MECH 4420 Lecture: State Space Control

David Bevly

State Space Control

- There are tools built to design controllers using state space representation
- These can be useful for all kinds of systems including vehicle dynamics
  - We already have a state space representation of the bicycle model:

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dfrac{-C_0}{mV} & -\left(1 + \dfrac{C_1}{mV^2}\right) \\ \dfrac{-C_1}{I_Z} & \dfrac{-C_2}{I_Z V} \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + \begin{bmatrix} \dfrac{C_{\alpha f}}{mV} \\ \dfrac{aC_{\alpha f}}{I_Z} \end{bmatrix} \delta$$

  - This model can easily be augmented to include heading and lateral position
  - Note that state space allows for multiple inputs (unlike transfer functions). This is advantageous in vehicle dynamics where there can be multiple inputs: front steer, rear steer, differential braking, etc.

State Space Control

- Recall the full state space representation:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

  - x is nx1 (n=# of states and also the system order)
  - y is mx1 (m=# of measurements or outputs)
  - u is Lx1 (L=# of inputs)
  - A is nxn
  - B is nxL
  - D is mxL (D is nearly always zero)
- Also recall the eigenvalues of the system will be the eigenvalues of the A Matrix!

Regulator

- Regulator: Drives states to zero $\dot{x} = Ax + Bu$
  - We will look at how to introduce a non-zero reference later
- Why not just set u=0?
  - Results in: $\dot{x} = Ax$
  - Note the differential equation becomes a homogeneous differential equation:

$$\dot{x} - Ax = 0$$

1) Doesn't guarantee $x \to 0$ (what if the system is unstable (i.e., eigenvalues of A are positive or in the right half plane)
2) Even if the system is stable such that $x \to 0$, can't dictate the response to zero (i.e. overshoot, settle time, etc.

Introduce State Feedback $\dot{x} = Ax + Bu$

- Instead of making u=0, set: $u = -Kx$
  - This is setting the reference vector (R) to zero

$$u = K(R - x) = -Kx$$

    - Known as regulator design (R=0)
    - Note K is a matrix of size Lxn
    - For a single input system, K is a vector of size 1xn

  - Substituting this into the state space equation:

$$\dot{x} = Ax + B(-Kx)$$

  - Results in:

$$\dot{x} = (A - BK)x = A_{CL}x$$

  - Where the new ("closed loop") A is:

$$A_{CL} = A - BK$$

Introduce State Feedback

- New closed loop differential equation is again homogeneous $\dot{x} = (A - BK)x$

$$\dot{x} = (A_{CL})x$$

  – However, now I have the ability to make A_{CL} behave like I want
    - I can set K to make A_{CL}:
      – Ensure stability
      – Set specifications (bandwidth, overshoot, rise time, etc).
        » All your 3140 knowledge is still useful with state space control design
  – Requires that I have access (knowledge of) all the states (i.e. entire *x* vector)
    - Called “Full State” Feedback

Solving for K

- Recall how to solve for the eigenvalues from state space representation:

$$eig(A) \Rightarrow \det(sI - A) = 0$$

  – Or in matlab:
    - >> eig(A)
- So now we substitute

$$eig(A_{CL}) \Rightarrow \det(sI - A_{CL}) = 0$$

  – where

$$A_{CL} = A - BK$$

Solving for K

- So basically, we solve for K such that:
  $$\det\{sI-(A-BK)\} = desired\ C.E.$$
  - Sound familiar?
    - Its essentially coefficient matching
  - Where does the desired characteristic equation (C.E.) come from
    - Same place as 3140 – from desired eigenvalue specs
      - bandwidth, overshoot, settle time, rise time, etc.
  - For 1st or 2nd order single input systems, solving for K (scalar or 1x2 vector) is simple algebra
  - Or use matlab
    - >>K=place(A,B,s_des)

Controllability

- Is it always possible to solve for K such that the roots of $\det\{sI-(A-BK)\}=0$ match our desired eigenvalues (s_des)?
  - It is if the system is “controllable”
- A system being controllable means that it is possible to find a matrix K such that the eigenvalues can be placed anywhere I choose.
  - Note the yaw dynamics are not controllable if the front tires saturate
    - But it is stable, so although can’t set the response of A-BK (i.e. the controlled system), the yaw rate will settle to zero on its own.

Determining Controllability

- Controllability is determined by checking the rank of the controllability Matrix:

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

  - *C* will be nxn
  - If *C* is full rank (rank=n), then the system is said to be "controllable"
    - Full rank means all the rows of *C* are independent
    - Again, this simply means that there ***exists*** a matrix K such that the eig(A-BK)=s_{des}
  - Matlab will form this matrix for you:

```
>>C=ctrb(A,B)
>>rank(C)
```

Additional Comments on Controllability

- Additionally, a non-controllable doesn't mean you can't control the system. It simply means you can't move the eigenvalues anywhere you want.
  - In the case of the vehicle with $C\alpha f=0$, notice that the input vector B=0. So in this case, you can't move the eigenvalues at all.
  - However in some cases, it may be possible to move the eigenvalues and control the system, you just can't place them anywhere.
    - This is similar to a proportional controller on a 2nd order system:

$$\ddot{x} + \dot{x} + K_p = 0$$

    - This system can be controlled, however there is only a limited subset of closed-loop eigenvalues (think root locus). There are not enough degrees of freedom to set the eigenvalues anywhere (this requires PD control).
- Matlab will give an error if the system is uncontrollable:
  - "Can't place the eigenvalues there"

Introducing a Reference

- In state space you can't just say:

$$u = K(R - x)$$

- Why not:
  1) You don't know what steady state u (u_{ss}) is needed to maintain the desired state
     - This is the because the closed loop G_{DC} may not be one
     - Its basically the same as the need for pre-reference scaling in classical (3140) control
  2) The reference (or desired state) *R* is a ***vector*** and you may not easily know what you want all the states to be
     - For some systems this is easy (if I am trying to control pendulum to 10 degrees, then I know I want $\theta_{des} = 10\ deg\ \&\ \ \dot{\theta}_{des} = 0$ or *R*=[10 0]
     - But think about the vehicle dynamic example. If you want the yaw rate to be 10 rad/sec, what should β_{des}=? since *R*=[r_{des} β_{des}]

Introducing a Reference

- So you must solve for what the reference should be for the other states and what the steady state input (u_{ss}) needs to be to hold that reference.
- Note that it is easy to solve for the steady state output in state space.
- Given the state space equations:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

- Simply set $\dot{x} = 0$ to solve for the steady state x:

$$0 = Ax_{ss} + Bu_{ss}$$
$$x_{ss} = -A^{-1}Bu_{ss}$$
$$y_{ss} = Cx_{ss} + Du_{ss}$$

Introducing a Reference

- y_{ss} sets which output (i.e. which state within the state vector x you are want as your desired state or desired reference *d*)
- So given:

$$0 = Ax_{ss} + Bu_{ss}$$

$$r = y_{ss} = Cx_{ss} + Du_{ss}$$

- So we now define the reference (the desired state) and steady state input to hold that reference as:

$$x_{ss} = N_x d$$

$$u_{ss} = N_u d$$

- These can now be placed into a coupled set of equations (written in matrix form):

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- Note: I switch to using "*d*" as the reference state since "*r*" is yaw rate in vehicle dynamics.

Introducing a Reference

- We can then solve the previous coupled equations by:

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- The reference is then utilized with the following equation:

$$u = N_u d - K(x - N_x d)$$

- Or written another way:

$$u = -Kx + (N_u + KN_x)d$$

- Substituting this input into the state space equations:

$$\dot{x} = Ax + Bu$$

$$\dot{x} = Ax + B\{-Kx + (N_u + KN_x)d\}$$

$$\dot{x} = (A - BK)x + (BN_u + BKN_x)d$$

Introducing a Reference

- So you must solve for what the reference should be for the other states and what the steady state input (u_{ss}) needs to be to hold that reference

$$\dot{x} = (A - BK)x + (BN_u + BKN_x)d$$

- Note the ***homogeneous*** solution does not change. It is still governed by

$$\dot{x} = (A - BK)x$$

  – Therefore the eigenvalues have not changed

- However, the particular solution has now changed such that

$$y = Cx_{ss} \rightarrow d$$

- Also note that:

$$x_{des} = R = N_x d$$

Adding Integral Control is SS

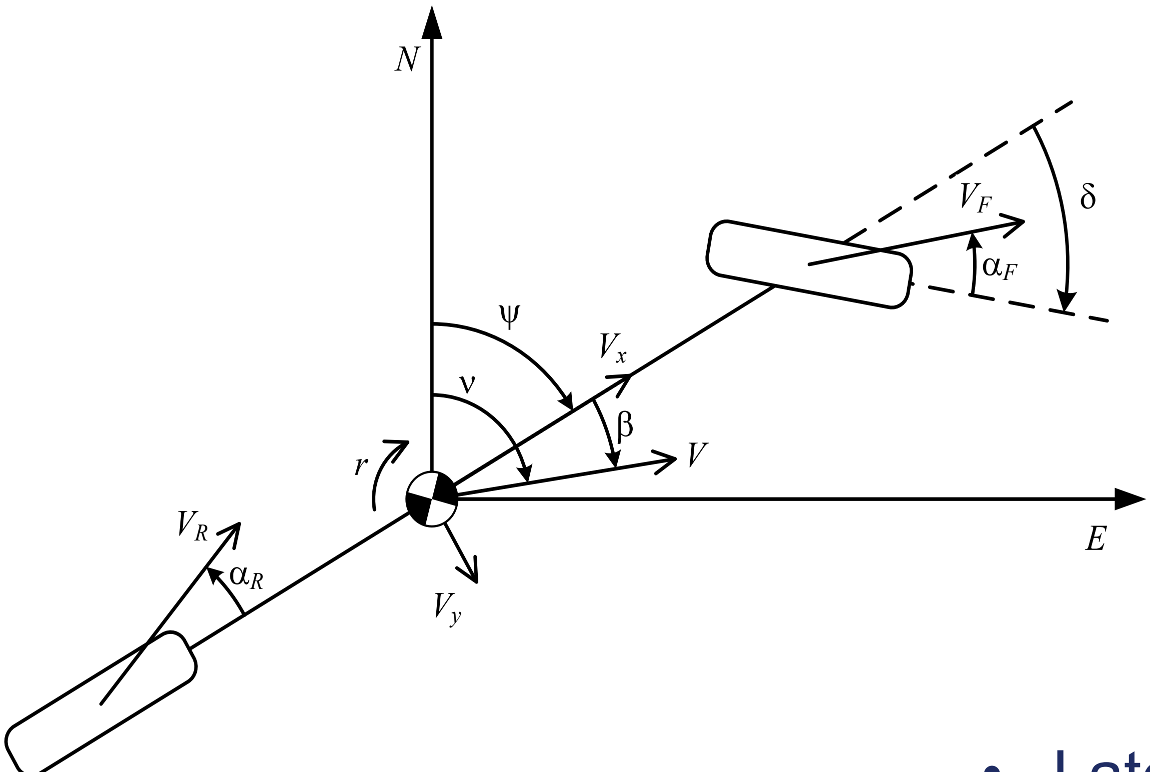
- To add integral control, simply augment the state space model by adding a new state that is the integral of the state:

$$\begin{bmatrix} \dot{x} \\ x \end{bmatrix} = \begin{bmatrix} A & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \int x \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u$$

- Note that the above new A matrix will have the same eigenvalues as the original A plus one additional eigenvalue at (you guessed it) the origin.
- The control input now becomes:

$$u = -\begin{bmatrix} K & K_I \end{bmatrix} \begin{bmatrix} x \\ \int x \end{bmatrix}$$

Augmenting the Yaw Dynamic States



V = velocity

r = yaw rate = $\dot{\psi}$

ψ = heading (or yaw)

ν = vehicle course

δ = steer angle

β = body sideslip angle

α = tire sideslip angle

$$\begin{bmatrix} \dot{V}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{-c_0}{mV} & \frac{-c_1}{mV} - V \\ \frac{-c_1}{I_z V} & \frac{-c_2}{I_z V} \end{bmatrix} \begin{bmatrix} V_y \\ r \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha f}}{m} \\ \frac{a C_{\alpha f}}{I_z} \end{bmatrix} \delta$$

- Lateral velocity is defined as:

$$\dot{y} = V sin(\psi + \beta)$$
$$= V_x \sin(\psi) + V_y \cos(\psi)$$

- Assuming small angles:

$$\dot{y} \approx V_x \psi + V_y$$

Adding Heading ($\dot{\psi} = r$)

$$\begin{bmatrix} \dot{V}_y \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{-c_0}{mV} & \frac{-c_1}{mV} - V & 0 \\ \frac{-c_1}{I_z V} & \frac{-c_2}{I_z V} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_y \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha f}}{m} \\ \frac{aC_{\alpha f}}{I_z} \\ 0 \end{bmatrix} \delta$$

- Now 3rd order, so 3 eigenvalues:
  - 2 from the bicycle model (unchanged) and 1 pure integrator (same as with transfer functions)
- The reference is now heading:

$$d = \psi = y_{ss} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_y \\ r \\ \psi \end{bmatrix}$$

Adding Lateral Position ($\dot{y} = V_x\psi + V_y$)

$$\begin{bmatrix} \dot{V}_y \\ \dot{r} \\ \dot{\psi} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{-c_0}{mV} & \frac{-c_1}{mV} - V & 0 & 0 \\ \frac{-c_1}{I_z V} & \frac{-c_2}{I_z V} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & V_x & 0 \end{bmatrix} \begin{bmatrix} V_y \\ r \\ \psi \\ y \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha f}}{m} \\ \frac{aC_{\alpha f}}{I_z} \\ 0 \\ 0 \end{bmatrix} \delta$$

- Now 4th order, so 4 eigenvalues:
  - 2 from the bicycle model (unchanged) and 2 integrators (one for heading and one for lateral position)
- Note: You don't have to treat the lateral velocity (V_y) as a disturbance like you did with transfer functions
  - It can be included directly in the state space format!

Adding Lateral Position ($\dot{y} = V_x\psi + V_y$)

$$\begin{bmatrix} \dot{V}_y \\ \dot{r} \\ \dot{\psi} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{-c_0}{mV} & \frac{-c_1}{mV} - V & 0 & 0 \\ \frac{-c_1}{I_z V} & \frac{-c_2}{I_z V} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & V_x & 0 \end{bmatrix} \begin{bmatrix} V_y \\ r \\ \psi \\ y \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha f}}{m} \\ \frac{aC_{\alpha f}}{I_z} \\ 0 \\ 0 \end{bmatrix} \delta$$

- The reference is now lateral position:

$$d = y = y_{ss} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_y \\ r \\ \psi \\ y \end{bmatrix}$$

Simple First Order Example

Given the model and control input:

$$\dot{x} = -1x + F$$

$$F = -Kx$$

Solving for the closed loop equation of motion:

$$\dot{x} = -(1 + K)x$$

Solving for the closed loop characteristic equation:

$$\det\{s + (1 + K)\} = 0$$

$$s + (1 + K) = 0$$

Solving for the control gain to place the closed loop eigenvalue at -10:

$$s_{des} = -10$$

$$K = 9$$

So the state space control becomes:

$$F = -9x$$

Simple First Order Example

- In Matlab:

```
>>A=-1;B=1;
>>eig(A)
>>K=place(A,B,-10)
>>eig(A-BK)
```

$$K = 9$$

2nd Order (Pendulum) Example

Lets consider the linearized pendulum model:

$$J\ddot{\theta} + b\dot{\theta} + mgl\theta = \tau$$

Picking some values for the pendulum and then placing into state space

$$\ddot{\theta} + 1\dot{\theta} + 25\theta = \tau$$

$$\begin{bmatrix} \ddot{\theta} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -1 & -25 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tau$$

Then using the state space regulator of the form *u=-Kx*

$$\tau = -K \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix} = -[K_1 \quad K_2] \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix}$$

Solving for the closed loop state space equations

$$\begin{bmatrix} \ddot{\theta} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -1 & -25 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix} - [K_1 \quad K_2] \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix} = \begin{bmatrix} -1 - K_1 & -25 - K_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix}$$

2^{nd} Order (Pendulum) Example

Solving for the closed loop characteristic equation:

$$\det\left\{sI - \begin{bmatrix} -1-K_1 & -25-K_2 \\ 1 & 0 \end{bmatrix}\right\} = \det\left\{\begin{bmatrix} s+1+K_1 & 25+K_2 \\ -1 & s \end{bmatrix}\right\} = 0$$

$$s^2 + (1+K_1)s + (25+K_2) = 0$$

Choosing a desired closed loop response of

$$\omega_n = 14.14 \;\; and \;\; \xi = 0.707 \quad \text{or} \quad s = -10 \pm 10j$$

Results in the following desired characteristic equation

$$s^2 + 20s + 200 = 0$$

Then solving for the control gains:

$$K_1 = 19$$
$$K_2 = 175 \quad \text{or} \quad K = \begin{bmatrix} 19 & 175 \end{bmatrix}$$

Simple First Order Example

- In Matlab:

```
>>A=[-1 -25;1 0];B=[1 ; 0];
>>eig(A)
>>K=place(A,B,[-10+10j,-10-10j])
>>eig(A-BK)
```

$$K = [19 \quad 175]$$

$$\tau = -[19 \quad 175]\begin{bmatrix}\dot{\theta} \\ \theta\end{bmatrix} = -19\dot{\theta} - 175\theta$$

2^{nd} Order (Pendulum) Example

- Note that K_1 is essentially K_d and K_2 is essentially K_p
  - Get the exact same values if you designed a PD controller for the pendulum using classical (3140) methods
  - However for the vehicle, it's a little different, since the states are not yaw rate and yaw acceleration (K_1 will be a lateral velocity gain and K_2 will be a yaw rate gain)
    - Although you can recast the bicycle model transfer function into a state space with the states of yaw rate and yaw acceleration.

State Space Control in Discrete

- State space control in discrete is incredibly convenient (and essentially the same as continuous)
- Discretize your state space model

```
>>[Ad,Bd,Cd,Dd]=c2dm(A,B,C,D,dt,'zoh');
```

- Convert your desired eigenvalues from continuous to discrete:

$$z_{des} = e^{s_{des}\Delta t}$$

- The rest is the same
  - Solve for discrete control gain matrix to place the eigenvalues of $(A_D\text{-}B_DK_D)$ at z_{des}
  - Same command in Matlab

```
>>Kd=place(Ad,Bd,Zdes)
```

Discrete State Space Control

- Discrete State Space Model

$$x_{k+1} = A_D x_k + B_D u_k$$

- Inserting controller:

$$u_k = -K_D x_k$$

- Results in

$$x_{k+1} = A_D x_k + B_D(-K_D x)$$

$$x_{k+1} = (A_D - B_D K_D) x_k$$

- Closed-looped discrete characteristic equation

$$\det\{zI - (A_D - B_D K_D)\} = 0$$

Simple 1st order example in Discrete

- Lets look at the exact same first order as we did previously in continuous:

$$\dot{x} = -1x + F$$

- First we will determine discrete state model using a sample rate of 0.01 seconds using Matlab which results in:

$$x_{k+1} = 0.99x_k + 0.01F_k$$

- Then we need to find the desired closed loop eigenvalue in discrete:

$$s_{des} = -10$$

$$z_{des} = e^{(-10)(0.01)} = 0.905$$

Simple First Order Example

Given the discrete model and control input:

$$x_{k+1} = 0.99x_k + 0.01F_k$$
$$F_k = -K_D x_k$$

Solving for the closed loop equation of motion:

$$x_{k+1} = (0.99 - 0.01K_D)x_k$$

Solving for the closed loop characteristic equation:

$$\det\{z - (0.99 - 0.01K_D)\} = 0$$
$$z - (0.99 - 0.01K_D) = 0$$

Solving for the control gain to place the discrete closed loop eigenvalue at 0.9048:

$$z_{des} = 0.9048 = (0.99 - 0.01K_D)$$
$$K_D = 8.62$$

So the state space control becomes:

$$F_k = -8.62x_k$$

Simple 1st order example in Discrete

- In Matlab

```
>>sdes=-10; dt=0.01;
>>A=-1;B=1;C=1;D=0;
>>[Ad,Cd,Dd,Dd]=c2dm(A,B,C,D,dt,'zoh');
>>zdes=exp(sdes*dt)
>>Kd=place(Ad,Bd,zdes)
```

$K_D = 8.5639$ (Note the value is different due to rounding)

$F_k = -8.5639 x_k$ (Control implementation)

$x_{k+1} = 0.99 x_k + 0.01 F_k$ (Discrete simulation)

Introducing a Reference in Discrete

- It is easy to solve for the steady state output in discrete state space.
  - The steady state *x* (and *y*) will be the same in continuous and discrete
- Given the state space equations: $$x_{k+1} = A_D x_k + B_D u_k$$ $$y_k = C_D x_k + D_D u_k$$
- Simply set $x_{k+1} = x_k$ to solve for the steady state x:

$$x_k = A_D x_k + B_D u_k$$

$$x_{ss} = (I - A_D)^{-1} B u_{ss}$$
$$y_{ss} = C_D x_{ss} + D_D u_{ss}$$

- Note that x_{ss} does not change from continuous to discrete!

Introducing a Reference in Discrete

- Since the steady state x (and y and u) are exactly the same in discrete (as continuous), the discrete reference is exactly the same as continuous. In fact you can simply solve for N_x and N_u from the continuous domain:

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} (I - A_D) & B_D \\ C_D & D_D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- The reference is then utilized in discrete with the following equation:

$$u_k = N_u d - K_D(x_k - N_x d)$$

- Or written another way:

$$u = -K_D x_k + (N_u + K_D N_x)d$$

- And (just as in continuous), the reference does not change the closed loop eigenvalue.