorname{rank} \left[\begin{array}{c:c} z_i I - A & B \\ \hdashline -C & D \end{array}\right] < n + \min\{p, q\}$$

1.2: State-space dynamic systems (discrete-time); stability

- Computer control requires analog-to-digital (A2D) and digital-to-analog (D2A) conversion.



- Discrete-time systems can also be represented in state-space form.

$$x[k+1] = A_d x[k] + B_d u[k]$$

$$y[k] = C_d x[k] + D_d u[k]$$

- The subscript "d" is used here to emphasize that, in general, the "A", "B", "C" and "D" matrices are *different* for discrete-time and continuous-time systems, even if the underlying plant is the same.
- I will usually drop the "d" and expect you to interpret the system from its context.
- The frequency-domain response of a discrete-time system is

$$Y(z) = \underbrace{[C(zI-A)^{-1}B + D]}_{\text{transfer function of system}} U(z) + \underbrace{C(zI-A)^{-1}zx[0]}_{\text{response to initial conditions}}.$$

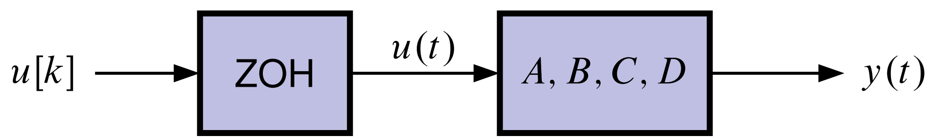
- Same form as for continuous-time systems.
- Poles of system are roots of $\det(zI - A) = 0$.
- The homogeneous time-domain response is $x[k] = A^k x[0]$.
- The full solution is:

$$x[k] = A^k x[0] + \underbrace{\sum_{j=0}^{k-1} A^{k-1-j} B u[j]}_{\text{convolution}}$$

$$y[k] = \underbrace{CA^k x[0]}_{\text{initial resp.}} + \underbrace{\sum_{j=0}^{k-1} CA^{k-1-j} B u[k]}_{\text{convolution}} + \underbrace{Du[k]}_{\text{feedthrough}}.$$

Converting plant dynamics to discrete time

- Combine the dynamics of the zero-order hold and the plant.



- The continuous-time dynamics of the plant are:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t).$$

- Then,

$$x[k+1] = A_d x[k] + B_d u[k]$$

where $A_d = e^{AT}$, $B_d = \int_0^T e^{A\sigma} B \, \mathrm{d}\sigma = A^{-1}(A_d - I)B$.

- Similarly,

$$y[k] = Cx[k] + Du[k].$$

That is, $C_d = C$; $D_d = D$.

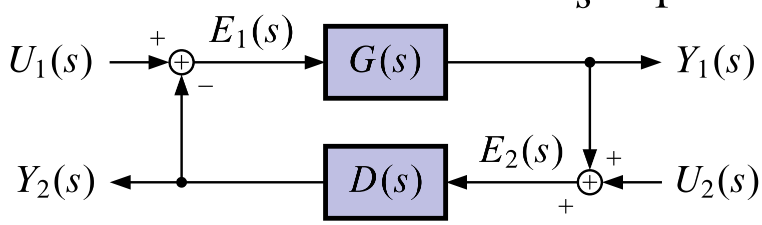
Internal stability

- In ECE5520, we studied two basic definitions of system stability.

- BIBO stability requires that the output is bounded for every possible bounded input.
- A system may be input-output stable and still have unbounded internal signals.
- The issue is *internal stability.* So, we also studied Lyapunov stability.

■ Here, we look at a slightly different formulation of Lyapunov stability.

■ Consider the following diagram. Let $G(s) = \dfrac{s}{s-1}$ and $D(s) = \dfrac{2}{s}$.



■ The transfer function of the system is $\dfrac{Y_1(s)}{U_1(s)} = \dfrac{s}{s+1}$ so the system is BIBO stable.

■ But, $\dfrac{E_1(s)}{U_2(s)} = \dfrac{-2(s-1)}{s(s+1)}$. The control input is unbounded if the measurement error contains a constant bias. The system is not internally stable.

■ Consider

$$\begin{bmatrix} E_1(s) \\ \hdashline E_2(s) \end{bmatrix} = \left[\begin{array}{c:c} (I+DG)^{-1} & -(I+DG)^{-1}D \\ \hdashline (I+GD)^{-1}G & (I+GD)^{-1} \end{array}\right] \begin{bmatrix} U_1(s) \\ \hdashline U_2(s) \end{bmatrix},$$

which are the transfer functions from the inputs to the "errors". (equivalent to transfer functions from inputs to outputs).

■ System is internally stable if all four transfer functions are stable.

1.3: Observability and controllability

Observability: Where am I?

- For either continuous- or discrete-time, define

$$\mathcal{O}(C,A) = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}.$$

- If $\mathcal{O}(C,A)$ is full rank, then the system is observable.
  - Can estimate all members of state vector from input and output.

Controllability: Can I get there from here?

- For either continuous- or discrete-time, define

$$\mathcal{C}(A,B) = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}.$$

- If $\mathcal{C}(A,B)$ is full rank, the system is controllable.
  - Can move all members of the state vector anywhere we want.

Continuous-time controllability Gramian

- If a continuous-time system is controllable, then

$$L_c(t) = \int_0^t e^{A\tau} B B^T e^{A^T\tau}\,\mathrm{d}\tau$$

is nonsingular for $t > 0$.

- Furthermore, the input

$$u(t) = -B^T e^{A^T(t_1-t)} L_c^{-1}(t_1)\left[e^{At_1}x_0 - x_1\right]$$

will transfer the state x_0 at time 0 to x_1 at time t_1.

- If a continuous-time system is controllable, and if it is also stable, then

$$L_c = \int_0^\infty e^{A\tau} B B^T e^{A^T \tau} \, \mathrm{d}\tau$$

can be found by solving for the unique (positive-definite) solution to the (Lyapunov) equation

$$A L_c + L_c A^T = -B B^T.$$

  - L_c is called the controllability Gramian.

- L_c measures the minimum energy required to reach a desired point x_1 starting at $x(0) = 0$ (with no limit on t)

$$\min \left\{ \int_0^t \|u(\tau)\|^2 \, \mathrm{d}\tau \quad \middle| \quad x(0) = 0,\ x(t) = x_1 \right\} = x_1^T L_c^{-1} x_1.$$

Continuous-time observability Gramian

- If a system is observable,

$$L_o(t) = \int_0^t e^{A^T \tau} C^T C e^{A\tau} \, \mathrm{d}\tau$$

is nonsingular for $t > 0$.

- Furthermore, we can find the initial state

$$x(0) = L_o^{-1}(t_1) \int_0^{t_1} e^{A^T t} C^T \bar{y}(t) \, \mathrm{d}t$$

where

$$\bar{y}(t) = y(t) - C \int_0^t e^{A(t-\tau)} B u(\tau) \, \mathrm{d}\tau - D u(t) = C e^{At} x(0).$$

- If a continuous-time system is observable, and if it is also stable, then

$$L_o = \int_0^\infty e^{A^T \tau} C^T C e^{A\tau} \, \mathrm{d}\tau$$

can be found as the unique (positive-definite) solution to the (Lyapunov) equation

$$A^T L_o + L_o A = -C^T C.$$

- L_o is called the observability Gramian.

- If measurement (sensor) noise is IID $\mathcal{N}(0, \sigma^2 I)$ then L_o is a measure of error covariance in measuring $x(0)$ from u and y over longer and longer periods

$$\lim_{t\to\infty} \mathbb{E}\,\|\hat{x}(0) - x(0)\|^2 = \sigma^2 x(0)^T L_o^{-1} x(0).$$

Discrete-time controllability Gramian

- In discrete-time, if a system is controllable, then

$$W_{dc}[n-1] = \sum_{m=0}^{n-1} A^m B B^T (A^T)^m$$

is nonsingular.

- In particular,

$$W_{dc} = \sum_{m=0}^{\infty} A^m B B^T (A^T)^m$$

is called the discrete-time controllability Gramian and is the unique positive-definite solution to the Lyapunov equation

$$W_{dc} - A W_{dc} A^T = B B^T.$$

- W_{dc} measures the minimum energy required to reach a desired point x_1 starting at $x[0] = 0$ (with no limit on m)

$$\min \left\{ \sum_{k=0}^{m} \|u[k]\|^2 \;\middle|\; x[0] = 0,\; x[m] = x_1 \right\} = x_1^T W_{dc}^{-1} x_1.$$

Discrete-time observability Gramian

- In discrete-time, if a system is observable, then

$$W_{do}[n-1] = \sum_{m=0}^{n-1} (A^T)^m C C^T A^m$$

is nonsingular.

- In particular,

$$W_{do} = \sum_{m=0}^{\infty} (A^T)^m C C^T A^m$$

is called the discrete-time observability Gramian and is the unique positive-definite solution to the Lyapunov equation

$$W_{do} - A^T W_{do} A = C^T C.$$

- As with continuous-time, if measurement (sensor) noise is IID $\mathcal{N}(0, \sigma^2 I)$ then W_{do} is a measure of error covariance in measuring $x[0]$ from u and y over longer and longer periods

$$\lim_{t \to \infty} \mathbb{E} \|\hat{x}[0] - x[0]\|^2 = \sigma^2 x(0)^T W_{do}^{-1} x[0].$$

Transformation to controllability form

- Given a continuous- or discrete-time controllable system with matrices A, B, C, and D, the matrix T

$$T = [B \; AB \; \cdots \; A^{n-1}B] = \mathcal{C}$$

transforms the system into controllability form.

EXTENSION I: To convert between any two realizations,

$$T = \mathcal{C}_{\text{old}} \mathcal{C}_{\text{new}}^{-1}.$$

EXTENSION II: If $x_{\text{old}} = T x_{\text{new}}$ then $T = \mathcal{O}_{\text{old}}^{-1} \mathcal{O}_{\text{new}}$.

1.4: Controller design

- Control is accomplished using linear state feedback

$$u(t) = r(t) - Kx(t), \quad K \in \mathbb{R}^{1\times n}.$$

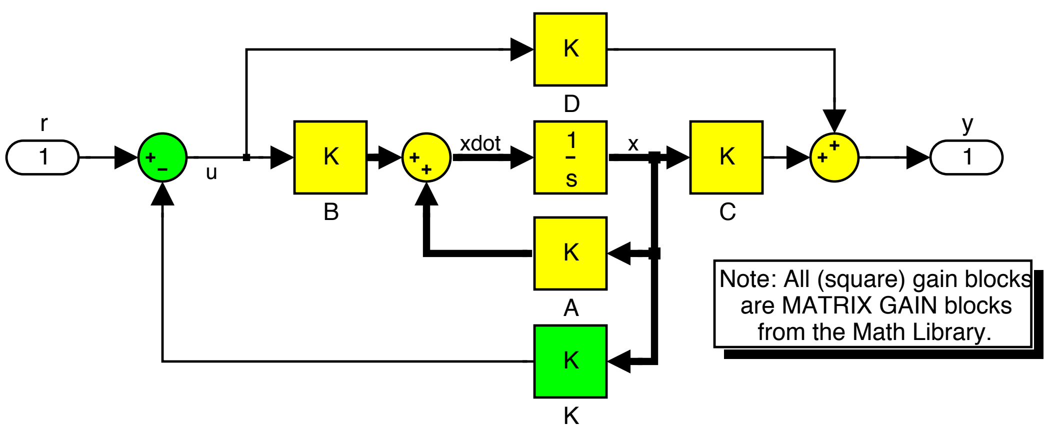
- Closed-loop poles are eigenvalues of $A - BK$.
- Open-loop system has characteristic equation

$$\chi(s) = s^n + a_1 s^{n-1} + \cdots + a_n.$$

- Desired closed-loop system has characteristic equation

$$\chi_d(s) = s^n + \alpha_1 s^{n-1} + \cdots + \alpha_n.$$

- Simulating state feedback in Simulink:



- Can use Bass–Gura, Ackermann, cyclic, or Lyapunov method to place closed-loop poles.

Reference tracking

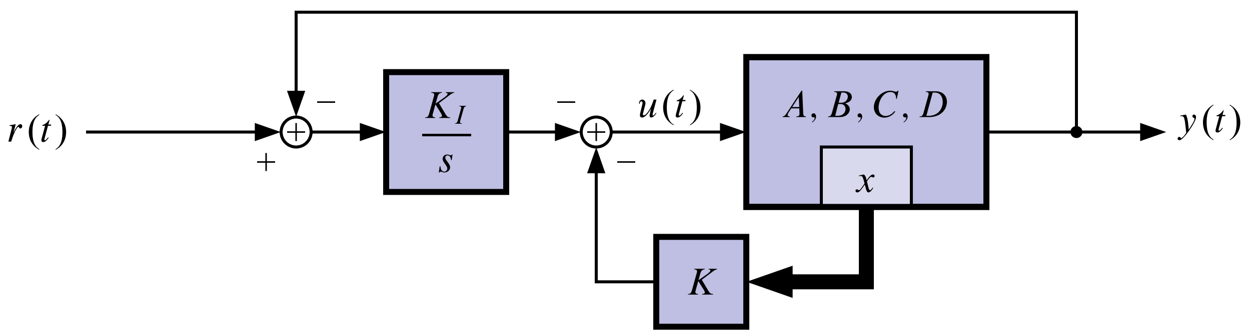
- Pole placement gives good transient response but often poor steady-state response.
- Can change steady-state response by computing

$$u(t) = N_u r(t) - K(x(t) - N_x r(t)).$$

- Can find N_x and N_u by solving

$$\left[\begin{array}{c:c} A & B \\ \hdashline C & D \end{array}\right]\left[\begin{array}{c} N_x \\ \hdashline N_u \end{array}\right]=\left[\begin{array}{c} 0 \\ \hdashline I \end{array}\right].$$

- We can also use $\bar{N} = N_u + KN_x$ such that $u(t) = \bar{N}r(t) - Kx(t)$.
- Integral control can also be used to counteract disturbances, plant variations, or other noises in the system.



- We can include the integral state into our normal state-space form by augmenting the system dynamics

$$\left[\begin{array}{c} \dot{x}_I(t) \\ \hdashline \dot{x}(t) \end{array}\right]=\left[\begin{array}{c:c} 0 & -C \\ \hdashline 0 & A \end{array}\right]\left[\begin{array}{c} x_I(t) \\ \hdashline x(t) \end{array}\right]+\left[\begin{array}{c} 0 \\ \hdashline B \end{array}\right]u(t)+\left[\begin{array}{c} I \\ \hdashline 0 \end{array}\right]r(t)$$

$$y(t) = Cx(t) + Du(t).$$

- Note that the new "A" matrix has an open-loop eigenvalue at the origin. This corresponds to increasing the system type, and integrates out steady-state error.
- The control law is,

$$u(t) = -\left[\begin{array}{c:c} K_I & K \end{array}\right]\left[\begin{array}{c} x_I(t) \\ \hdashline x(t) \end{array}\right].$$

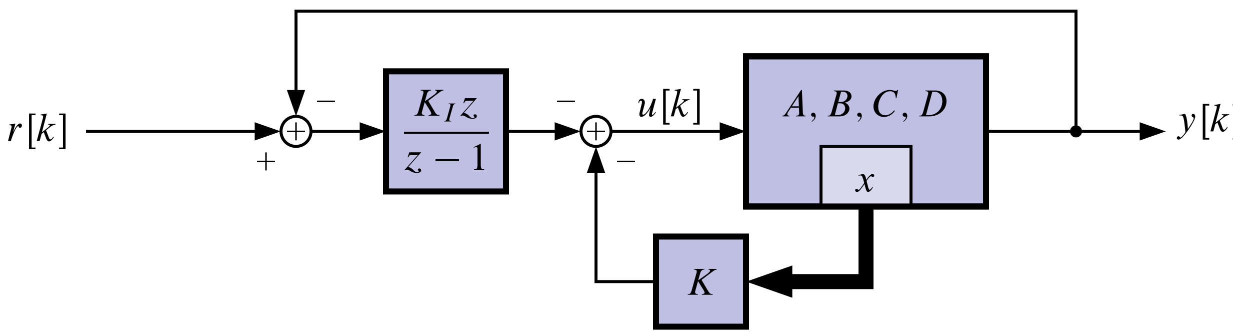
- So, we now have the task of choosing $n + n_I$ closed-loop poles.

State feedback for discrete-time systems

- The result is identical.

  Characteristic frequencies of controllable modes are freely assignable by state feedback; characteristic frequencies of uncontrollable modes do not change with state feedback.

- For reference tracking, we can again augment our system with a (discrete-time) integrator:



- We can include the integral state into our normal state-space form by augmenting the system dynamics

$$\begin{bmatrix} x_I[k+1] \\ \hdashline x[k+1] \end{bmatrix} = \begin{bmatrix} 1 & -C \\ 0 & A \end{bmatrix} \begin{bmatrix} x_I[k] \\ \hdashline x[k] \end{bmatrix} + \begin{bmatrix} 0 \\ \hdashline B \end{bmatrix} u[k] + \begin{bmatrix} I \\ \hdashline 0 \end{bmatrix} r[k]$$

$$y[k] = Cx[k] + Du[k].$$

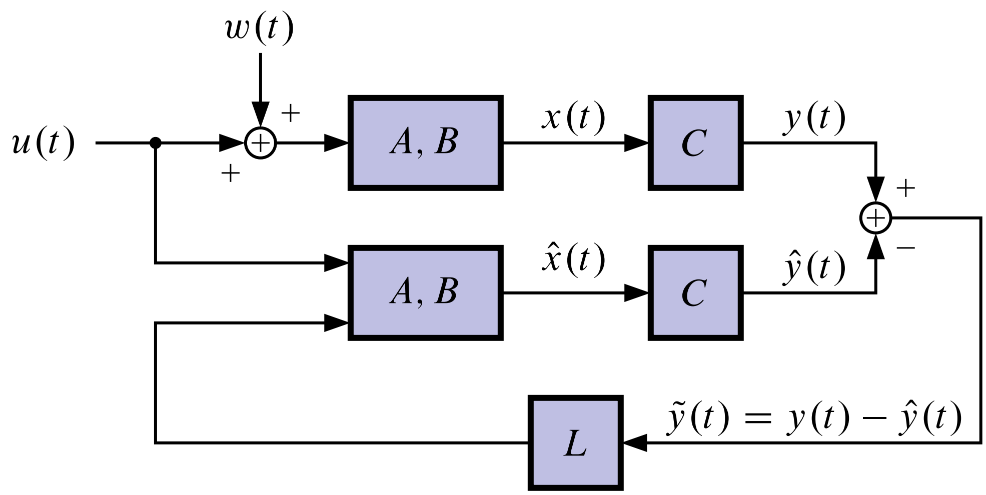
- Notice the new open-loop eigenvalue of "A" at $z = 1$.
- The control law is,

$$u[k] = -\begin{bmatrix} K_I & K \end{bmatrix} \begin{bmatrix} x_I[k] \\ \hdashline x[k] \end{bmatrix} + KN_x r[k].$$

- So, we now have the task of choosing $n + n_I$ closed-loop poles.

1.5: Closed-loop estimator design

- In the design of state-feedback control, we assumed that all states of our plant were measured ➠ often *impossible* or *expensive.*
- So, we investigated methods of reconstructing the plant state vector given only limited measurements.



- The closed-loop estimator equation was

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L\left(y(t) - C\hat{x}(t)\right).$$

- This had corresponding error

$$\dot{\tilde{x}}(t) = (A - LC)\,\tilde{x}(t),$$

or, $\hat{x}(t) \to x(t)$ if $A - LC$ is stable, for any value of $\hat{x}(0)$ and any $u(t)$, whether or not A is stable.

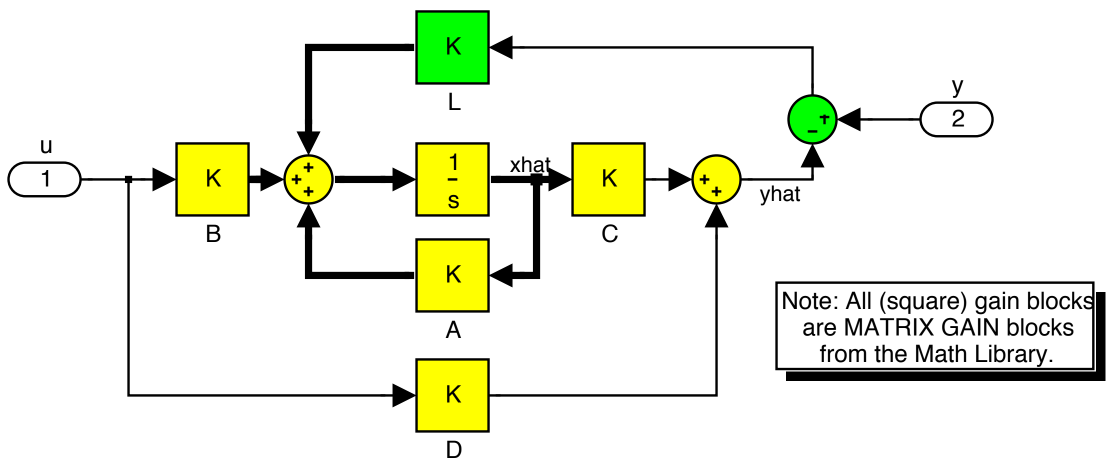
- This has dynamics related to the roots of the characteristic equation

$$\chi_{ob}(s) = \det\left(sI - A + LC\right) = 0.$$

- So, for our estimator, we specify the convergence rate of $\hat{x}(t) \to x(t)$ by choosing desired pole locations: Choose L such that

$$\chi_{ob,des}(s) = \det\left(sI - A + LC\right).$$

- This is called the "observer gain problem".
  - Can use Bass–Gura, Ackerman, cyclic, or Lyapunov methods for pole placement.
- In Simulink, the following diagram implements a closed-loop estimator. The output is `xhat`.



Discrete-time prediction estimator

- In discrete-time, we can do the same thing. The update equation for the closed-loop (prediction) estimator is

$$\hat{x}_p[k+1] = A\hat{x}_p[k] + Bu[k] + L_p\left(y[k] - C\hat{x}_p[k]\right).$$

- The prediction-estimation error can likewise be written as

$$\tilde{x}[k+1] = \left(A - L_pC\right)\tilde{x}[k],$$

which has dynamics related to the roots of the characteristic equation

$$\chi_{ob}(z) = \det\left(zI - A + L_pC\right) = 0.$$

- For our prediction estimator, we specify the convergence rate of $\hat{x}_p[k] \to x[k]$ by choosing desired pole locations: Choose L_p such that

$$\chi_{ob,des}(z) = \det\left(zI - A + L_p C\right).$$

Current estimator/ compensator

- "Time update": Predict new state from old state estimate and system dynamics

$$\hat{x}_p[k] = A\hat{x}_c[k-1] + Bu[k-1].$$

- "Measurement update": Measure the output and use that to update/correct the estimate

$$\hat{x}_c[k] = \hat{x}_p[k] + L_c\left(y[k] - C\hat{x}_p[k]\right).$$

- L_c is called the "current estimator gain."
- The prediction and current estimate errors have dynamics

$$\tilde{x}_p = x - \hat{x}_p \quad \Rightarrow \quad \tilde{x}_p[k+1] = (A - L_c CA)\,\tilde{x}_p[k]$$

$$\tilde{x}_c = x - \hat{x}_c \quad \Rightarrow \quad \tilde{x}_c[k+1] = (A - AL_c C)\,\tilde{x}_c[k].$$

Regulator design: Separation principle

- Now that we have a structure to estimate the state $x(t)$, let's feed back $\hat{x}(t)$ to control the plant. That is,

$$u(t) = r(t) - K\hat{x}(t),$$

where K was designed assuming that $u(t) = r(t) - Kx(t)$. Is this going to work? How risky is it to interconnect two well-behaved, stable systems? (Assume $r(t) = 0$ for now).

- Our combined closed-loop-system state equations are

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\tilde{x}}(t) \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x(t) \\ \tilde{x}(t) \end{bmatrix}.$$

- The $2n$ closed-loop poles of the combined regulator/estimator system are the n eigenvalues of $A - BK$ combined with the n eigenvalues of $A - LC$.

The Compensator

- In continuous-time, the compensator is

$$D(s) = \frac{U(s)}{Y(s)} = -K(sI - A + BK + LC - LDK)^{-1}L.$$

- In discrete-time, the prediction compensator is

$$D(z) = \frac{U(z)}{Y(z)} = -K\left(zI - A + BK + L_pC - L_pDK\right)^{-1} L_p.$$

- And, the current compensator is

$$D(z) = -KL_c - K(I - L_cC)\left(zI - (A - BK)(I - L_cC)\right)^{-1}(A - BK)L_c.$$

1.6: Vector and signal norms; MIMO frequency response

- Control system designed to "improve performance."
- Need a way to quantify performance. Then, we can make informed trade-off analysis of competing control-system designs.
- Can also talk about "optimal" control design with respect to some performance criterion.

Vector p-norms

- The p-norm of a vector is defined as

$$\|x\|_p = \lim_{\alpha \to p} \sqrt[\alpha]{\sum_{k=1}^{n_x} |x_k|^\alpha}.$$

- One common example is the (Euclidean) 2-norm

$$\|x\| = \|x\|_2 = \sqrt{x_1^2 + \cdots + x_n^2} = \sqrt{x^T x}.$$

- $\|x\|$ measures length of vector (from origin).
- A modified weighted 2-norm is

$$\|x\|_W = \sqrt{x^T W x}$$

for some positive-semi-definite matrix W.

- Another common vector norm is the ∞-norm

$$\|x\|_\infty = \lim_{\alpha \to \infty} \sqrt[\alpha]{\sum_{k=1}^{n_x} |x_k|^\alpha} = \max_k |x_k|.$$

Signal p-norms

- The linear space of signals is given by the symbol $\mathscr{L}$.
- The p-norm of a signal is defined as

$$\|x(t)\|_p = \lim_{\alpha \to p} \sqrt[\alpha]{\int_{-\infty}^{\infty} \sum_{k=1}^{n_x} |x_k(t)|^\alpha \, \mathrm{d}t}.$$

- The signal 2-norm is (signal energy)

$$\|x(t)\| = \|x(t)\|_2 = \sqrt{\int_{-\infty}^{\infty} x^T(t)x(t)\, \mathrm{d}t}.$$

- The linear space of all signals with finite 2-norm is denoted $\mathscr{L}_2$.
- The weighted signal 2-norm is defined as

$$\|x(t)\|_{W(t)} = \sqrt{\int_{-\infty}^{\infty} x^T(t)W(t)x(t)\, \mathrm{d}t},$$

where $W(t)$ is positive-semi-definite at all times.

- A generalization of the 2-norm is defined over finite time

$$\|x(t)\|_{2,[t_o,t_f]} = \sqrt{\int_{t_o}^{t_f} x^T(t)x(t)\, \mathrm{d}t}.$$

- The signal ∞-norm is defined as

$$\|x(t)\|_\infty = \sup_t \max_k |x_k(t)|.$$

The supremum is used since the set of times is infinite: $x_k(t)$ may approach a value it never reaches. *e.g.,* $1 - e^{-t}$, $t \ge 0$.

- The finite-time ∞-norm is

$$\|x(t)\|_{\infty,[t_o,t_f]} = \max_{t \in [t_o,t_f]} \max_k |x_k(t)|.$$

- The linear space of all signals with finite ∞-norm is denoted $\mathscr{L}_\infty$.

MIMO frequency response

- To compute system norms, we'll need to understand frequency response.
- The forced output of a linear system with sinusoidal input. Easily computed from transfer function.
- If $u(t) = u_o e^{j\omega t}$ then $y(t) = G(j\omega)u_o e^{j\omega t}$ in steady-state.

SISO: Change in magnitude (gain) and change in phase are termed the frequency response.

MIMO: Gain and phase shift depend on which input is used and which output is observed.

- More general method of determining frequency response is necessary.
- Consider the gain of the system

$$\text{Gain} = \frac{\|y(t)\|}{\|u(t)\|} = \frac{\left\|G(j\omega)u_o e^{j\omega t}\right\|}{\|u_o e^{j\omega t}\|} = \frac{\|G(j\omega)u_o\|}{\|u_o\|},$$

where $\|\cdot\|$ denotes the Euclidean vector norm

$$\|v\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}.$$

- The gain depends on both u_o and ω, so a unique gain as a function of frequency cannot be defined. Instead,

$$\min_{u_o} \frac{\|G(j\omega)u_o\|}{\|u_o\|} \le \text{Gain} \le \max_{u_o} \frac{\|G(j\omega)u_o\|}{\|u_o\|}$$

or

$$\min_{\|u_o\|=1} \|G(j\omega)u_o\| \le \text{Gain} \le \max_{\|u_o\|=1} \|G(j\omega)u_o\|$$

since multiplying input by scalar has no effect (can assume $\|u_o\| = 1$).

- (Phase response not meaningful for MIMO system since most phases usually possible via choosing a combination of inputs and outputs).

Singular-value decomposition

- The singular-value decomposition (SVD) is a way to factor a matrix $M \in \mathbb{C}^{n_y \times n_u}$

$$M = USV^{\dagger} = \sum_{i=1}^{p} \sigma_i U_i V_i^{\dagger}$$

where $U \in \mathbb{C}^{n_y \times n_y}$, $V \in \mathbb{C}^{n_u \times n_u}$ are unitary matrices, $p = \min\{n_y, n_u\}$, "†" denotes the conjugate transpose, and U_i and V_i are the ith columns of U and V, respectively.

- A unitary matrix is one such that $VV^{\dagger} = V^{\dagger}V = I$. Equivalently, a matrix whose columns are orthonormal.
- The vectors U_i and V_i are called the left- and right-singular vectors of M, respectively.
- The matrix $S \in \mathbb{C}^{n_y \times n_u}$ is "diagonal"

$$S = \left[\begin{array}{ccc:c} \sigma_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & \sigma_p & \end{array}\right] \text{ or } S = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_p \end{bmatrix} \text{ or } S = \left[\begin{array}{ccc} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_p \\ \hdashline & 0 & \end{array}\right],$$

when $n_y < n_u$, $n_y = n_u$ and $n_y > n_u$, respectively.

- The parameters σ_i are called the singular values of M and are ordered (by convention)

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n_u} \geq 0.$$

- The largest singular value is given notation $\overline{\sigma} = \sigma_1$.

- The smallest singular value is given notation $\underline{\sigma} = \sigma_{n_u}$.
- The singular-value decomposition shows how a matrix operates on a vector.

$$Mx = \sum_{i=1}^{p} \sigma_i U_i V_i^{\dagger} x.$$

- The term $V_i^{\dagger} x$ gives the length of the input in the direction defined by the given right singular vector.
- This length is multiplied by the associated singular value.
- This scales the vector U_i which contributes to the output.
- The singular-value decomposition has the property that

$$\frac{\|Mx\|}{\|x\|} \le \sigma_1 = \overline{\sigma} \quad \text{and} \quad \frac{\|Mx\|}{\|x\|} \ge \sigma_{n_u} = \underline{\sigma} = \begin{cases} \sigma_p, & n_y \ge n_u; \\ 0, & n_y < n_u. \end{cases}$$

- Useful fact: $\overline{\sigma}(MA) \le \overline{\sigma}(M)\overline{\sigma}(A)$.
- In MATLAB, `svd.m` and `svds.m`

Principal gains

- Back to the frequency-response problem. $y(t) = G(j\omega)u_o e^{j\omega t}$. System gain through the matrix $G(j\omega)$.
- The SVD is

$$G(j\omega) = \sum_{i=1}^{p} \sigma_i(\omega) U_i(\omega) V_i^{\dagger}(\omega).$$

- The frequency-dependent singular values of the transfer-function matrix are called the principle gains of the system.
- We need to keep track of only $\overline{\sigma}(G(j\omega))$ and $\underline{\sigma}(G(j\omega))$.

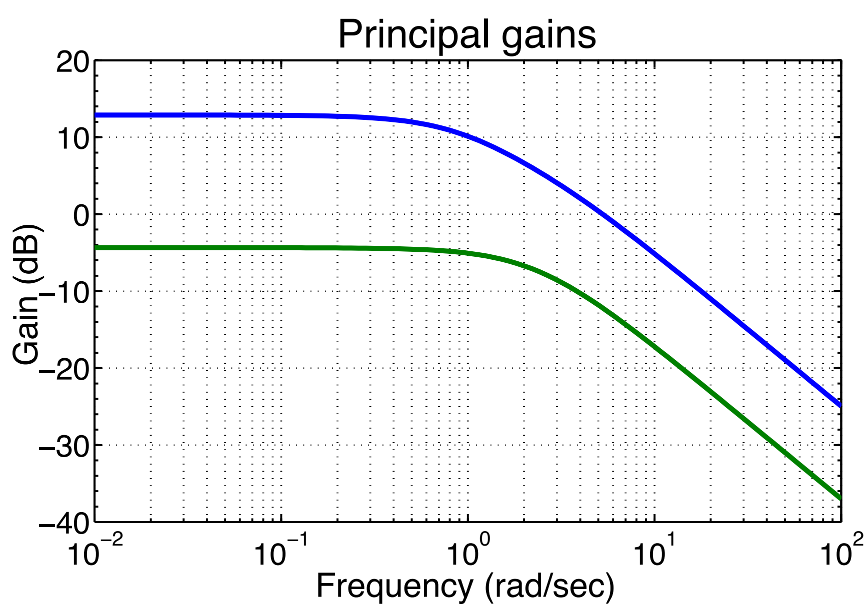
- The system gain is between these two values, as a function of ω.

EXAMPLE: Consider the transfer-function matrix

$$G(j\omega) = \begin{bmatrix} \dfrac{1}{j\omega+1} & \dfrac{4}{j\omega+1} \\ \dfrac{1}{j\omega+1} & \dfrac{-4(j\omega-1)}{(j\omega+1)(j\omega+3)} \end{bmatrix}.$$

```
omega = logspace(-2,2,100);
for i = 1:length(omega),
  Oi = omega(i);
  G = [ (1./(j*Oi+1)) (4./(j*Oi+1)); ...
      (1./(j*Oi+1)) (-4*(j*Oi-1)./((j*Oi+1).*(j*Oi+3)))];
  bigsigma(i) = max(svd(G));
  smallsigma(i) = min(svd(G));
end;
semilogx(omega,20*log10(bigsigma),omega,20*log10(smallsigma));
```

- Then, the principal gains are



1.7: System norms

The system 2-norm

- The system 2-norm computes system gain between input and output

$$\|G\|_2 = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} \text{trace}\left\{G^\dagger(j\omega)G(j\omega)\right\}\,\mathrm{d}\omega}.$$

- Note that $\text{trace}\{G^\dagger(j\omega)G(j\omega)\}$ is the sum of magnitude square of all elements of $G(j\omega)$. So, $\|G\|_2$ is proportional to average system gain.
- By Parseval's theorem, we can write

$$\|G\|_2 = \sqrt{\int_0^{\infty} \text{trace}\left\{g^T(t)g(t)\right\}\,\mathrm{d}t}.$$

- This integral exists only when system is stable—norm only makes sense when system stable.
- $\mathscr{H}_2$ is the space of all LTI systems with finite 2-norm.

Interpretation of the 2-norm

- Consider a system with i.i.d. white noise input

$$\mathbb{E}\left[w(t)w^T(t+\tau)\right] = S_w I\delta(\tau), \qquad S_w \text{ a scalar.}$$

- If $y(t)$ is the system output, then the mean-square output power is

$$\mathbb{E}\left[y^T(t)y(t)\right] = \text{trace}\{\Sigma_y\}$$

and

$$\Sigma_y = S_w\int_0^{\infty} g(t)g^T(t)\,\mathrm{d}t.$$

- Combining,

$$\mathbb{E}\left[y^T(t)y(t)\right] = \text{trace}\left\{S_w \int_0^\infty g(t)g^T(t)\,\mathrm{d}t\right\} = S_w \|G\|_2^2$$

and the RMS noise power at the output is

$$\|G\|_2 \sqrt{S_w}.$$

- Therefore, the system 2-norm is the gain between the square-root of spectral density of white i.i.d. input and RMS value of output.

Computing the 2-norm

- The impulse response of a stable system is

$$g(t) = \left\{Ce^{At}B + D\delta(t)\right\} 1(t).$$

- Substitute into expression for 2-norm

$$\|G\|_2 = \sqrt{\int_0^\infty \text{trace}\left\{[Ce^{At}B + D\delta(t)]^T[Ce^{At}B + D\delta(t)]\right\}\,\mathrm{d}t}.$$

- This integral involves $\delta^2(t)$ terms which have infinite integral. Therefore, $D = 0$ for finite 2-norm.
- Then,

$$\begin{aligned}\|G\|_2 &= \sqrt{\int_0^\infty \text{trace}\left\{B^T e^{A^T t} C^T C e^{At} B\right\}\,\mathrm{d}t} \\ &= \sqrt{\int_0^\infty \text{trace}\left\{C e^{At} B B^T e^{A^T t} C^T\right\}\,\mathrm{d}t},\end{aligned}$$

or

$$\|G\|_2 = \sqrt{\text{trace}\left\{ B^T \underbrace{\int_0^\infty e^{A^T t} C^T C e^{At}\, \text{d}t}_{\text{Obs Gramian } L_o} B \right\}}$$

$$= \sqrt{\text{trace}\left\{ C \underbrace{\int_0^\infty e^{At} B B^T e^{A^T t}\, \text{d}t}_{\text{Ctrl Gramian } L_c} C^T \right\}}.$$

- So,

$$\|G\|_2 = \sqrt{\text{trace}\{B^T L_o B\}} = \sqrt{\text{trace}\{C L_c C^T\}}$$

where

$$A^T L_o + L_o A = -C^T C$$

$$A L_c + L_c A^T = -B B^T$$

(Lyapunov equations which may be easily solved via MATLAB).

The system ∞-norm

- The system ∞-norm is the maximum gain over all frequencies.

$$\|G\|_\infty = \sup_\omega \overline{\sigma}[G(j\omega)].$$

- The ∞-norm is often used to provide worst-case robustness analysis.
- It gives a bound on maximum system 2-norm (gain)

$$\|g(t) * w(t)\|_2 \le \|G\|_\infty \|w(t)\|_2 .$$

- Re-arranging, we can also define

$$\|G\|_\infty = \sup_{w \neq 0} \frac{\|g(t) * w(t)\|_2}{\|w(t)\|_2}.$$

- Another property is

$$\|G_1G_2\|_\infty \le \|G_1\|_\infty \|G_2\|_\infty ,$$

shown by

$$\|g_1(t) * g_2(t) * w(t)\|_2 \le \|G_1\|_\infty \|g_2(t) * w(t)\|_2 \le \|G_1\|_\infty \|G_2\|_\infty \|w(t)\|_2 .$$

- Computing the ∞-norm "hard." Must plot $\overline{\sigma}[G(j\omega)]$ and find supremum.
- No closed-form solution.
- $\mathscr{H}_\infty$ is the space of all LTI systems with finite ∞-norm.

Weighted system norms

- In many cases, different constraints apply to different outputs. We will want to weight performance versus individual outputs differently.
- Can define weighted norms. *e.g.,*

$$\|GW_i\|_\infty = \sup_\omega \overline{\sigma}[G(j\omega)W_i(j\omega)]$$
$$\|W_oG\|_\infty = \sup_\omega \overline{\sigma}[W_o(j\omega)G(j\omega)],$$

where W_i is input weighting matrix, W_o is output weighting matrix.

- Both can be functions of frequency to provide different constraints/ penalties for signals at different frequencies.