Space feddback control

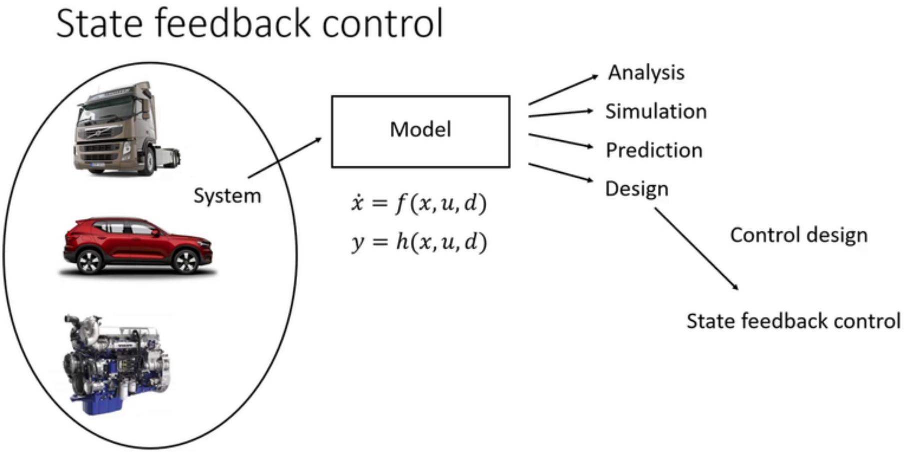
Dr. Ing. Rodrigo Gonzalez

rodralez@ingenieria.uncu.edu.ar

Control y Sistemas

Facultad de Ingeniería,
Universidad Nacional de Cuyo

State feedback control



State feedback control

Consider a linear time-invariant state-space model given by:

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

where $x(t) \in \mathbb{R}^n$ is the state (vector), $u(t) \in \mathbb{R}^p$ is the input or control signal and $y(t) \in \mathbb{R}^q$ is the output signal. (For SISO case, $p = 1, q = 1$)

State feedback control

Consider a linear time-invariant state-space model given by:

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

where $x(t) \in \mathbb{R}^n$ is the state (vector), $u(t) \in \mathbb{R}^p$ is the input or control signal and $y(t) \in \mathbb{R}^q$ is the output signal. (For SISO case, $p = 1, q = 1$)

The system poles are given by the eigenvalues of the system matrix $A \in \mathbb{R}^{n \times n}$.

State feedback control

Idea with control design: Modify the eigenvalues of A by using the input $u(t)$

State feedback control

Idea with control design: Modify the eigenvalues of A by using the input $u(t)$

State feedback controller: $u(t) = -Kx(t) + k_r r(t)$

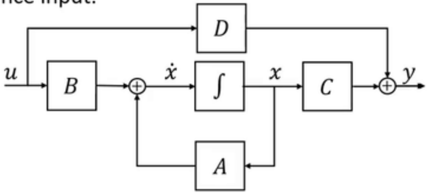
where $K \in \mathbb{R}^{p \times n}$ is the feedback gain, $k_r \in \mathbb{R}^{p \times r}$ is the steady-state reference gain and $r(t) \in \mathbb{R}^r$ is the reference input.

State feedback control

***Idea with control design*:** Modify the eigenvalues of A by using the input $u(t)$

***State feedback controller*:** $u(t) = -Kx(t) + k_r r(t)$

where $K \in \mathbb{R}^{p\times n}$ is the feedback gain, $k_r \in \mathbb{R}^{p\times r}$ is the steady-state reference gain and $r(t) \in \mathbb{R}^r$ is the reference input.

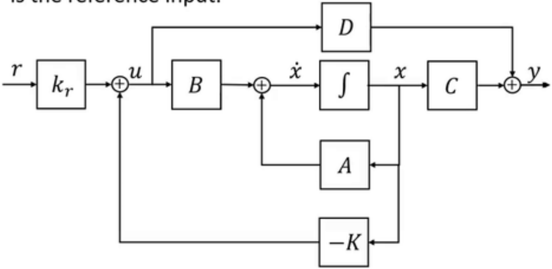


State feedback control

***Idea with control design*:** Modify the eigenvalues of A by using the input $u(t)$

State feedback controller: $u(t) = -Kx(t) + k_r r(t)$

where $K \in \mathbb{R}^{p\times n}$ is the feedback gain, $k_r \in \mathbb{R}^{p\times r}$ is the steady-state reference gain and $r(t) \in \mathbb{R}^r$ is the reference input.

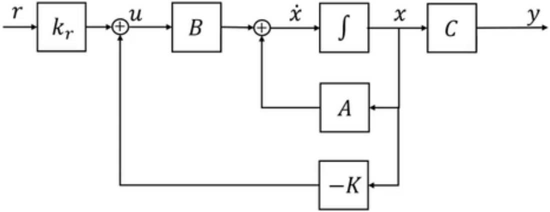


State feedback control

Idea with control design: Modify the eigenvalues of A by using the input $u(t)$

State feedback controller: $u(t) = -Kx(t) + k_r r(t)$

where $K \in \mathbb{R}^{p\times n}$ is the feedback gain, $k_r \in \mathbb{R}^{p\times r}$ is the steady-state reference gain and $r(t) \in \mathbb{R}^r$ is the reference input.



State feedback control

Idea with control design: Modify the eigenvalues of A by using the input $u(t)$

State feedback controller: $u(t) = -Kx(t) + k_r r(t)$

where $K \in \mathbb{R}^{p\times n}$ is the feedback gain, $k_r \in \mathbb{R}^{p\times r}$ is the steady-state reference gain and $r(t) \in \mathbb{R}^r$ is the reference input.

Using the state feedback controller the closed loop dynamics becomes:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + B(-Kx(t) + k_r r(t)) \\ &= (A - BK)x(t) + Bk_r r(t)\end{aligned}$$

State feedback control

Idea with control design: Modify the eigenvalues of A by using the input $u(t)$

State feedback controller: $u(t) = -Kx(t) + k_r r(t)$

where $K \in \mathbb{R}^{p \times n}$ is the feedback gain, $k_r \in \mathbb{R}^{p \times r}$ is the steady-state reference gain and $r(t) \in \mathbb{R}^r$ is the reference input.

Using the state feedback controller the closed loop dynamics becomes:

$$\dot{x}(t) = Ax(t) + B(-Kx(t) + k_r r(t))$$
$$= (A - BK)x(t) + Bk_r r(t)$$

Control objective: Choose K such that the closed loop dynamics $A - BK$ get desired properties, i.e fulfill the specifications or stabilize the system.

State feedback control

Idea with control design: Modify the eigenvalues of A by using the input $u(t)$

State feedback controller: $u(t) = -Kx(t) + k_r r(t)$

where $K \in \mathbb{R}^{p \times n}$ is the feedback gain, $k_r \in \mathbb{R}^{p \times r}$ is the steady-state reference gain and $r(t) \in \mathbb{R}^r$ is the reference input.

Using the state feedback controller the closed loop dynamics becomes:

$$\dot{x}(t) = Ax(t) + B(-Kx(t) + k_r r(t))$$
$$= (A - BK)x(t) + Bk_r r(t)$$

Control objective: Choose K such that the closed loop dynamics $A - BK$ get desired properties, i.e fulfill the specifications or stabilize the system.

SISO case: *n parameters in K and n eigenvalues in A, so it might be possible!*

Reference tracking

The steady-state reference gain, k_r, does not affect the stability, but it does affect the steady-state solution.

Reference tracking

The steady-state reference gain, k_r, does not affect the stability, but it does affect the steady-state solution.

The steady-state gain is usually chosen such that:

$$y(t) \approx r(t) \;\; as \; t \to \infty$$

Reference tracking

The steady-state reference gain, k_r, does not affect the stability, but it does affect the steady-state solution.

The steady-state gain is usually chosen such that:

$$y(t) \approx r(t) \;\; as\, t \to \infty$$

At steady-state the time derivative of the state variable is $\dot{x}\,(t) \equiv 0$, so

$$0 = (A - BK)x(t) + Bk_r r(t)$$

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

Reference tracking

The steady-state reference gain, k_r, does not affect the stability, but it does affect the steady-state solution.

The steady-state gain is usually chosen such that:

$$y(t) \approx r(t) \ \ as\, t \to \infty$$

At steady-state the time derivative of the state variable is $\dot{x}\,(t) \equiv 0$, so

$$0 = (A - BK)x(t) + Bk_r r(t)$$

$$y(t) = Cx(t)$$

Reference tracking

The steady-state reference gain, k_r, does not affect the stability, but it does affect the steady-state solution.

The steady-state gain is usually chosen such that:

$$y(t) \approx r(t) \;\; as\; t \to \infty$$

At steady-state the time derivative of the state variable is $\dot{x}\,(t) \equiv 0$, so

$$\left.\begin{aligned} 0 &= (A - BK)x(t) + Bk_r r(t) \\ y(t) &= Cx(t) \end{aligned}\right\} \qquad y = -C(A - BK)^{-1}Bk_r r$$

Reference tracking

The steady-state reference gain, k_r, does not affect the stability, but it does affect the steady-state solution.

The steady-state gain is usually chosen such that:

$$y(t) \approx r(t) \ \ as\, t \to \infty$$

At steady-state the time derivative of the state variable is $\dot{x}(t) \equiv 0$, so

$$\left. \begin{aligned} 0 &= (A - BK)x(t) + Bk_r r(t) \\ y(t) &= Cx(t) \end{aligned} \right\} \quad y = -C(A - BK)^{-1} B k_r r$$

If $y(t) \approx r(t) \ \ as\, t \to \infty$, then k_r should be chosen as

$$k_r = -(C(A - BK)^{-1}B)^{-1} \quad \text{or} \quad k_r = -1/C(A - BK)^{-1}B$$

Integral action

Using the steady-state feedback gain, k_{r}, can achieve zero steady-state error, but it does depend on the model parameters, as

$$k_{r} = -(C(A-BK)^{-1}B)^{-1} \quad \text{or} \quad k_{r} = -1/C(A-BK)^{-1}B$$

Integral action

Using the steady-state feedback gain, k_r, can achieve zero steady-state error, but it does depend on the model parameters, as

$$k_r = -(C(A-BK)^{-1}B)^{-1} \quad \text{or} \quad k_r = -1/C(A-BK)^{-1}B$$

Introduce ***integral action*** to remove the steady-state error. Approach: introduce an additional state variable in our system which computes the integral of the error

$$\dot{z}(t) = y(t) - r(t)$$

Integral action

Using the steady-state feedback gain, k_{r}, can achieve zero steady-state error, but it does depend on the model parameters, as

$$k_{r} = -(C(A-BK)^{-1}B)^{-1} \quad \text{or} \quad k_{r} = -1/C(A-BK)^{-1}B$$

Introduce ***integral action*** to remove the steady-state error. Approach: introduce an additional state variable in our system which computes the integral of the error

$$\dot{z}\,(t) = y(t) - r(t)$$

The new state-space model becomes:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} Ax - Bu \\ y - r \end{bmatrix} = \begin{bmatrix} Ax - Bu \\ Cx - r \end{bmatrix}$$

Integral action

Using the steady-state feedback gain, k_{r}, can achieve zero steady-state error, but it does depend on the model parameters, as

$$k_r = -(C(A-BK)^{-1}B)^{-1} \quad \text{or} \quad k_r = -1/C(A-BK)^{-1}B$$

Introduce ***integral action*** to remove the steady-state error. Approach: introduce an additional state variable in our system which computes the integral of the error

$$\dot{z}(t) = y(t) - r(t)$$

The new state-space model becomes:

$$\begin{bmatrix}\dot{x}\\ \dot{z}\end{bmatrix} = \begin{bmatrix}Ax-Bu\\ y-r\end{bmatrix} = \begin{bmatrix}Ax-Bu\\ Cx-r\end{bmatrix} = \begin{bmatrix}A & 0\\ C & 0\end{bmatrix}\begin{bmatrix}x\\ z\end{bmatrix} + \begin{bmatrix}B\\ 0\end{bmatrix}u + \begin{bmatrix}0\\ -1\end{bmatrix}r$$

Integral action

Using the steady-state feedback gain, k_{r}, can achieve zero steady-state error, but it does depend on the model parameters, as

$$k_{r} = -(C(A-BK)^{-1}B)^{-1} \quad \text{or} \quad k_{r} = -1/C(A-BK)^{-1}B$$

Introduce ***integral action*** to remove the steady-state error. Approach: introduce an additional state variable in our system which computes the integral of the error

$$\dot{z}(t) = y(t) - r(t)$$

The new state-space model becomes:

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} Ax - Bu \\ y - r \end{bmatrix} = \begin{bmatrix} Ax - Bu \\ Cx - r \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ -1 \end{bmatrix} r$$

Given the new state-space model, we design a controller in the usual fashion and the resulting controller becomes:

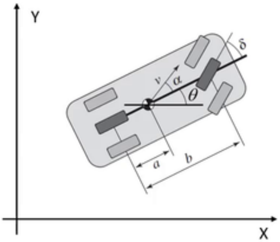
$$u(t) = -Kx(t) - K_I z(t) + k_r r(t)$$

Example 1 - Vehicle steering (Ex 7.4)

Consider the following system:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & v_0\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}av_0/b\\ v_0/b\end{bmatrix}u$$
$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + [0]\,u$$

where x_1 is the lateral position Y , x_2 is the heading orientation θ and u is the steering angle δ.



Vehicle data: $v_0 = 12\ m/s$
$a = 2\ m$
$b = 4\ m$

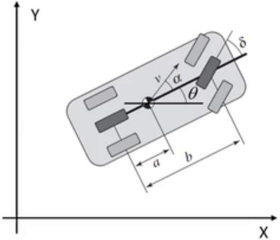
Example 1 - Vehicle steering (Ex 7.4)

Consider the following system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & v_0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} av_0/b \\ v_0/b \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]\, u$$

where x_1 is the lateral position Y , x_2 is the heading orientation θ and u is the steering angle δ.



The idea is to design a controller that ***stabilizes*** the dynamics and ***tracks*** a given lateral position of the vehicle.

Vehicle data: $v_0 = 12\, m/s$
$a = 2\, m$
$b = 4\, m$

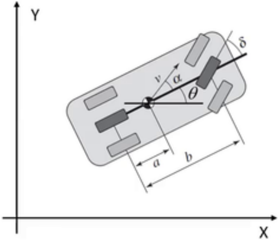
Example 1 - Vehicle steering (Ex 7.4)

Consider the following system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & v_0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} av_0/b \\ v_0/b \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]\, u$$

where x_1 is the lateral position Y , x_2 is the heading orientation θ and u is the steering angle δ.



The idea is to design a controller that ***stabilizes*** the dynamics and ***tracks*** a given lateral position of the vehicle.

Vehicle data: $v_0 = 12\, m/s$
$a = 2\, m$
$b = 4\, m$

Specification: Desired characteristic polynomial:

$$p_{des}(\lambda) = \lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2$$

Example 1 - Vehicle steering (Ex 7.4)

State feedback control:

$$u = -Kx + k_r r = -k_1 x_1 - k_2 x_2 + k_r r$$

Example 1 - Vehicle steering (Ex 7.4)

State feedback control:

$$u = -Kx + k_r r = -k_1 x_1 - k_2 x_2 + k_r r$$

The closed loop system dynamics becomes

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

Example 1 - Vehicle steering (Ex 7.4)

State feedback control:

$$u = -Kx + k_r r = -k_1 x_1 - k_2 x_2 + k_r r$$

The closed loop system dynamics becomes

$$\dot{x} = (A - BK)x + Bk_r r = \left(\begin{bmatrix} 0 & v_0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} k_1 a v_0/b & k_2 a v_0/b \\ k_1 v_0/b & k_2 v_0/b \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} k_r a v_0/b \\ k_r v_0/b \end{bmatrix} r$$

Example 1 - Vehicle steering (Ex 7.4)

State feedback control:

$$u = -Kx + k_r r = -k_1 x_1 - k_2 x_2 + k_r r$$

The closed loop system dynamics becomes

$$\dot{x} = (A - BK)x + Bk_r r = \begin{bmatrix} -k_1 a v_0/b & v_0 - k_2 a v_0/b \\ -k_1 v_0/b & -k_2 v_0/b \end{bmatrix} x + \begin{bmatrix} k_r a v_0/b \\ k_r v_0/b \end{bmatrix} r$$

Example 1 - Vehicle steering (Ex 7.4)

State feedback control:

$$u = -Kx + k_r r = -k_1 x_1 - k_2 x_2 + k_r r$$

The closed loop system dynamics becomes

$$\dot{x} = (A - BK)x + Bk_r r = \begin{bmatrix} -k_1 a v_0/b & v_0 - k_2 a v_0/b \\ -k_1 v_0/b & -k_2 v_0/b \end{bmatrix} x + \begin{bmatrix} k_r a v_0/b \\ k_r v_0/b \end{bmatrix} r$$

$$y = Cx + Du = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Example 1 - Vehicle steering (Ex 7.4)

State feedback control:

$$u = -Kx + k_r r = -k_1 x_1 - k_2 x_2 + k_r r$$

The closed loop system dynamics becomes

$$\dot{x} = (A - BK)x + Bk_r r = \begin{bmatrix} -k_1 a v_0/b & v_0 - k_2 a v_0/b \\ -k_1 v_0/b & -k_2 v_0/b \end{bmatrix} x + \begin{bmatrix} k_r a v_0/b \\ k_r v_0/b \end{bmatrix} r$$

$$y = Cx + Du = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The closed loop system has the characteristic polynomial

$$\det(\lambda I - A + BK) = \cdots = \lambda^2 + \frac{v_0}{b}(ak_1 + k_2)\lambda + \frac{k_1 v_0^2}{b}$$

Example 1 - Vehicle steering (Ex 7.4)

State feedback control:

$$u = -Kx + k_r r = -k_1 x_1 - k_2 x_2 + k_r r$$

The closed loop system dynamics becomes

$$\dot{x} = (A - BK)x + Bk_r r = \begin{bmatrix} -k_1 a v_0/b & v_0 - k_2 a v_0/b \\ -k_1 v_0/b & -k_2 v_0/b \end{bmatrix} x + \begin{bmatrix} k_r a v_0/b \\ k_r v_0/b \end{bmatrix} r$$

$$y = Cx + Du = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The closed loop system has the characteristic polynomial

$$\det(\lambda I - A + BK) = \cdots = \lambda^2 + \frac{v_0}{b}(ak_1 + k_2)\lambda + \frac{k_1 v_0^2}{b}$$

Matching with desired characteristic polynomial gives:

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 \equiv \lambda^2 + \frac{v_0}{b}(ak_1 + k_2)\lambda + \frac{k_1 v_0^2}{b}$$

Example 1 - Vehicle steering (Ex 7.4)

The steady-state gain can be determined:

$$k_r = -1/C(A-BK)^{-1}B = \cdots = k_1 = \frac{b\omega_n^2}{v_0^2}$$

Example 1 - Vehicle steering (Ex 7.4)

The steady-state gain can be determined:

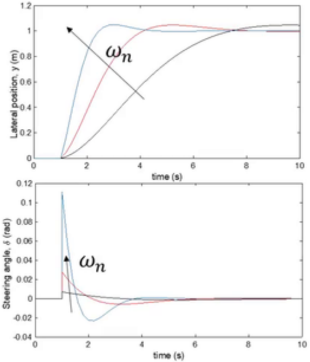
$$k_r = -1/C(A-BK)^{-1}B = \cdots = k_1 = \frac{b\omega_n^2}{v_0^2}$$

Inserting these control design parameters into the feedback controller gives:

$$u = -k_1x_1 - k_2x_2 + k_rr = -\frac{b\omega_n^2}{v_0^2}x_1 - \left(\frac{2\zeta\omega_nb}{v_0} - \frac{ab\omega_n^2}{v_0^2}\right)x_2 + \frac{b\omega_n^2}{v_0^2}r$$

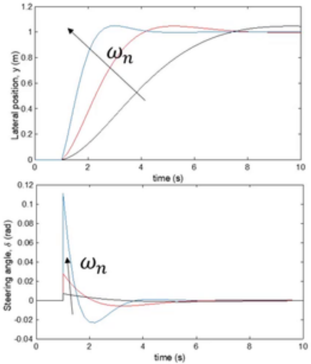
Example 1 - Vehicle steering (Ex 7.4)

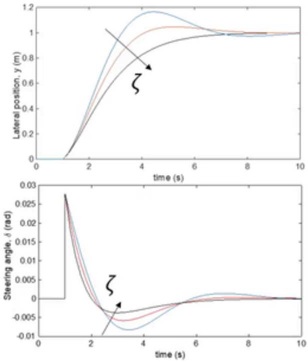
Simulations with different values of ζ and ω_n:



Example 1 - Vehicle steering (Ex 7.4)

Simulations with different values of ζ and ω_n:





Example 2

Consider the following system (similar to Example 1 with one difference):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]\, u$$

Example 2

Consider the following system (similar to Example 1 with one difference):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]\, u$$

Again, the idea is to design a controller that ***stabilizes*** the dynamics and ***tracks*** a given reference signal.

Specification: Desired characteristic polynomial:

$$p_{des}(\lambda) = \lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2$$

State feedback control:

$$u = -Kx + k_r r = -k_1 x_1 - k_2 x_2 + k_r r$$

Example 2

The closed loop system dynamics becomes

$$\dot{x} = (A - BK)x + Bk_r r = \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} k_1 & k_2 \\ 0 & 0 \end{bmatrix}\right)\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} k_r \\ 0 \end{bmatrix} r$$

$$y = Cx + Du = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Example 2

The closed loop system dynamics becomes

$$\dot{x} = (A - BK)x + Bk_r r = \begin{bmatrix} -k_1 & 1 - k_2 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} k_r \\ 0 \end{bmatrix} r$$

$$y = Cx + Du = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Example 2

The closed loop system dynamics becomes

$$\dot{x} = (A - BK)x + Bk_r r = \begin{bmatrix} -k_1 & 1 - k_2 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} k_r \\ 0 \end{bmatrix} r$$

$$y = Cx + Du = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The closed loop system has the characteristic polynomial

$$\det(\lambda I - A + BK) = \det \begin{pmatrix} \lambda + k_1 & k_2 - 1 \\ 0 & \lambda \end{pmatrix} = \lambda^2 + k_1 \lambda = \lambda(\lambda + k_1)$$

Example 2

The closed loop system dynamics becomes

$$\dot{x} = (A - BK)x + Bk_r r = \begin{bmatrix} -k_1 & 1 - k_2 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} k_r \\ 0 \end{bmatrix} r$$

$$y = Cx + Du = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The closed loop system has the characteristic polynomial

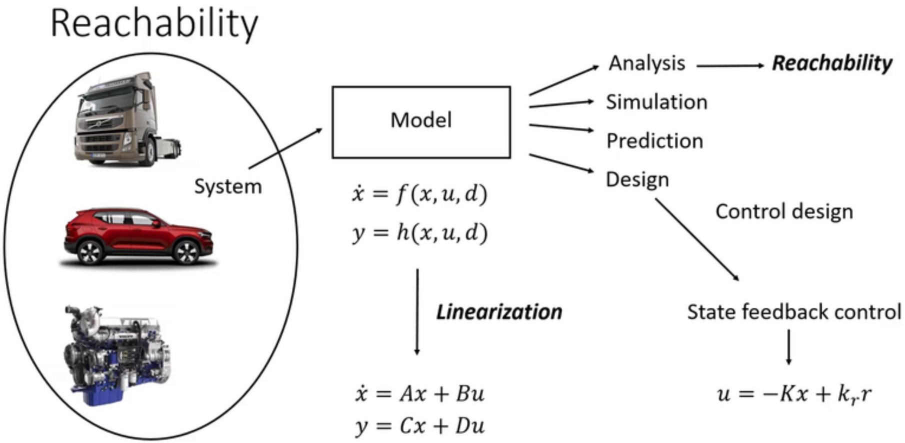
$$\det(\lambda I - A + BK) = \det \begin{pmatrix} \lambda + k_1 & k_2 - 1 \\ 0 & \lambda \end{pmatrix} = \lambda^2 + k_1 \lambda = \lambda(\lambda + k_1)$$

While matching with the desired characteristic polynomial

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 \equiv \lambda^2 + k_1\lambda$$

we notice that it is not possible to make them identical to each other.

It is not possible to control all the eigenvalues (shape the dynamics). We say that the system is not ***controllable*** or not ***reachable***. Furthermore, one eigenvalue is always 0, which means that the closed loop system is unstable.


Reachability
System
Model
$\dot{x} = f(x,u,d)$
$y = h(x,u,d)$
Analysis
Reachability
Simulation
Prediction
Design
Control design
Linearization
State feedback control
$\dot{x} = Ax + Bu$
$y = Cx + Du$
$u = -Kx + k_r r$

Revisit - Example

Consider the following system:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix}u$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + [0]\,u$$

State feedback control: $u = -Kx + k_r r = -k_1 x_1 - k_2 x_2 + k_r r$

The closed loop system dynamics becomes

$$\dot{x} = (A - BK)x + Bk_r r = \begin{bmatrix}-k_1 & 1 - k_2\\ 0 & 0\end{bmatrix}x + \begin{bmatrix}k_r\\ 0\end{bmatrix}r$$

Uncontrollable

and the characteristic polynomial is given as

Controllable

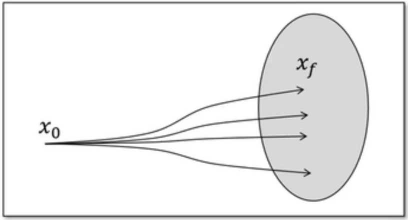
$$\det(\lambda I - A + BK) = \det\begin{pmatrix}\lambda + k_1 & k_2 - 1\\ 0 & \lambda\end{pmatrix} = \lambda^2 + k_1\lambda = \lambda(\lambda + k_1)$$

Reachability

Definition (Reachability): A linear system is ***reachable*** if for any $x_0, x_f \in \mathbb{R}^n$ there exists a $T > 0$ and $u\colon [0,T] \to \mathbb{R}$ such that if $x(0) = x_0$ then the corresponding solution satisfies $x(T) = x_f$.

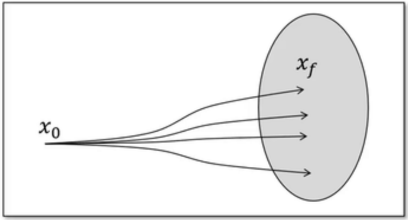
Reachability

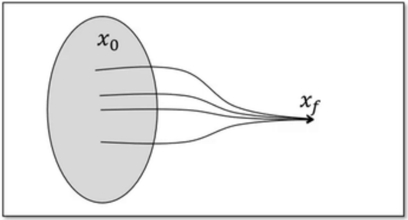
Definition (Reachability): A linear system is ***reachable*** if for any $x_0, x_f \in \mathbb{R}^n$ there exists a $T > 0$ and u: $[0, T] \to \mathbb{R}$ such that if $x(0) = x_0$ then the corresponding solution satisfies $x(T) = x_f$.



Reachability

Definition (Reachability): A linear system is ***reachable*** if for any $x_0, x_f \in \mathbb{R}^n$ there exists a $T > 0$ and u: $[0, T] \rightarrow \mathbb{R}$ such that if $x(0) = x_0$ then the corresponding solution satisfies $x(T) = x_f$.



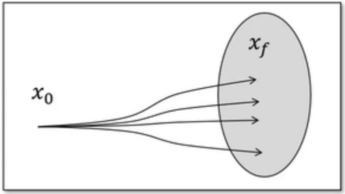


Sometimes the definition of ***controllable*** and ***controllability*** is used, and that is similar.

Linear system representations

$$
\begin{tikzcd}
a\left(\frac{d}{dt}\right)y(t) = b\left(\frac{d}{dt}\right)u(t) \arrow[r, "\mathcal{L}"] & a(s)Y(s) = b(s)U(s) \arrow[d, "G(s) = \frac{b(s)}{a(s)}"] \\
y(t) = \int_0^t g(t-\tau)\,u(\tau)d\tau & Y(s) = G(s)U(s) \arrow[l, "\mathcal{L}^{-1}"'] \\
\begin{gathered} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{gathered} \arrow[u, "g(t) = Ce^{At}B + D\delta(t)"'] \arrow[r, "\mathcal{L}"] & \begin{gathered} sX(s) - x(0) = AX(s) + BU(s) \\ Y(s) = CX(s) + DU(s) \end{gathered} \arrow[u, "G(s) = C(sI-A)^{-1}B + D"']
\end{tikzcd}
$$

Reachability



To see that an arbitrary point can be reached, we can use the convolution equation.

Assume that the system starts from zero, the state of a linear system is given by:

$$x(t) = \int_0^t e^{A(t-\tau)} B\, u(\tau) d\tau = \int_0^t e^{A\tau} B\, u(t-\tau) d\tau$$

From linear theory it can be shown that

$$e^{A\tau} = I\alpha_0(\tau) + A\alpha_1(\tau) + \cdots + A^{n-1}\alpha_{n-1}(\tau)$$

where $\alpha_i(t)$ are scalar functions, so that

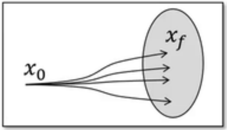
$$x(t) = B \int_0^t \alpha_0(\tau)\, u(t-\tau) d\tau + AB \int_0^t \alpha_1(\tau)\, u(t-\tau) d\tau + \cdots + A^{n-1} B \int_0^t \alpha_{n-1}(\tau)\, u(t-\tau) d\tau$$

Reachability

By writing it in "vector form", we get:

$$x(t) = \underbrace{[B \quad AB \quad \dots \quad A^{n-1}B]}_{W_r} \begin{bmatrix} \int_0^t \alpha_0(\tau)\, u(t-\tau)d\tau \\ \int_0^t \alpha_1(\tau)\, u(t-\tau)d\tau \\ \vdots \\ \int_0^t \alpha_{n-1}(\tau)\, u(t-\tau)d\tau \end{bmatrix}$$

Reachability

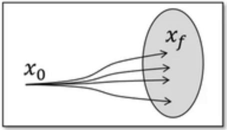


By writing it in "vector form", we get:

$$x(t) = \underbrace{[B \quad AB \quad \dots \quad A^{n-1}B]}_{W_r} \begin{bmatrix} \int_0^t \alpha_0(\tau)\, u(t-\tau) d\tau \\ \int_0^t \alpha_1(\tau)\, u(t-\tau) d\tau \\ \vdots \\ \int_0^t \alpha_{n-1}(\tau)\, u(t-\tau) d\tau \end{bmatrix}$$

To reach an arbitrary point is state-space, we require that W_r is nonsingular. The matrix W_r is called the ***reachability matrix***.

Reachability



By writing it in "vector form", we get:

$$x(t) = \underbrace{[B \quad AB \quad \dots \quad A^{n-1}B]}_{W_r} \begin{bmatrix} \int_0^t \alpha_0(\tau)\, u(t-\tau) d\tau \\ \int_0^t \alpha_1(\tau)\, u(t-\tau) d\tau \\ \vdots \\ \int_0^t \alpha_{n-1}(\tau)\, u(t-\tau) d\tau \end{bmatrix}$$

To reach an arbitrary point is state-space, we require that W_r is nonsingular. The matrix W_r is called the ***reachability matrix***.

Theorem (Reachability rank condition): *A linear system is reachable if and only if the reachability matrix is invertable (has full rank).*

Revisit - Example

Return to our example, with the following system:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}1\\ 0\end{bmatrix}u$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + [0]\,u$$

Reachability matrix:

$$W_r = \begin{bmatrix}B & AB\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}$$

Revisit - Example

Return to our example, with the following system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]\, u$$

Reachability matrix:

$$W_r = [B \quad AB] = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Note: A square matrix M ($n \times n$) has full rank n iff the det(M)≠0

Compute the determinant:

Revisit - Example

Return to our example, with the following system:

$$\begin{bmatrix}\dot{x}_1\\\dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\0 & 0\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} + \begin{bmatrix}1\\0\end{bmatrix}u$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} + [0]\,u$$

Reachability matrix:

$$W_r = [B \quad AB] = \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}$$

Note: A square matrix M (n×n) has full rank n iff the det(M)≠0

Compute the determinant:

$$\det(W_r) = \begin{vmatrix}1 & 0\\0 & 0\end{vmatrix} = 1\cdot 0 - 0\cdot 0 = 0$$

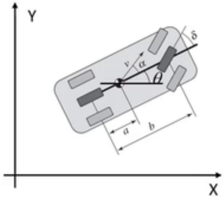
The system is not reachable!

Revisit Example - Vehicle steering (Ex 7.4)

Consider the following system:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & v_0\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}av_0/b\\ v_0/b\end{bmatrix}u$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + [0]\,u$$



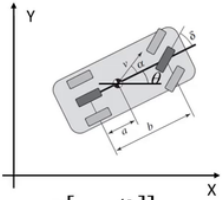
Reachability matrix:

Revisit Example - Vehicle steering (Ex 7.4)

Consider the following system:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & v_0\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}av_0/b\\ v_0/b\end{bmatrix}u$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + [0]\,u$$



Reachability matrix:

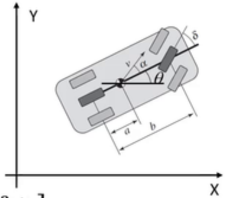
$$W_r = [B \quad AB] = \left[\begin{matrix}av_0/b\\ v_0/b\end{matrix} \quad \begin{bmatrix}0 & v_0\\ 0 & 0\end{bmatrix}\begin{bmatrix}av_0/b\\ v_0/b\end{bmatrix}\right]$$

Revisit Example - Vehicle steering (Ex 7.4)

Consider the following system:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & v_0\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}av_0/b\\ v_0/b\end{bmatrix}u$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + [0]\,u$$



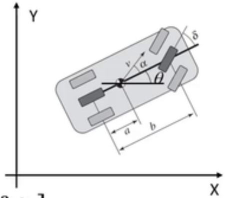
Reachability matrix:

$$W_r = [B \quad AB] = \begin{bmatrix}av_0/b & v_0^2/b\\ v_0/b & 0\end{bmatrix}$$

Revisit Example - Vehicle steering (Ex 7.4)

Consider the following system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & v_0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} av_0/b \\ v_0/b \end{bmatrix} u$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]\, u$$



Reachability matrix:

$$W_r = [B \quad AB] = \begin{bmatrix} av_0/b & v_0^2/b \\ v_0/b & 0 \end{bmatrix}$$

Compute the determinant:

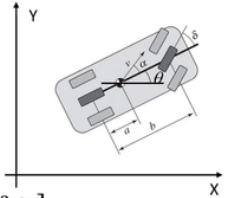
$$\det(W_r) = \begin{vmatrix} av_0/b & v_0^2/b \\ v_0/b & 0 \end{vmatrix} = av_0/b \cdot 0 - v_0/b \cdot v_0^2/b = -v_0^3/b^2 \neq 0$$

Revisit Example - Vehicle steering (Ex 7.4)

Consider the following system:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & v_0\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}av_0/b\\ v_0/b\end{bmatrix}u$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + [0]\,u$$



Reachability matrix:

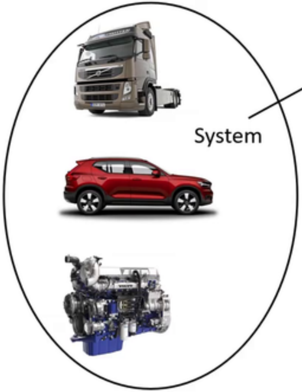
$$W_r = [B \quad AB] = \begin{bmatrix}av_0/b & v_0^2/b\\ v_0/b & 0\end{bmatrix}$$

Compute the determinant:

$$\det(W_r) = \begin{vmatrix}av_0/b & v_0^2/b\\ v_0/b & 0\end{vmatrix} = av_0/b\cdot 0 - v_0/b\cdot v_0^2/b = -v_0^3/b^2 \neq 0$$

The system is reachable, as long as $v_0 \neq 0$.

Summary - state feedback control



Model

$$\dot{x} = f(x, u, d)$$
$$y = h(x, u, d)$$

Analysis → ***Reachability***

Simulation

Prediction

Design

Control design

Tuning

Linearization

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

State feedback control

$$u = -Kx + k_r r$$

Pole placement

So far we have learnt how a state feedback looks like and when it is possible to design a state feedback controller to stabilize a system:

$$\dot{x} = Ax + Bu$$
$$u = -Kx + k_r r$$

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

Pole placement

So far we have learnt how a state feedback looks like and when it is possible to design a state feedback controller to stabilize a system:

$$\dot{x} = Ax + Bu$$
$$u = -Kx + k_r r$$

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

The questions that remain are: *How do we design a state feedback controller* and *Where do we place the closed loop system's poles?*

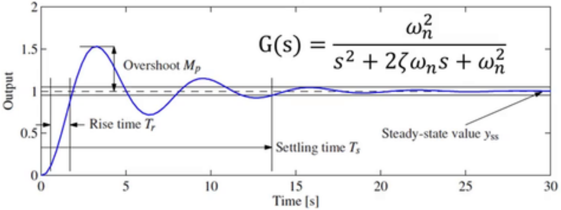
Pole placement

So far we have learnt how a state feedback looks like and when it is possible to design a state feedback controller to stabilize a system:

$$\dot{x} = Ax + Bu$$
$$u = -Kx + k_r r$$

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

The questions that remain are: *How do we design a state feedback controller* and *Where do we place the closed loop system's poles?*



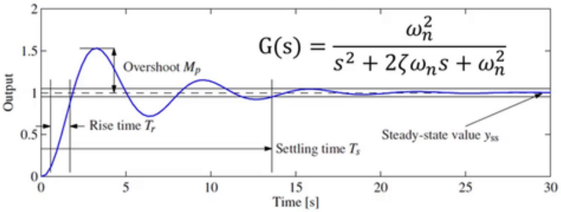
Pole placement

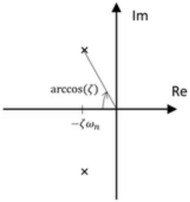
So far we have learnt how a state feedback looks like and when it is possible to design a state feedback controller to stabilize a system:

$$\dot{x} = Ax + Bu$$
$$u = -Kx + k_r r$$

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

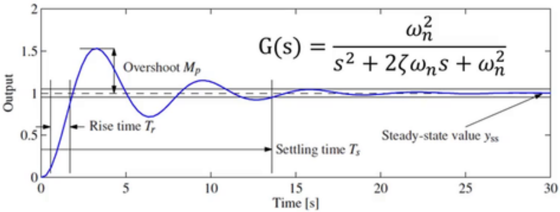
The questions that remain are: *How do we design a state feedback controller* and *Where do we place the closed loop system's poles?*

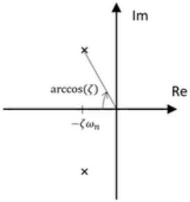




Specifications and pole placement

Property	Value	$\zeta = 0.5$	$\zeta = 1/\sqrt{2}$	$\zeta = 1$
Rise time	$T_r \approx 1/\omega_n \cdot e^{arccos\zeta/\tan(arccos\zeta)}$	$1.8/\omega_n$	$2.2/\omega_n$	$2.7/\omega_n$
Overshoot	$M_p \approx e^{-\pi\zeta\sqrt{1-\zeta^2}}$	16%	4%	0%
Settling time (2%)	$T_s \approx 4/\zeta\omega_n$	$8.0/\omega_n$	$5.9/\omega_n$	$4.0/\omega_n$





Pole placement

Where do we place the closed loop system's poles?

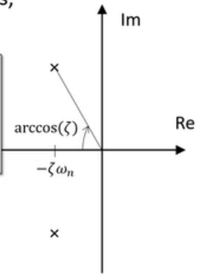
Pole placement

Where do we place the closed loop system's poles?

Idea:

- Use time domain specifications to place the dominant poles, as a second order system, $s^2 + 2\zeta\omega_n s + \omega_n^2$.

Property	Value	$\zeta = 0.5$	$\zeta = 1/\sqrt{2}$	$\zeta = 1$
Rise time	$T_r \approx 1/\omega_n \cdot e^{arccos\zeta/\tan(arccos\zeta)}$	$1.8/\omega_n$	$2.2/\omega_n$	$2.7/\omega_n$
Overshoot	$M_p \approx e^{-\pi\zeta\sqrt{1-\zeta^2}}$	16%	4%	0%
Settling time (2%)	$T_s \approx 4/\zeta\omega_n$	$8.0/\omega_n$	$5.9/\omega_n$	$4.0/\omega_n$

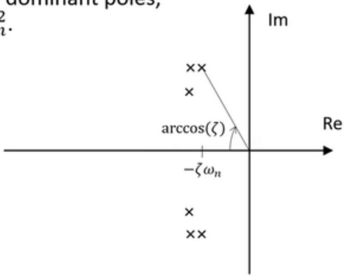


Pole placement

Where do we place the closed loop system's poles?

Idea:

- Use time domain specifications to place the dominant poles, as a second order system, $s^2 + 2\zeta\omega_n s + \omega_n^2$.
- Place the rest of the poles so they become faster than the dominant poles.

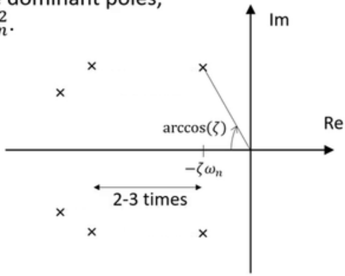


Pole placement

Where do we place the closed loop system's poles?

Idea:

- Use time domain specifications to place the dominant poles, as a second order system, $s^2 + 2\zeta\omega_n s + \omega_n^2$.
- Place the rest of the poles so they become faster than the dominant poles.



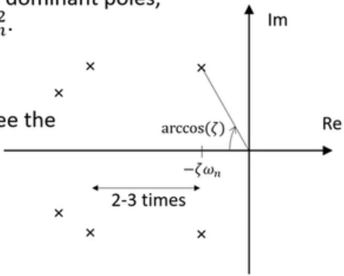
Pole placement

Where do we place the closed loop system's poles?

Idea:

- Use time domain specifications to place the dominant poles, as a second order system, $s^2 + 2\zeta\omega_n s + \omega_n^2$.
- Place the rest of the poles so they become faster than the dominant poles.

(The dominant second order poles, guarantee the time specification.)



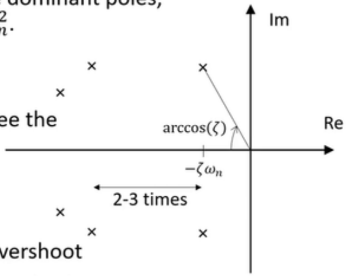
Pole placement

Where do we place the closed loop system's poles?

Idea:

- Use time domain specifications to place the dominant poles, as a second order system, $s^2 + 2\zeta\omega_n s + \omega_n^2$.
- Place the rest of the poles so they become faster than the dominant poles.

(The dominant second order poles, guarantee the time specification.)



Usually you end up with some zeros as well:

- Zeros in the left half plane give additional overshoot
- Zeros in the right half plane give a negative undershoot

Pole placement (Ackermann's formula)

Pole placement is performed by matching the desired characteristic polynomial with the closed loop system's characteristic polynomial.

Pole placement (Ackermann's formula)

Pole placement is performed by matching the desired characteristic polynomial with the closed loop system's characteristic polynomial.

From earlier example (vehicle steering) we have seen:

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 \equiv \lambda^2 + \frac{v_0}{b}(ak_1 + k_2)\lambda + \frac{k_1 v_0^2}{b}$$

$$k_1 = \frac{b\omega_n^2}{v_0^2} \qquad k_2 = \frac{2\zeta\omega_n b}{v_0} - \frac{ab\omega_n^2}{v_0^2}$$

Pole placement (Ackermann's formula)

Pole placement is performed by matching the desired characteristic polynomial with the closed loop system's characteristic polynomial.

From earlier example (vehicle steering) we have seen:

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 \equiv \lambda^2 + \frac{v_0}{b}(ak_1 + k_2)\lambda + \frac{k_1 v_0^2}{b}$$

$$k_1 = \frac{b\omega_n^2}{v_0^2} \qquad k_2 = \frac{2\zeta\omega_n b}{v_0} - \frac{ab\omega_n^2}{v_0^2}$$

For low order systems it is ok, but for larger systems this is boring work.

Pole placement (Ackermann's formula)

Pole placement is performed by matching the desired characteristic polynomial with the closed loop system's characteristic polynomial.

From earlier example (vehicle steering) we have seen:

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 \equiv \lambda^2 + \frac{v_0}{b}(ak_1 + k_2)\lambda + \frac{k_1 v_0^2}{b}$$

$$k_1 = \frac{b\omega_n^2}{v_0^2} \qquad k_2 = \frac{2\zeta\omega_n b}{v_0} - \frac{ab\omega_n^2}{v_0^2}$$

For low order systems it is ok, but for larger systems this is boring work.

Ackermann's formula offers us a method to do this in one computational step.

Pole placement (Ackermann's formula)

Consider a system $\dot{x} = Ax + Bu$ with the characteristic polynomial

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

Pole placement (Ackermann's formula)

Consider a system $\dot{x} = Ax + Bu$ with the characteristic polynomial

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

If the system is reachable, then there exist a control law, $u = -Kx$, that gives a closed loop system with the charateristic polynomial

$$p(s) = s^n + p_1 s^{n-1} + \cdots p_{n-1} s + p_n.$$

Pole placement (Ackermann's formula)

The feedback gain is given by

$$K = [p_1 - a_1 \quad p_2 - a_2 \quad \dots \quad p_n - a_n]\widetilde{W}_r W_r^{-1}$$

Pole placement (Ackermann's formula)

The feedback gain is given by

$$K = [p_1 - a_1 \quad p_2 - a_2 \quad \dots \quad p_n - a_n]\widetilde{W}_r W_r^{-1}$$

where W_r is the reachability matrix

$$W_r = [B \quad AB \quad \dots \quad A^{n-1}B]$$

and

$$\widetilde{W}_r = \begin{bmatrix} 1 & a_1 & a_2 & \cdots & a_{n-1} \\ 0 & 1 & a_1 & \cdots & a_{n-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & a_1 \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}^{-1}$$

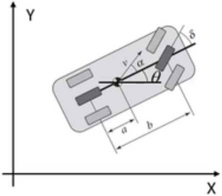
This is called ***Ackermann's formula***.

Revisit Example - Vehicle steering (Ex 7.4)

Consider the following system:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & v_0\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}av_0/b\\ v_0/b\end{bmatrix}u$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + [0]\,u$$

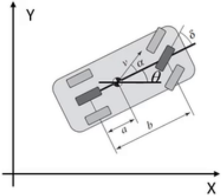


Revisit Example - Vehicle steering (Ex 7.4)

Consider the following system:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & v_0\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}av_0/b\\ v_0/b\end{bmatrix}u$$

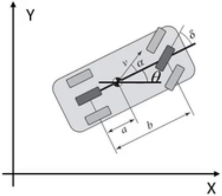
$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + [0]\,u$$



Determine the characteristic polynomial for the system:

$$\det(sI - A) = \begin{vmatrix}s & -v_0\\ 0 & s\end{vmatrix} = s^2 = s^2 + 0s + 0$$

Revisit Example - Vehicle steering (Ex 7.4)



Consider the following system:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & v_0\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}av_0/b\\ v_0/b\end{bmatrix}u$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + [0]\,u$$

Determine the characteristic polynomial for the system:

$$\det(sI - A) = \begin{vmatrix}s & -v_0\\ 0 & s\end{vmatrix} = s^2 = s^2 + 0s + 0$$

Desired characteristic polynomial for the closed loop system:

$$p(s) = s^2 + 2\zeta\omega_n s + \omega_n^2$$

Revisit Example - Vehicle steering (Ex 7.4)

The feedback gain is given by

$$K = [p_1 - a_1 \quad p_2 - a_2]\widetilde{W}_r W_r^{-1} = [2\zeta\omega_n \quad \omega_n^2]\widetilde{W}_r W_r^{-1}$$

Revisit Example - Vehicle steering (Ex 7.4)

The feedback gain is given by

$$K = [p_1 - a_1 \quad p_2 - a_2]\widetilde{W}_r W_r^{-1} = [2\zeta\omega_n \quad \omega_n^2]\widetilde{W}_r W_r^{-1}$$

where W_r is the reachability matrix

$$W_r = [B \quad AB] = \begin{bmatrix} av_0/b & v_0^2/b \\ v_0/b & 0 \end{bmatrix}$$

and

$$\widetilde{W}_r = \begin{bmatrix} 1 & a_1 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Revisit Example - Vehicle steering (Ex 7.4)

The feedback gain is given by

$$K = [p_1 - a_1 \quad p_2 - a_2]\widetilde{W}_r W_r^{-1} = [2\zeta\omega_n \quad \omega_n^2]\widetilde{W}_r W_r^{-1}$$

where W_r is the reachability matrix

$$W_r = [B \quad AB] = \begin{bmatrix} av_0/b & v_0^2/b \\ v_0/b & 0 \end{bmatrix}$$

and

$$\widetilde{W}_r = \begin{bmatrix} 1 & a_1 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The feedback gain, can then be computed as

$$K = [2\zeta\omega_n \quad \omega_n^2]\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} av_0/b & v_0^2/b \\ v_0/b & 0 \end{bmatrix}^{-1}$$

Revisit Example - Vehicle steering (Ex 7.4)

The feedback gain is given by

$$K = [p_1 - a_1 \quad p_2 - a_2]\widetilde{W}_r W_r^{-1} = [2\zeta\omega_n \quad \omega_n^2]\widetilde{W}_r W_r^{-1}$$

where W_r is the reachability matrix

$$W_r = [B \quad AB] = \begin{bmatrix} av_0/b & v_0^2/b \\ v_0/b & 0 \end{bmatrix}$$

and

$$\widetilde{W}_r = \begin{bmatrix} 1 & a_1 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The feedback gain, can then be computed as

$$K = [2\zeta\omega_n \quad \omega_n^2]\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} av_0/b & v_0^2/b \\ v_0/b & 0 \end{bmatrix}^{-1} = [2\zeta\omega_n \quad \omega_n^2]\begin{bmatrix} 0 & b/v_0 \\ b/v_0^2 & -ab/v_0^2 \end{bmatrix}$$

Revisit Example - Vehicle steering (Ex 7.4)

The feedback gain is given by

$$K = [p_1 - a_1 \quad p_2 - a_2]\tilde{W}_r W_r^{-1} = [2\zeta\omega_n \quad \omega_n^2]\tilde{W}_r W_r^{-1}$$

where W_r is the reachability matrix

$$W_r = [B \quad AB] = \begin{bmatrix} av_0/b & v_0^2/b \\ v_0/b & 0 \end{bmatrix}$$

and

$$\tilde{W}_r = \begin{bmatrix} 1 & a_1 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The feedback gain, can then be computed as

$$K = [2\zeta\omega_n \quad \omega_n^2]\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} av_0/b & v_0^2/b \\ v_0/b & 0 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{b\omega_n^2}{v_0^2} & \frac{2\zeta\omega_n b}{v_0} - \frac{ab\omega_n^2}{v_0^2} \end{bmatrix}$$

Revisit Example - Vehicle steering (Ex 7.4)

The feedback gain is given by

$$K = [p_1 - a_1 \quad p_2 - a_2]\widetilde{W}_r W_r^{-1} = [2\zeta\omega_n \quad \omega_n^2]\widetilde{W}_r W_r^{-1}$$

where W_r is the reachability matrix

$$W_r = [B \quad AB] = \begin{bmatrix} av_0/b & v_0^2/b \\ v_0/b & 0 \end{bmatrix}$$

and

$$\widetilde{W}_r = \begin{bmatrix} 1 & a_1 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The feedback gain, can then be computed as

$$K = [2\zeta\omega_n \quad \omega_n^2]\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} av_0/b & v_0^2/b \\ v_0/b & 0 \end{bmatrix}^{-1} = \begin{bmatrix} \underbrace{\dfrac{b\omega_n^2}{v_0^2}}_{k_1} & \underbrace{\dfrac{2\zeta\omega_n b}{v_0} - \dfrac{ab\omega_n^2}{v_0^2}}_{k_2} \end{bmatrix}$$

Linear Quadratic Regulator

An alternative method to pole placement is to place the poles so that the closed loop system optimizes a cost function:

Linear Quadratic Regulator

An alternative method to pole placement is to place the poles so that the closed loop system optimizes a cost function:

$$J = \int_0^\infty (x^T Q_x x + u^T Q_u u) dt$$

Linear Quadratic Regulator

An alternative method to pole placement is to place the poles so that the closed loop system optimizes a cost function:

$$J = \int_0^{\infty} (x^T Q_x x + u^T Q_u u) dt$$

where $x^T Q_x x$ is the ***state cost*** and $u^T Q_u u$ is the ***control cost***. The matrices Q_x and Q_u are symmetric, positive (semi-) definite matrices. This is called the ***linear quadratic regulator (LQR)*** problem.

Linear Quadratic Regulator

An alternative method to pole placement is to place the poles so that the closed loop system optimizes a cost function:

$$J = \int_0^\infty (x^T Q_x x + u^T Q_u u) dt$$

where $x^T Q_x x$ is the ***state cost*** and $u^T Q_u u$ is the ***control cost***. The matrices Q_x and Q_u are symmetric, positive (semi-) definite matrices. This is called the ***linear quadratic regulator (LQR)*** problem.

The solution to the LQR problem is given by

$$u = -Kx, \qquad K = Q_u^{-1} B^T S$$

Linear Quadratic Regulator

An alternative method to pole placement is to place the poles so that the closed loop system optimizes a cost function:

$$J = \int_0^\infty (x^T Q_x x + u^T Q_u u) dt$$

where $x^T Q_x x$ is the ***state cost*** and $u^T Q_u u$ is the ***control cost***. The matrices Q_x and Q_u are symmetric, positive (semi-) definite matrices. This is called the ***linear quadratic regulator (LQR)*** problem.

The solution to the LQR problem is given by

$$u = -Kx, \qquad K = Q_u^{-1} B^T S$$

where S is a positive definite, symmetric matrix given by

$$A^T S + SA - SBQ_u^{-1} B^T S + Q_x = 0$$

This equation is called the ***algebraic Riccati equation***.

Linear Quadratic Regulator

The tuning of the LQR is to choose the weighting matrices Q_x and Q_u. To guarantee that a solution exists, the system must be ***reachable*** and that $\boldsymbol{Q_x} \succcurlyeq \mathbf{0}$ and $\boldsymbol{Q_u} \succ \mathbf{0}$.

Linear Quadratic Regulator

The tuning of the LQR is to choose the weighting matrices Q_x and Q_u. To guarantee that a solution exists, the system must be ***reachable*** and that $\boldsymbol{Q_x} \succcurlyeq \mathbf{0}$ and $\boldsymbol{Q_u} \succ \mathbf{0}$.

1. Simplest choice: $Q_x = I$ and $Q_x = \rho I$

$$J = \int_0^\infty (x^T x + \rho u^T u) dt$$

Linear Quadratic Regulator

The tuning of the LQR is to choose the weighting matrices Q_x and Q_u. To guarantee that a solution exists, the system must be ***reachable*** and that $\boldsymbol{Q_x} \succcurlyeq \mathbf{0}$ and $\boldsymbol{Q_u} \succ \mathbf{0}$.

1. Simplest choice: $Q_x = I$ and $Q_x = \rho I$

$$J = \int_0^\infty (x^T x + \rho u^T u) dt \quad \Longrightarrow \quad \text{trade-off} \Longrightarrow \|x\|^2 \;\; vs \;\; \rho\|u\|^2$$

This reduce the tuning to select ρ, which then becomes a trade-off between state cost and control cost.

Linear Quadratic Regulator

The tuning of the LQR is to choose the weighting matrices Q_x and Q_u. To guarantee that a solution exists, the system must be ***reachable*** and that $\boldsymbol{Q_x \succcurlyeq 0}$ and $\boldsymbol{Q_u \succ 0}$.

1. Simplest choice: $Q_x = I$ and $Q_x = \rho I$

$$J = \int_0^\infty (x^T x + \rho u^T u) dt \quad \Longrightarrow \quad \text{trade-off} \Longrightarrow \|x\|^2 \ \ vs \ \ \rho\|u\|^2$$

   This reduce the tuning to select ρ, which then becomes a trade-off between state cost and control cost.

2. Output weighting. Let $z = C_z x$ be the output you want to keep small.

   Choose $Q_x = C_z^T C_z$, and $Q_x = \rho I$.

Linear Quadratic Regulator

The tuning of the LQR is to choose the weighting matrices Q_x and Q_u. To guarantee that a solution exists, the system must be ***reachable*** and that $\boldsymbol{Q_x \succcurlyeq 0}$ and $\boldsymbol{Q_u \succ 0}$.

1. Simplest choice: $Q_x = I$ and $Q_x = \rho I$

$$J = \int_0^\infty (x^T x + \rho u^T u)dt \qquad \Longrightarrow \qquad \text{trade-off} \Longrightarrow \|x\|^2 \quad vs \quad \rho\|u\|^2$$

   This reduce the tuning to select ρ, which then becomes a trade-off between state cost and control cost.

2. Output weighting. Let $z = C_z x$ be the output you want to keep small.

   Choose $Q_x = C_z^T C_z$, and $Q_x = \rho I$. $\qquad \Longrightarrow \qquad$ trade-off $\Longrightarrow \|z\|^2 \quad vs \quad \rho\|u\|^2$

Linear Quadratic Regulator

3. Diagnonal weighting.

$$Q_x = \begin{bmatrix} q_1 & & 0 \\ & \ddots & \\ 0 & & q_n \end{bmatrix} \quad Q_u = \begin{bmatrix} \rho_1 & & 0 \\ & \ddots & \\ 0 & & \rho_p \end{bmatrix}$$

Linear Quadratic Regulator

3. Diagnonal weighting.

$$Q_x = \begin{bmatrix} q_1 & & 0 \\ & \ddots & \\ 0 & & q_n \end{bmatrix} \quad Q_u = \begin{bmatrix} \rho_1 & & 0 \\ & \ddots & \\ 0 & & \rho_p \end{bmatrix}$$

Choose the individual diagonal elements based on how much each state or input signal should contribute to the overall cost.

Linear Quadratic Regulator

3. Diagnonal weighting.

$$Q_x = \begin{bmatrix} q_1 & & 0 \\ & \ddots & \\ 0 & & q_n \end{bmatrix} \quad Q_u = \begin{bmatrix} \rho_1 & & 0 \\ & \ddots & \\ 0 & & \rho_p \end{bmatrix}$$

Choose the individual diagonal elements based on how much each state or input signal should contribute to the overall cost.

Alternative, (***Bryson's rule***) choose the diagonal weights as $q_i = \alpha_i^2 / x_{i,max}^2$ and $\rho_i = \beta_i^2 / u_{i,max}^2$, where $x_{i,max}$ *and* $u_{i,max}$ represents the largest response. α and β are used for additional individual weighting of the state and control cost,

$$\sum_{i=1}^{n} \alpha_i^2 = 1 \qquad \sum_{i=1}^{p} \beta_i^2 = 1$$

Linear Quadratic Regulator

3. Diagnonal weighting.

$$Q_x = \begin{bmatrix} q_1 & & 0 \\ & \ddots & \\ 0 & & q_n \end{bmatrix} \quad Q_u = \begin{bmatrix} \rho_1 & & 0 \\ & \ddots & \\ 0 & & \rho_p \end{bmatrix}$$

Choose the individual diagonal elements based on how much each state or input signal should contribute to the overall cost.

Alternative, (***Bryson's rule***) choose the diagonal weights as $q_i = \alpha_i^2 / x_{i,max}^2$ and $\rho_i = \beta_i^2 / u_{i,max}^2$, where $x_{i,max}$ *and* $u_{i,max}$ represents the largest response. α and β are used for additional individual weighting of the state and control cost,

$$\sum_{i=1}^{n} \alpha_i^2 = 1 \qquad \sum_{i=1}^{p} \beta_i^2 = 1$$

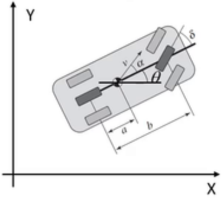
4. Trial and error

Revisit Example - Vehicle steering (Ex 7.4)

Consider the following system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & v_0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} av_0/b \\ v_0/b \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]\, u$$



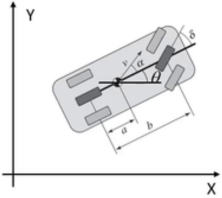
Vehicle data: $v_0 = 12\ m/s$
$a = 2\ m$
$b = 4\ m$

Revisit Example - Vehicle steering (Ex 7.4)

Consider the following system:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 12\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}6\\ 3\end{bmatrix}u$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + [0]\,u$$



Place the poles so that the closed loop system optimizes the cost function:

$$J = \int_0^{\infty}\left(x^T Q_x x + u^T Q_u u\right)dt$$

where

$$Q_x = \begin{bmatrix}q_1 & 0\\ 0 & q_2\end{bmatrix} \qquad Q_u = \rho$$

Vehicle data: $v_0 = 12\,m/s$

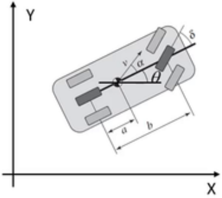
$a = 2\,m$

$b = 4\,m$

Revisit Example - Vehicle steering (Ex 7.4)

For the case when

$$Q_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad Q_u = 10$$



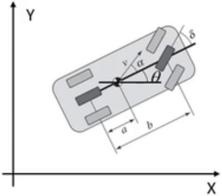
Revisit Example - Vehicle steering (Ex 7.4)

For the case when

$$Q_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad Q_u = 10$$

The solution to the algebraic Ricatti equation is

$$A^T S + SA - SBQ_u^{-1} B^T S + Q_x = 0$$



In MATLAB: The LQR problem can be solved using the `lqr` command

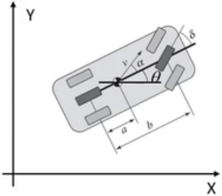
Revisit Example - Vehicle steering (Ex 7.4)

For the case when

$$Q_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad Q_u = 10$$

The solution to the algebraic Ricatti equation is

$$S = \begin{bmatrix} 0.292 & 0.470 \\ 0.470 & 2.754 \end{bmatrix}$$



In MATLAB: The LQR problem can be solved using the `lqr` command

Revisit Example - Vehicle steering (Ex 7.4)

For the case when

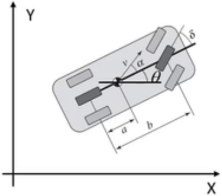
$$Q_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad Q_u = 10$$

The solution to the algebraic Ricatti equation is

$$S = \begin{bmatrix} 0.292 & 0.470 \\ 0.470 & 2.754 \end{bmatrix}$$

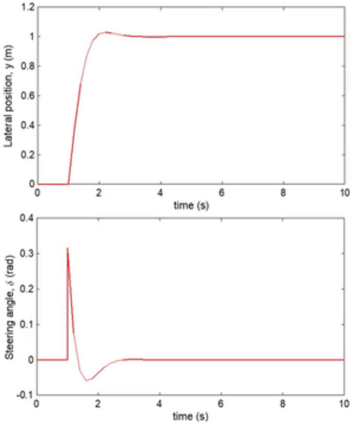
and the corresponding control law becomes

$$u = -Kx, \qquad K = Q_u^{-1} B^T S = \begin{bmatrix} 0.316 & 1.108 \end{bmatrix}$$

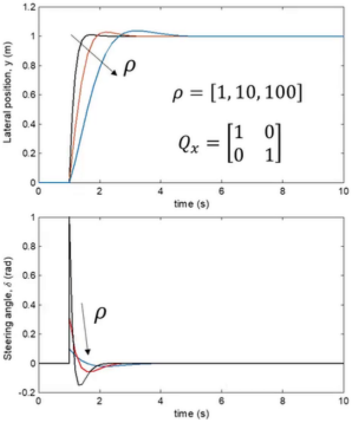


In MATLAB: The LQR problem can be solved using the `lqr` command

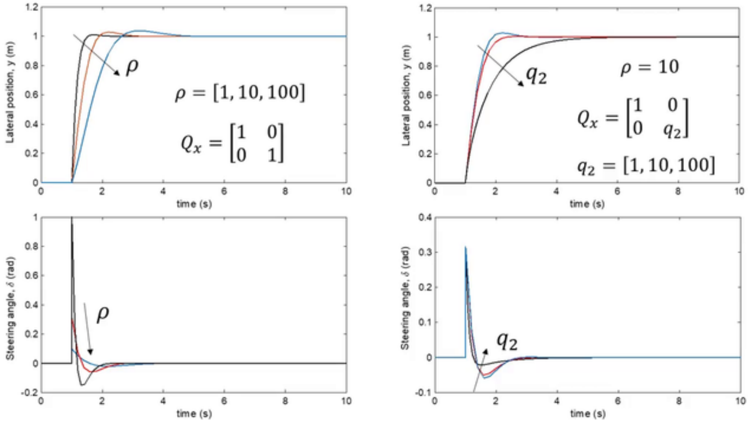
Revisit Example - Vehicle steering (Ex 7.4)



Revisit Example - Vehicle steering (Ex 7.4)



Revisit Example - Vehicle steering (Ex 7.4)



Revisit Example - Vehicle steering (Ex 7.4)

For the case when

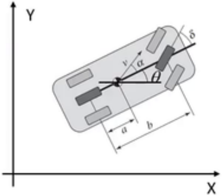
$$Q_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad Q_u = 10$$

The solution to the algebraic Ricatti equation is

$$S = \begin{bmatrix} 0.292 & 0.470 \\ 0.470 & 2.754 \end{bmatrix}$$

and the corresponding control law becomes

$$u = -Kx, \qquad K = Q_u^{-1} B^T S = \begin{bmatrix} 0.316 & 1.108 \end{bmatrix}$$



In MATLAB: The LQR problem can be solved using the `lqr` command

Revisit Example - Vehicle steering (Ex 7.4)

For the case when

$$Q_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad Q_u = 10$$

The solution to the algebraic Ricatti equation is

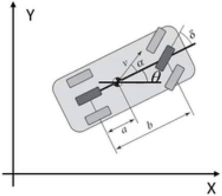
$$S = \begin{bmatrix} 0.292 & 0.470 \\ 0.470 & 2.754 \end{bmatrix}$$

and the corresponding control law becomes

$$u = -Kx, \qquad K = Q_u^{-1} B^T S = \begin{bmatrix} 0.316 & 1.108 \end{bmatrix}$$

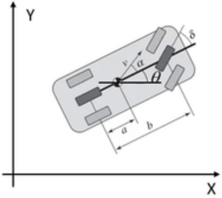
The closed loop system poles are

$$E = \begin{bmatrix} -2.6110 + 2.1371i \\ -2.6110 - 2.1371i \end{bmatrix}$$



In MATLAB: The LQR problem can be solved using the `lqr` command

Revisit Example - Vehicle steering (Ex 7.4)



For the case when

$$Q_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad Q_u = 10$$

The solution to the algebraic Ricatti equation is

$$S = \begin{bmatrix} 0.292 & 0.470 \\ 0.470 & 2.754 \end{bmatrix}$$

and the corresponding control law becomes

$$u = -Kx, \qquad K = Q_u^{-1} B^T S = [0.316 \quad 1.108]$$

The closed loop system poles are

$$E = \begin{bmatrix} -2.6110 + 2.1371i \\ -2.6110 - 2.1371i \end{bmatrix}$$

Compared to the pole placement design, this corresponds to $\zeta = 0.77$ and $\omega_n = 3.44$.

In MATLAB: The LQR problem can be solved using the `lqr` command

Bibliography

- Karl J. Astrom and Richard M. Murray *Feedback Systems*. Version v3.0i. Princeton University Press. September 2018. Chapter 11.