NTNU | Norwegian University of Science and Technology

SMALL-SIGNAL STABILITY:

Linearization and eigenvalue analysis

Gilbert Bergna-Diaz

September 17, 2024

Objectives of this lecture

Lecture goals:

- Learn how to linearize your nonlinear system,
  - ...and validate your linear model numerically in Matlab/Simulink.
- Understand how linear systems are expected to behave.
- Learn how to perform an eigenvalue-based stability analysis,
  - ...using your linearized system,
  - ...and use this tool to help you tune your control parameters.

Recommended reading:

- Sections 2.2-2.3, 3.1-3.3, 6.1, 6.4 (p.410).

Outline

Recap from previous lectures

Linearization of nonlinear systems

Bloc 3 Challenge $\times 1/2$

Response of a linear system

Modal transformation & eigenvalues

Bloc 3 (full) Challenge

Outline

Recap from previous lectures

Linearization of nonlinear systems

Bloc 3 Challenge $\times 1/2$

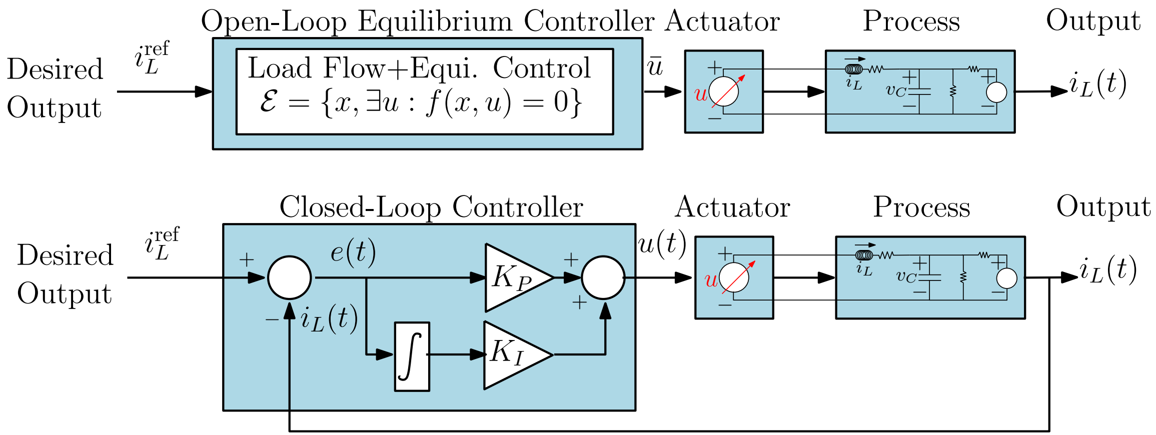
Response of a linear system

Modal transformation & eigenvalues

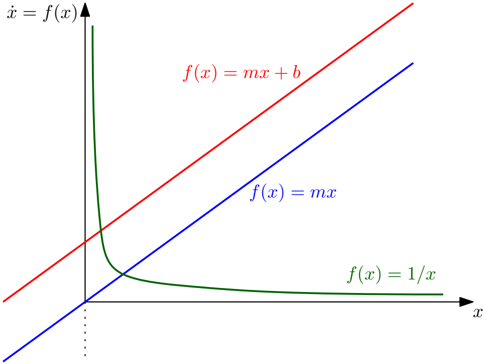
Bloc 3 (full) Challenge

What we have so far:

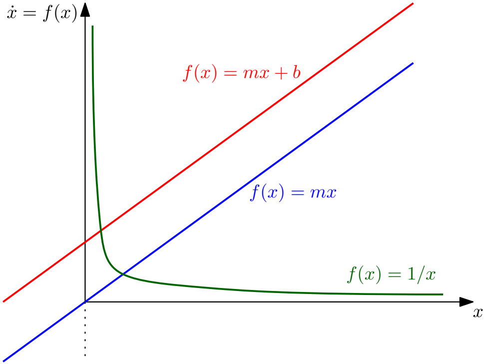
What we have so far:



Is my model linear or nonlinear? (Chapt. 2.3)

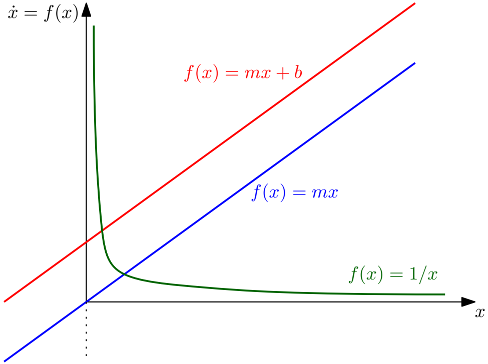


Is my model linear or nonlinear? (Chapt. 2.3)



- A system (vetor field or function) $f(x)$ is linear when it satisfies the properties of:
  - **Superposition** $\rightarrow f(x_1 + x_2) = f(x_1) + f(x_2)$.
  - **Homogeneity** $\rightarrow f(\alpha x) = \alpha f(x)$.

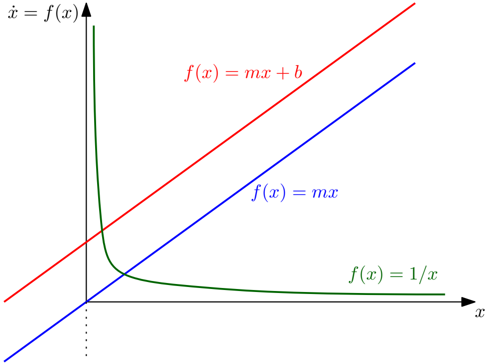
Is my model linear or nonlinear? (Chapt. 2.3)



- $\underbrace{m(x_1 + x_2)}_{f(x_1+x_2)} = \underbrace{mx_1}_{f(x_1)} + \underbrace{mx_2}_{f(x_2)} \checkmark;$

- A system (vetor field or function) $f(x)$ is linear when it satisfies the properties of:
  - **Superposition** $\rightarrow f(x_1 + x_2) = f(x_1) + f(x_2)$.
  - **Homogeneity** $\rightarrow f(\alpha x) = \alpha f(x)$.

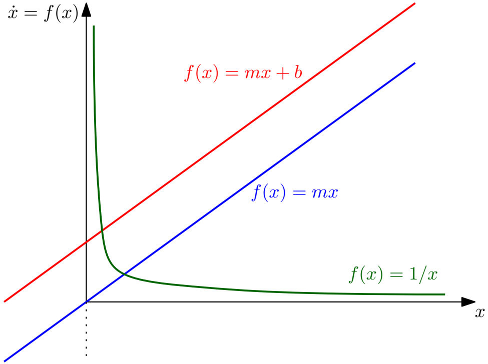
Is my model linear or nonlinear? (Chapt. 2.3)



- $\underbrace{m(x_1 + x_2)}_{f(x_1+x_2)} = \underbrace{mx_1}_{f(x_1)} + \underbrace{mx_2}_{f(x_2)} \checkmark;\quad \underbrace{m\alpha x}_{f(\alpha x)} = \underbrace{\alpha mx}_{\alpha f(x)} \checkmark$

- A system (vetor field or function) $f(x)$ is linear when it satisfies the properties of:
  - **Superposition**$\rightarrow f(x_1 + x_2) = f(x_1) + f(x_2)$.
  - **Homogeneity**$\rightarrow f(\alpha x) = \alpha f(x)$.

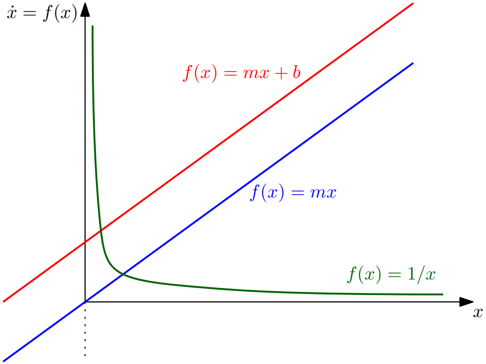
Is my model linear or nonlinear? (Chapt. 2.3)



- $\underbrace{m(x_1 + x_2)}_{f(x_1+x_2)} = \underbrace{mx_1}_{f(x_1)} + \underbrace{mx_2}_{f(x_2)} \checkmark;\quad \underbrace{m\alpha x}_{f(\alpha x)} = \underbrace{\alpha m x}_{\alpha f(x)} \checkmark$
- $\underbrace{m(x_1 + x_2) + b}_{f(x_1+x_2)} \neq \underbrace{mx_1 + b}_{f(x_1)} + \underbrace{mx_2 + b}_{f(x_2)}$

- A system (vetor field or function) $f(x)$ is linear when it satisfies the properties of:
  - **Superposition** $\rightarrow f(x_1 + x_2) = f(x_1) + f(x_2)$.
  - **Homogeneity** $\rightarrow f(\alpha x) = \alpha f(x)$.

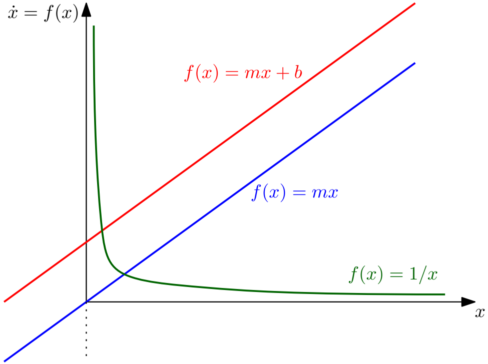
Is my model linear or nonlinear? (Chapt. 2.3)



- $\underbrace{m(x_1 + x_2)}_{f(x_1+x_2)} = \underbrace{mx_1}_{f(x_1)} + \underbrace{mx_2}_{f(x_2)} \checkmark; \quad \underbrace{m\alpha x}_{f(\alpha x)} = \underbrace{\alpha m x}_{\alpha f(x)} \checkmark$
- $\underbrace{m(x_1 + x_2) + b}_{f(x_1+x_2)} \neq \underbrace{mx_1 + b}_{f(x_1)} + \underbrace{mx_2 + b}_{f(x_2)}$
- $\underbrace{m\alpha x + b}_{f(\alpha x)} \neq \underbrace{\alpha(mx + b)}_{\alpha f(x)}$

- A system (vetor field or function) $f(x)$ is linear when it satisfies the properties of:
  - **Superposition** $\rightarrow f(x_1 + x_2) = f(x_1) + f(x_2)$.
  - **Homogeneity** $\rightarrow f(\alpha x) = \alpha f(x)$.

Is my model linear or nonlinear? (Chapt. 2.3)



- $\underbrace{m(x_1 + x_2)}_{f(x_1+x_2)} = \underbrace{mx_1}_{f(x_1)} + \underbrace{mx_2}_{f(x_2)} \checkmark$; $\underbrace{m\alpha x}_{f(\alpha x)} = \underbrace{\alpha m x}_{\alpha f(x)} \checkmark$
- $\underbrace{m(x_1 + x_2) + b}_{f(x_1+x_2)} \neq \underbrace{mx_1 + b}_{f(x_1)} + \underbrace{mx_2 + b}_{f(x_2)}$
- $\underbrace{m\alpha x + b}_{f(\alpha x)} \neq \underbrace{\alpha(mx + b)}_{\alpha f(x)}$
- $\dfrac{1}{\underbrace{x_1 + x_2}_{f(x_1+x_2)}} \neq \dfrac{1}{\underbrace{x_1}_{f(x_1)}} + \dfrac{1}{\underbrace{x_2}_{f(x_2)}}$; $\dfrac{1}{\underbrace{\alpha x}_{f(\alpha x)}} \neq \underbrace{\alpha \dfrac{1}{x}}_{\alpha f(x)}$

- A system (vetor field or function) $f(x)$ is linear when it satisfies the properties of:
  - **Superposition**$\rightarrow$ $f(x_1 + x_2) = f(x_1) + f(x_2)$.
  - **Homogeneity**$\rightarrow$ $f(\alpha x) = \alpha f(x)$.

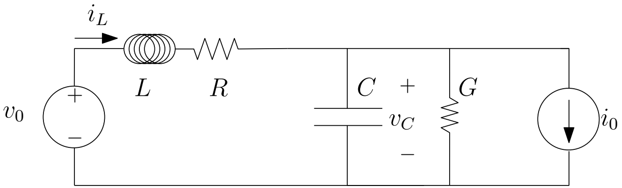
Is my model linear or nonlinear? (Chapt. 2.3)

However, the system $\dot{x} = mx + b$ may be considered linear about an operating point $\bar{x}$.
Define $\tilde{x} := x - \bar{x}$, then we have that:

$$\begin{aligned} \dot{x} &= mx + b \\ - \quad & \\ \dot{\bar{x}} &= m\bar{x} + b \\ \hline \dot{\tilde{x}} &= m\tilde{x} \end{aligned}$$

which is linear!

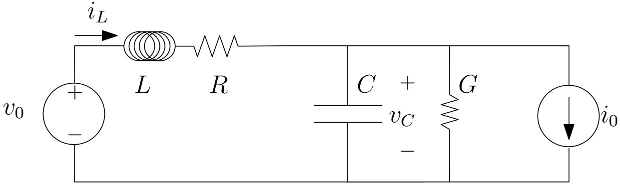
Is my model linear or nonlinear? (Chapt. 2.3)



State-Space Model:

$$
\begin{aligned}
\frac{d}{dt} i_L &= -\frac{R}{L} i_L - \frac{1}{L} v_C + \frac{1}{L} v_0 \\
\frac{d}{dt} v_C &= -\frac{G}{C} v_C + \frac{1}{C} i_L - \frac{1}{C} i_0
\end{aligned}
$$

Is my model linear or nonlinear? (Chapt. 2.3)

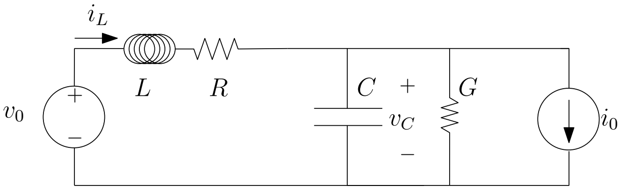


State-Space Model:

$$\frac{d}{dt} i_L = -\frac{R}{L} i_L - \frac{1}{L} v_C + \frac{1}{L} v_0$$
$$\frac{d}{dt} v_C = -\frac{G}{C} v_C + \frac{1}{C} i_L - \frac{1}{C} i_0$$

$$\frac{d}{dt} \underbrace{\begin{bmatrix} i_L \\ v_C \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & -\frac{G}{C} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} i_L \\ v_C \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} \frac{v_0}{L} \\ -\frac{i_0}{C} \end{bmatrix}}_{E}$$

Is my model linear or nonlinear? (Chapt. 2.3)



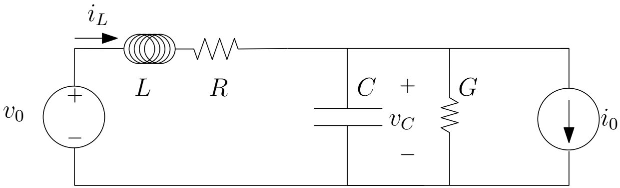
State-Space Model:

$$\frac{d}{dt}i_L = -\frac{R}{L}i_L - \frac{1}{L}v_C + \frac{1}{L}v_0$$
$$\frac{d}{dt}v_C = -\frac{G}{C}v_C + \frac{1}{C}i_L - \frac{1}{C}i_0$$

$$\frac{d}{dt}\underbrace{\begin{bmatrix} i_L \\ v_C \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & -\frac{G}{C} \end{bmatrix}}_{A}\underbrace{\begin{bmatrix} i_L \\ v_C \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} \frac{v_0}{L} \\ -\frac{i_0}{C} \end{bmatrix}}_{E}$$

- In **compact form**: $\frac{d}{dt}x = Ax + E$

Is my model linear or nonlinear? (Chapt. 2.3)



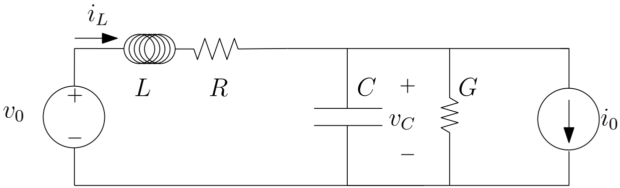
State-Space Model:

$$\frac{d}{dt} i_L = -\frac{R}{L} i_L - \frac{1}{L} v_C + \frac{1}{L} v_0$$
$$\frac{d}{dt} v_C = -\frac{G}{C} v_C + \frac{1}{C} i_L - \frac{1}{C} i_0$$

$$\frac{d}{dt} \underbrace{\begin{bmatrix} i_L \\ v_C \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & -\frac{G}{C} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} i_L \\ v_C \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} \frac{v_0}{L} \\ -\frac{i_0}{C} \end{bmatrix}}_{E}$$

- In **compact form**: $\frac{d}{dt} x = Ax + E$
- At the **equilibrium:** $\frac{d}{dt} \bar{x} = A\bar{x} + E$

Is my model linear or nonlinear? (Chapt. 2.3)



State-Space Model:

$$\frac{d}{dt}\underbrace{\begin{bmatrix} i_L \\ v_C \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & -\frac{G}{C} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} i_L \\ v_C \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} \frac{v_0}{L} \\ -\frac{i_0}{C} \end{bmatrix}}_{E}$$

$$\frac{d}{dt} i_L = -\frac{R}{L} i_L - \frac{1}{L} v_C + \frac{1}{L} v_0$$
$$\frac{d}{dt} v_C = -\frac{G}{C} v_C + \frac{1}{C} i_L - \frac{1}{C} i_0$$

- In **compact form**: $\frac{d}{dt} x = Ax + E$
- At the **equilibrium:** $\frac{d}{dt} \bar{x} = A\bar{x} + E$
- **Incremental** model: $\frac{d}{dt} \tilde{x} = A\tilde{x} \leftarrow$ linear!
  - (with $\tilde{x} := x - \bar{x}$).

Outline

Recap from previous lectures

Linearization of nonlinear systems

Bloc 3 Challenge$\times 1/2$

Response of a linear system

Modal transformation & eigenvalues

Bloc 3 (full) Challenge

Why do we want to linearize our nonlinear system?

- **Where we' at:** So far, we are able to control our systems with acceptable performance but...

Why do we want to linearize our nonlinear system?

- **Where we' at:** So far, we are able to control our systems with acceptable performance but...
- **Main enemy:** there is always a risk of instability!

Why do we want to linearize our nonlinear system?

- **Where we' at:** So far, we are able to control our systems with acceptable performance but...
- **Main enemy:** there is always a risk of instability!
  - Recall: PI controller gives you exactly what you want or unstable performance.

Why do we want to linearize our nonlinear system?

- **Where we' at:** So far, we are able to control our systems with acceptable performance but...
- **Main enemy:** there is always a risk of instability!
  - Recall: PI controller gives you exactly what you want or unstable performance.
- **Main motivation (1):** Predict if your system will be stable or...tune it such that this is guaranteed.

Why do we want to linearize our nonlinear system?

- **Where we' at:** So far, we are able to control our systems with acceptable performance but...
- **Main enemy:** there is always a risk of instability!
  - Recall: PI controller gives you exactly what you want or unstable performance.
- **Main motivation (1):** Predict if your system will be stable or...tune it such that this is guaranteed.
- **Main motivation (2):** Design your system to perform in a certain way.

Why do we want to linearize our nonlinear system?

- **Where we' at:** So far, we are able to control our systems with acceptable performance but...
- **Main enemy:** there is always a risk of instability!
  - Recall: PI controller gives you exactly what you want or unstable performance.
- **Main motivation (1):** Predict if your system will be stable or...tune it such that this is guaranteed.
- **Main motivation (2):** Design your system to perform in a certain way.
- Linear systems are easier than nonlinear systems, particularly to address points **(1)** and **(2)**,

Why do we want to linearize our nonlinear system?

- **Where we' at:** So far, we are able to control our systems with acceptable performance but...
- **Main enemy:** there is always a risk of instability!
  - Recall: PI controller gives you exactly what you want or unstable performance.
- **Main motivation (1):** Predict if your system will be stable or...tune it such that this is guaranteed.
- **Main motivation (2):** Design your system to perform in a certain way.
- Linear systems are easier than nonlinear systems, particularly to address points **(1)** and **(2)**,
  - as we have a large set of mathematical and computation tools available for them.

Why do we want to linearize our nonlinear system?

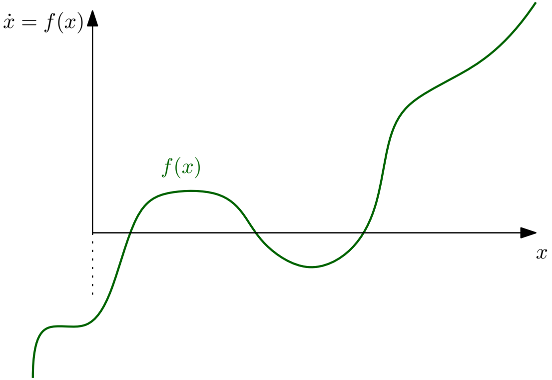
- **Where we' at:** So far, we are able to control our systems with acceptable performance but...
- **Main enemy:** there is always a risk of instability!
  - Recall: PI controller gives you exactly what you want or unstable performance.
- **Main motivation (1):** Predict if your system will be stable or...tune it such that this is guaranteed.
- **Main motivation (2):** Design your system to perform in a certain way.
- Linear systems are easier than nonlinear systems, particularly to address points **(1)** and **(2)**,
  - as we have a large set of mathematical and computation tools available for them.
  - ...less so for nonlinear systems.

Why do we want to linearize our nonlinear system?

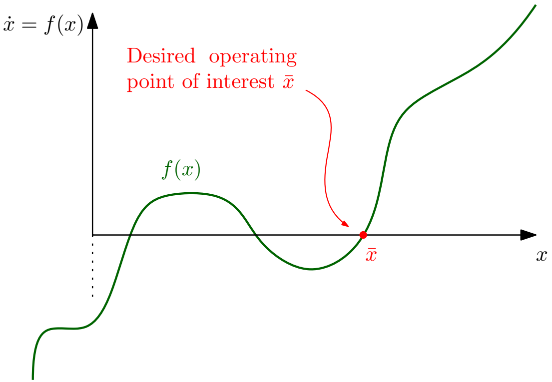
- **Where we' at:** So far, we are able to control our systems with acceptable performance but...
- **Main enemy:** there is always a risk of instability!
  - Recall: PI controller gives you exactly what you want or unstable performance.
- **Main motivation (1):** Predict if your system will be stable or...tune it such that this is guaranteed.
- **Main motivation (2):** Design your system to perform in a certain way.
- Linear systems are easier than nonlinear systems, particularly to address points **(1)** and **(2)**,
  - as we have a large set of mathematical and computation tools available for them.
  - ...less so for nonlinear systems.

Linearization of closed systems - scalar (easy) case

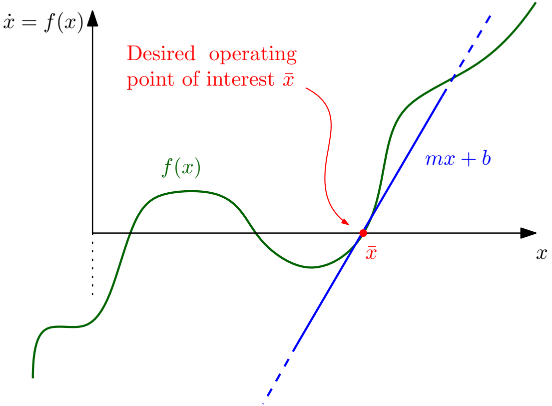
Linearization of closed systems - scalar (easy) case



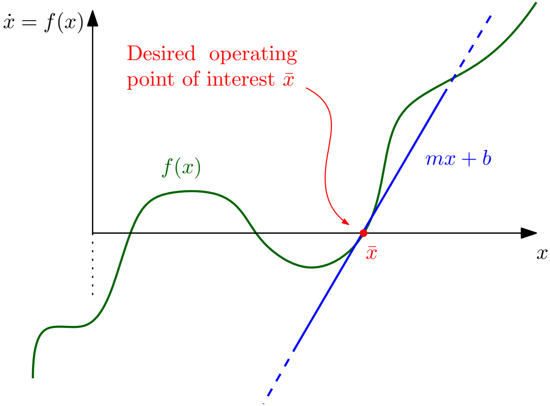
Linearization of closed systems - scalar (easy) case



Linearization of closed systems - scalar (easy) case

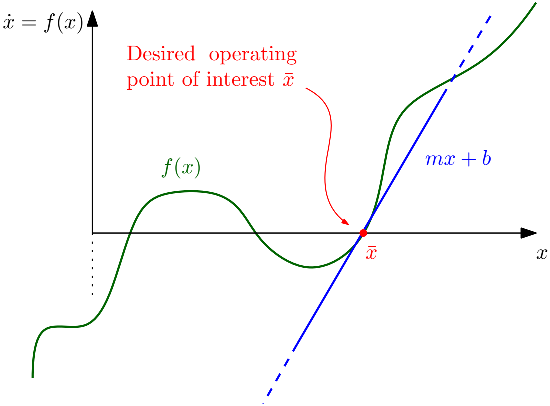


Linearization of closed systems - scalar (easy) case



$$m := \left.\frac{\partial f(x)}{\partial x}\right|_{x=\bar{x}}, \quad \text{and} \quad b := -m\bar{x}$$

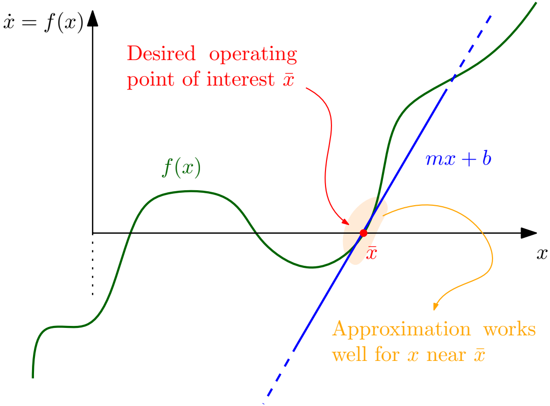
Linearization of closed systems - scalar (easy) case



$$m := \left.\frac{\partial f(x)}{\partial x}\right|_{x=\bar{x}}, \quad \text{and} \quad b := -m\bar{x}$$

$$\text{since} \quad \underbrace{y}_{\dot{x}} - \underbrace{\bar{y}}_{f(\bar{x})=0} = m\underbrace{(x - \bar{x})}_{\tilde{x}},$$

Linearization of closed systems - scalar (easy) case

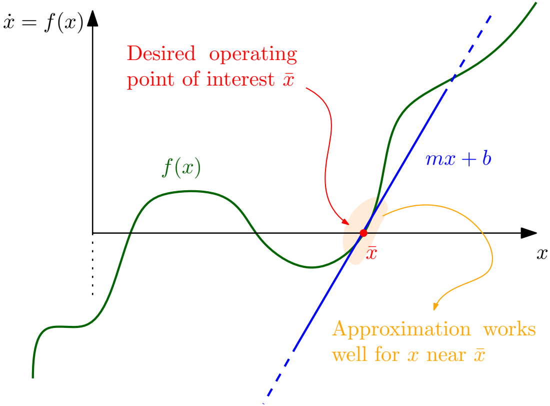


$$m := \left.\frac{\partial f(x)}{\partial x}\right|_{x=\bar{x}}, \quad \text{and} \quad b := -m\bar{x}$$

since $$\underbrace{y}_{\dot{x}} - \underbrace{\bar{y}}_{f(\bar{x})=0} = m(\underbrace{x - \bar{x}}_{\tilde{x}}),$$

then $$\dot{x} \approx m(x - \bar{x}) \Leftrightarrow \dot{\tilde{x}} \approx m\tilde{x}$$

Linearization of closed systems - scalar (easy) case



$$m := \left.\frac{\partial f(x)}{\partial x}\right|_{x=\bar{x}}, \quad \text{and} \quad b := -m\bar{x}$$

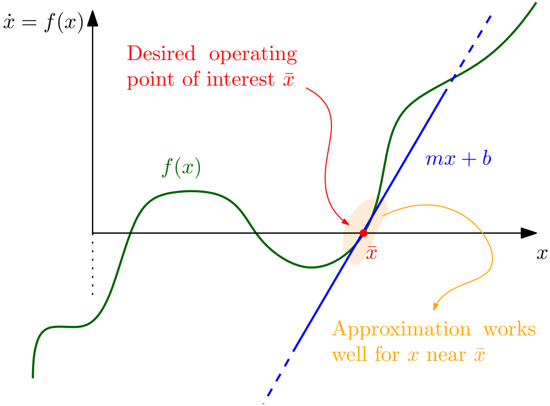
$$\text{since} \quad \underbrace{y}_{\dot{x}} - \underbrace{\bar{y}}_{f(\bar{x})=0} = m\underbrace{(x - \bar{x})}_{\tilde{x}},$$

$$\text{then} \quad \dot{x} \approx m(x - \bar{x}) \Leftrightarrow \dot{\tilde{x}} \approx m\tilde{x}$$

In summary

- We approximate our system $\dot{x} = f(x)$ with $\dot{x} \approx m\tilde{x}$, which holds sufficiently close to the operating points $\leftrightarrow$ under small deviations.
- This gives rise to *small-signal* (stability) studies.

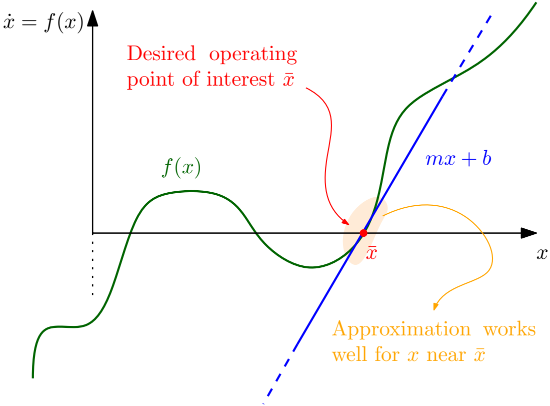
Linearization of closed systems - multidimensional case



Now we have as state variable $x \in \mathbb{R}^n$ and vector field $f : \mathbb{R}^n \to \mathbb{R}^n$; i.e.,

$$\dot{x} = \frac{d}{dt}\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} f_1(x_1, \ldots, x_n) \\ \vdots \\ f_n(x_1, \ldots, x_n) \end{bmatrix}$$

Linearization of closed systems - multidimensional case



Now we have as state variable $x \in \mathbb{R}^n$ and vector field $f : \mathbb{R}^n \to \mathbb{R}^n$; i.e.,

$$\dot{x} = \frac{d}{dt}\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_n(x_1, \dots, x_n) \end{bmatrix}$$

Like the scalar case, we can linearize $f(x)$ around $\bar{x}$ as $\dot{x} \approx A\underbrace{(x - \bar{x})}_{\tilde{x}}$, with

$A \in \mathbb{R}^{n \times n}$ the *Jacobian* of $f(x)$, **evaluated at** $\bar{x}$.

Linearization of closed systems - multidimensional case

More precisely, the A-matrix can be computed by first calculating the Jacobian:

$$\frac{\partial}{\partial x} f(x) = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \cdots & \frac{\partial f_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \cdots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix},$$

and then **evaluating** the Jacobian at $x = \bar{x}$; i.e., $A := \frac{\partial}{\partial x} f(x)\big|_{x=\bar{x}}$. We then have the following approximated linear system,

$$\dot{\tilde{x}} = A\tilde{x} \longleftrightarrow \dot{x} = A(x - \bar{x})$$

for small-signal studies.

Step-by-step example

Consider the following nonlinear system: $$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 5x_1^2 + 2x_2^3 \\ x_1 x_2 + 5 \end{bmatrix}$$

Step-by-step example

Consider the following nonlinear system: $\begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}5x_1^2 + 2x_2^3 \\ x_1 x_2 + 5\end{bmatrix}$

1. identify your variables: x_1, x_2 and your vector fields $f_1(x_1, x_2) = 5x_1^2 + 2x_2^3,\ f_2(x_1, x_2) = x_1 x_2 + 5$

Step-by-step example

Consider the following nonlinear system: $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 5x_1^2 + 2x_2^3 \\ x_1 x_2 + 5 \end{bmatrix}$

1. identify your variables: x_1, x_2 and your vector fields $f_1(x_1, x_2) = 5x_1^2 + 2x_2^3,\ f_2(x_1, x_2) = x_1 x_2 + 5$
2. derivate: $\frac{\partial f_1(x_1,x_2)}{\partial x_1} = 10x_1$, $\frac{\partial f_1(x_1,x_2)}{\partial x_2} = 6x_2$, $\frac{\partial f_2(x_1,x_2)}{\partial x_1} = x_2$ and $\frac{\partial f_2(x_1,x_2)}{\partial x_2} = x_1$

Step-by-step example

Consider the following nonlinear system: $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 5x_1^2 + 2x_2^3 \\ x_1 x_2 + 5 \end{bmatrix}$

1. identify your variables: x_1, x_2 and your vector fields $f_1(x_1, x_2) = 5x_1^2 + 2x_2^3,\ f_2(x_1, x_2) = x_1 x_2 + 5$

2. derivate: $\frac{\partial f_1(x_1,x_2)}{\partial x_1} = 10x_1$, $\frac{\partial f_1(x_1,x_2)}{\partial x_2} = 6x_2$, $\frac{\partial f_2(x_1,x_2)}{\partial x_1} = x_2$ and $\frac{\partial f_2(x_1,x_2)}{\partial x_2} = x_1$

3. Evaluate at equilibrium to form the A-matrix $\rightarrow A = \begin{bmatrix} 10\bar{x}_1 & 6\bar{x}_2 \\ \bar{x}_2 & \bar{x}_1 \end{bmatrix}$

Step-by-step example

Consider the following nonlinear system: $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 5x_1^2 + 2x_2^3 \\ x_1 x_2 + 5 \end{bmatrix}$

1. identify your variables: x_1, x_2 and your vector fields $f_1(x_1, x_2) = 5x_1^2 + 2x_2^3,\ f_2(x_1, x_2) = x_1 x_2 + 5$
   - %%%**Matlab:**%%% `syms x1 x2;    F1=5*x1^2 + 2*x2^3;    F2=x1*x2+5;`
2. derivate: $\frac{\partial f_1(x_1,x_2)}{\partial x_1} = 10x_1$, $\frac{\partial f_1(x_1,x_2)}{\partial x_2} = 6x_2$, $\frac{\partial f_2(x_1,x_2)}{\partial x_1} = x_2$ and $\frac{\partial f_2(x_1,x_2)}{\partial x_2} = x_1$
3. Evaluate at equilibrium to form the A-matrix $\rightarrow A = \begin{bmatrix} 10\bar{x}_1 & 6\bar{x}_2 \\ \bar{x}_2 & \bar{x}_1 \end{bmatrix}$

Step-by-step example

Consider the following nonlinear system: $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 5x_1^2 + 2x_2^3 \\ x_1 x_2 + 5 \end{bmatrix}$

1. identify your variables: x_1, x_2 and your vector fields $f_1(x_1, x_2) = 5x_1^2 + 2x_2^3,\ f_2(x_1, x_2) = x_1 x_2 + 5$
   - %%%**Matlab:**%%% `syms x1 x2;    F1=5*x1^2 + 2*x2^3;    F2=x1*x2+5;`
2. derivate: $\frac{\partial f_1(x_1,x_2)}{\partial x_1} = 10x_1$, $\frac{\partial f_1(x_1,x_2)}{\partial x_2} = 6x_2$, $\frac{\partial f_2(x_1,x_2)}{\partial x_1} = x_2$ and $\frac{\partial f_2(x_1,x_2)}{\partial x_2} = x_1$
   - %%%**Matlab:**%%% `dF1dx1 = diff(F1,x1); dF1dx2 = diff(F1,x2); dF2dx1 = diff(F2,x1); dF2dx2 = diff(F2,x2);%%%%%%`
3. Evaluate at equilibrium to form the A-matrix $\rightarrow A = \begin{bmatrix} 10\bar{x}_1 & 6\bar{x}_2 \\ \bar{x}_2 & \bar{x}_1 \end{bmatrix}$

Step-by-step example

Consider the following nonlinear system: $\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}5x_1^2 + 2x_2^3\\ x_1x_2 + 5\end{bmatrix}$

1. identify your variables: x_1, x_2 and your vector fields $f_1(x_1, x_2) = 5x_1^2 + 2x_2^3,\ f_2(x_1, x_2) = x_1x_2 + 5$
   - %%%**Matlab:**%%% `syms x1 x2;    F1=5*x1^2 + 2*x2^3;    F2=x1*x2+5;`
2. derivate: $\frac{\partial f_1(x_1,x_2)}{\partial x_1} = 10x_1$, $\frac{\partial f_1(x_1,x_2)}{\partial x_2} = 6x_2$, $\frac{\partial f_2(x_1,x_2)}{\partial x_1} = x_2$ and $\frac{\partial f_2(x_1,x_2)}{\partial x_2} = x_1$
   - %%%**Matlab:**%%% `dF1dx1 = diff(F1,x1); dF1dx2 = diff(F1,x2); dF2dx1 = diff(F2,x1); dF2dx2 = diff(F2,x2);%%%%%%`
3. Evaluate at equilibrium to form the A-matrix $\rightarrow A = \begin{bmatrix}10\bar{x}_1 & 6\bar{x}_2\\ \bar{x}_2 & \bar{x}_1\end{bmatrix}$
   - %%%**Matlab:**%%% `A11 = subs(dF1dx1,x1,x1bar);%% and replace other variable if necessary:  A11 = subs(dF1dx1,x2,x2bar)%%; A12 = subs(dF1dx2,x1,x1bar);%% etc...`

Step-by-step example

Consider the following nonlinear system: $\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 5x_1^2 + 2x_2^3 \\ x_1x_2 + 5 \end{bmatrix}$

1. identify your variables: x_1, x_2 and your vector fields $f_1(x_1, x_2) = 5x_1^2 + 2x_2^3$, $f_2(x_1, x_2) = x_1x_2 + 5$
   - %%%**Matlab:**%%% `syms x1 x2;    F1=5*x1^2 + 2*x2^3;    F2=x1*x2+5;`
2. derivate: $\frac{\partial f_1(x_1,x_2)}{\partial x_1} = 10x_1$, $\frac{\partial f_1(x_1,x_2)}{\partial x_2} = 6x_2$, $\frac{\partial f_2(x_1,x_2)}{\partial x_1} = x_2$ and $\frac{\partial f_2(x_1,x_2)}{\partial x_2} = x_1$
   - %%%**Matlab:**%%% `dF1dx1 = diff(F1,x1); dF1dx2 = diff(F1,x2); dF2dx1 = diff(F2,x1); dF2dx2 = diff(F2,x2);%%%%%%`
3. Evaluate at equilibrium to form the A-matrix $\rightarrow A = \begin{bmatrix} 10\bar{x}_1 & 6\bar{x}_2 \\ \bar{x}_2 & \bar{x}_1 \end{bmatrix}$
   - %%%**Matlab:**%%% `A11 = subs(dF1dx1,x1,x1bar);%% and replace other variable if necessary:  A11 = subs(dF1dx1,x2,x2bar)%%; A12 = subs(dF1dx2,x1,x1bar);%% etc...`
   - `A = double([A11, A12; A21, A22]);`

Linearization of open systems - multidimensional case

- Say we have now the **open** nonlinear system $\dot{x} = f(x, u)$; i.e., we have not yet replaced the control equation.

Linearization of open systems - multidimensional case

- Say we have now the **open** nonlinear system $\dot{x} = f(x, u)$; i.e., we have not yet replaced the control equation.
- Similarly, we can approximate the above system with

$$\dot{x} \approx A(x - \bar{x}) + B(u - \bar{u}) \longleftrightarrow \dot{\tilde{x}} \approx A\tilde{x} + B\tilde{u},$$

Linearization of open systems - multidimensional case

- Say we have now the **open** nonlinear system $\dot{x} = f(x, u)$; i.e., we have not yet replaced the control equation.
- Similarly, we can approximate the above system with

$$\dot{x} \approx A(x - \bar{x}) + B(u - \bar{u}) \longleftrightarrow \dot{\tilde{x}} \approx A\tilde{x} + B\tilde{u},$$

with $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ defined as:

$$A := \frac{\partial}{\partial x} f(x, u)\bigg|_{x=\bar{x}, u=\bar{u}}, \quad \text{and} \quad B := \frac{\partial}{\partial u} f(x, u)\bigg|_{x=\bar{x}, u=\bar{u}}$$

,

Linearization of open systems - multidimensional case

More precisely, the **constant** A and B-matrices can be computed by first calculating their Jacobians, and then evaluating them at $x = \bar{x}, u = \bar{u}$, as:

Linearization of open systems - multidimensional case

More precisely, the **constant** A and B-matrices can be computed by first calculating their Jacobians, and then evaluating them at $x = \bar{x}, u = \bar{u}$, as:

$$A := \frac{\partial}{\partial x} f(x, u)\bigg|_{x=\bar{x}, u=\bar{u}} = \begin{bmatrix} \frac{\partial f_1(x,u)}{\partial x_1} & \frac{\partial f_1(x,u)}{\partial x_2} & \cdots & \frac{\partial f_1(x,u)}{\partial x_n} \\ \frac{\partial f_2(x,u)}{\partial x_1} & \frac{\partial f_2(x,u)}{\partial x_2} & \cdots & \frac{\partial f_2(x,u)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(x,u)}{\partial x_1} & \frac{\partial f_n(x,u)}{\partial x_2} & \cdots & \frac{\partial f_n(x,u)}{\partial x_n} \end{bmatrix}_{x=\bar{x}, u=\bar{u}} \in \mathbb{R}^{n \times n}$$

Linearization of open systems - multidimensional case

More precisely, the **constant** A and B-matrices can be computed by first calculating their Jacobians, and then evaluating them at $x = \bar{x}, u = \bar{u}$, as:

$$A := \frac{\partial}{\partial x} f(x,u)\bigg|_{x=\bar{x},u=\bar{u}} = \begin{bmatrix} \frac{\partial f_1(x,u)}{\partial x_1} & \frac{\partial f_1(x,u)}{\partial x_2} & \cdots & \frac{\partial f_1(x,u)}{\partial x_n} \\ \frac{\partial f_2(x,u)}{\partial x_1} & \frac{\partial f_2(x,u)}{\partial x_2} & \cdots & \frac{\partial f_2(x,u)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(x,u)}{\partial x_1} & \frac{\partial f_n(x,u)}{\partial x_2} & \cdots & \frac{\partial f_n(x,u)}{\partial x_n} \end{bmatrix}_{x=\bar{x},u=\bar{u}} \in \mathbb{R}^{n\times n}$$

$$B := \frac{\partial}{\partial u} f(x,u)\bigg|_{x=\bar{x},u=\bar{u}} = \begin{bmatrix} \frac{\partial f_1(x,u)}{\partial u_1} & \frac{\partial f_1(x,u)}{\partial u_2} & \cdots & \frac{\partial f_1(x,u)}{\partial u_m} \\ \frac{\partial f_2(x,u)}{\partial u_1} & \frac{\partial f_2(x,u)}{\partial u_2} & \cdots & \frac{\partial f_2(x,u)}{\partial u_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(x,u)}{\partial u_1} & \frac{\partial f_n(x,u)}{\partial u_2} & \cdots & \frac{\partial f_n(x,u)}{\partial u_m} \end{bmatrix}_{x=\bar{x},u=\bar{u}} \in \mathbb{R}^{n\times m}$$

Outline

Recap from previous lectures

Linearization of nonlinear systems

Bloc 3 Challenge$\times 1/2$

Response of a linear system

Modal transformation & eigenvalues

Bloc 3 (full) Challenge

Bloc 3 Challenge $\times 3/4$:

1. Show linearization procedure of your nonlinear state-space equations for your *open* system $\rightarrow$ when no control is being applied.

Bloc 3 Challenge$\times 3/4$:

1. Show linearization procedure of your nonlinear state-space equations for your *open* system $\rightarrow$ when no control is being applied.
2. **Validate** your linearized model in Simulink by comparing its results against the nonlinear model under *direct equilibrium control.*

Bloc 3 Challenge $\times 3/4$:

1. Show linearization procedure of your nonlinear state-space equations for your *open* system $\rightarrow$ when no control is being applied.
2. **Validate** your linearized model in Simulink by comparing its results against the nonlinear model under *direct equilibrium control.*
3. **Briefly** extend your model to include the (*already linear*) PI control; obtaining an $A_{\text{PI}} \in \mathbb{R}^{3\times 3}$ and $B_{\text{PI}} \in \mathbb{R}^{3\times 1}$ using a state x_1^{ref} as the new input.

Bloc 3 Challenge$\times 3/4$:

1. Show linearization procedure of your nonlinear state-space equations for your *open* system $\rightarrow$ when no control is being applied.
2. **Validate** your linearized model in Simulink by comparing its results against the nonlinear model under *direct equilibrium control.*
3. **Briefly** extend your model to include the (*already linear*) PI control; obtaining an $A_{\mathrm{PI}} \in \mathbb{R}^{3\times 3}$ and $B_{\mathrm{PI}} \in \mathbb{R}^{3\times 1}$ using a state x_1^{ref} as the new input.

   if you have some time, you may wanna validate this model too, to be extra sure...otherwise, thread carefully!

Bloc 3 Challenge $\times 3/4$:

1. Show linearization procedure of your nonlinear state-space equations for your *open* system $\rightarrow$ when no control is being applied.
2. **Validate** your linearized model in Simulink by comparing its results against the nonlinear model under *direct equilibrium control.*
3. **Briefly** extend your model to include the (*already linear*) PI control; obtaining an $A_{\mathrm{PI}} \in \mathbb{R}^{3\times 3}$ and $B_{\mathrm{PI}} \in \mathbb{R}^{3\times 1}$ using a state x_1^{ref} as the new input.

   if you have some time, you may wanna validate this model too, to be extra sure...otherwise, thread carefully!
4. Some more stuff to do...see last slide for full challenge.

Outline

Recap from previous lectures

Linearization of nonlinear systems

Bloc 3 Challenge $\times 1/2$

Response of a linear system

Modal transformation & eigenvalues

Bloc 3 (full) Challenge

Response of a linear system - scalar (easy) case

Consider a general scalar, closed linear system

$$\dot{z} = \lambda z + p \longleftrightarrow \dot{\tilde{z}} = \lambda \tilde{z}, \quad \text{with} \quad \bar{z} := -p/\lambda,$$

with state variable $z \in \mathbb{R}$ with initial value $z(0)$, and a constant parameters $\lambda, p \in \mathbb{R}$.

Response of a linear system - scalar (easy) case

Consider a general scalar, closed linear system

$$\dot{z} = \lambda z + p \longleftrightarrow \dot{\tilde{z}} = \lambda \tilde{z}, \quad \text{with} \quad \bar{z} := -p/\lambda,$$

with state variable $z \in \mathbb{R}$ with initial value $z(0)$, and a constant parameters $\lambda, p \in \mathbb{R}$. **Question:** What is the solution $z(t)$, and when is it stable?

Response of a linear system - scalar (easy) case

Consider a general scalar, closed linear system

$$\dot{z} = \lambda z + p \longleftrightarrow \dot{\tilde{z}} = \lambda \tilde{z}, \quad \text{with} \quad \bar{z} := -p/\lambda,$$

with state variable $z \in \mathbb{R}$ with initial value $z(0)$, and a constant parameters $\lambda, p \in \mathbb{R}$. **Question:** What is the solution $z(t)$, and when is it stable?

- We propose $\tilde{z}(t) = c \cdot e^{st}$, and replace it in $\dot{\tilde{z}} = \lambda \tilde{z}$

Response of a linear system - scalar (easy) case

Consider a general scalar, closed linear system

$$\dot{z} = \lambda z + p \longleftrightarrow \dot{\tilde{z}} = \lambda \tilde{z}, \quad \text{with} \quad \bar{z} := -p/\lambda,$$

with state variable $z \in \mathbb{R}$ with initial value $z(0)$, and a constant parameters $\lambda, p \in \mathbb{R}$. **Question:** What is the solution $z(t)$, and when is it stable?

- We propose $\tilde{z}(t) = c \cdot e^{st}$, and replace it in $\dot{\tilde{z}} = \lambda \tilde{z}$, where we get $c \cdot s e^{st} = \lambda c \cdot e^{st} \implies s = \lambda$.

Response of a linear system - scalar (easy) case

Consider a general scalar, closed linear system

$$\dot{z} = \lambda z + p \longleftrightarrow \dot{\tilde{z}} = \lambda \tilde{z}, \quad \text{with} \quad \bar{z} := -p/\lambda,$$

with state variable $z \in \mathbb{R}$ with initial value $z(0)$, and a constant parameters $\lambda, p \in \mathbb{R}$. **Question:** What is the solution $z(t)$, and when is it stable?

- We propose $\tilde{z}(t) = c \cdot e^{st}$, and replace it in $\dot{\tilde{z}} = \lambda \tilde{z}$, where we get $c \cdot se^{st} = \lambda c \cdot e^{st} \implies s = \lambda$.
- Evaluating our (updated) solution $z(t) = c \cdot e^{\lambda t} + \bar{z}$ at $t = 0$ gives $z(0) = c + \bar{z}$.

Response of a linear system - scalar (easy) case

Consider a general scalar, closed linear system

$$\dot{z} = \lambda z + p \longleftrightarrow \dot{\tilde{z}} = \lambda \tilde{z}, \quad \text{with} \quad \bar{z} := -p/\lambda,$$

with state variable $z \in \mathbb{R}$ with initial value $z(0)$, and a constant parameters $\lambda, p \in \mathbb{R}$. **Question:** What is the solution $z(t)$, and when is it stable?

- We propose $\tilde{z}(t) = c \cdot e^{st}$, and replace it in $\dot{\tilde{z}} = \lambda \tilde{z}$, where we get $c \cdot s e^{st} = \lambda c \cdot e^{st} \implies s = \lambda$.
- Evaluating our (updated) solution $z(t) = c \cdot e^{\lambda t} + \bar{z}$ at $t = 0$ gives $z(0) = c + \bar{z}$.
- Thus, the final solution is:

$$z(t) = [z(0) - \bar{z}]e^{\lambda t} + \bar{z}$$

Response of a linear system - scalar (easy) case

Consider a general scalar, closed linear system

$$\dot{z} = \lambda z + p \longleftrightarrow \dot{\tilde{z}} = \lambda \tilde{z}, \quad \text{with} \quad \bar{z} := -p/\lambda,$$

with state variable $z \in \mathbb{R}$ with initial value $z(0)$, and a constant parameters $\lambda, p \in \mathbb{R}$. **Question:** What is the solution $z(t)$, and when is it stable?

- We propose $\tilde{z}(t) = c \cdot e^{st}$, and replace it in $\dot{\tilde{z}} = \lambda \tilde{z}$, where we get $c \cdot se^{st} = \lambda c \cdot e^{st} \implies s = \lambda$.
- Evaluating our (updated) solution $z(t) = c \cdot e^{\lambda t} + \bar{z}$ at $t = 0$ gives $z(0) = c + \bar{z}$.
- Thus, the final solution is:

$$z(t) = [z(0) - \bar{z}]e^{\lambda t} + \bar{z}$$

- **Stability:** From the above, the stability of our system requires a $\lambda < 0$; i.e., a negative eigenvalue, and is independent of $z(0)$ and $\bar{z}$. Therefore,

Response of a linear system - scalar (easy) case

Consider a general scalar, closed linear system

$$\dot{z} = \lambda z + p \longleftrightarrow \dot{\tilde{z}} = \lambda \tilde{z}, \quad \text{with} \quad \bar{z} := -p/\lambda,$$

with state variable $z \in \mathbb{R}$ with initial value $z(0)$, and a constant parameters $\lambda, p \in \mathbb{R}$. **Question:** What is the solution $z(t)$, and when is it stable?

- We propose $\tilde{z}(t) = c \cdot e^{st}$, and replace it in $\dot{\tilde{z}} = \lambda \tilde{z}$, where we get $c \cdot se^{st} = \lambda c \cdot e^{st} \implies s = \lambda$.
- Evaluating our (updated) solution $z(t) = c \cdot e^{\lambda t} + \bar{z}$ at $t = 0$ gives $z(0) = c + \bar{z}$.
- Thus, the final solution is:

$$z(t) = [z(0) - \bar{z}]e^{\lambda t} + \bar{z}$$

- **Stability:** From the above, the stability of our system requires a $\lambda < 0$; i.e., a negative eigenvalue, and is independent of $z(0)$ and $\bar{z}$. Therefore,

$$\lim_{t \to \infty} z(t) = \lim_{t \to \infty} \left([z(0) - \bar{z}]e^{\lambda t} + \bar{z} \right) = \bar{z}, \quad \forall \quad \lambda < 0$$

Response of a linear system - multidimensional case (easy)

Similar (stability) conclusions follow directly for n independent systems:

Response of a linear system - multidimensional case (easy)

Similar (stability) conclusions follow directly for n independent systems:

$$\dot{z}_1 = \lambda_1 z_1 + p_1 \longrightarrow z_1(t) = [z_1(0) - \bar{z}_1]e^{\lambda_1 t} + \bar{z}_1$$

Response of a linear system - multidimensional case (easy)

Similar (stability) conclusions follow directly for n independent systems:

$$\dot{z}_1 = \lambda_1 z_1 + p_1 \longrightarrow z_1(t) = [z_1(0) - \bar{z}_1]e^{\lambda_1 t} + \bar{z}_1$$
$$\dot{z}_2 = \lambda_2 z_2 + p_2 \longrightarrow z_2(t) = [z_2(0) - \bar{z}_2]e^{\lambda_2 t} + \bar{z}_2$$

Response of a linear system - multidimensional case (easy)

Similar (stability) conclusions follow directly for n independent systems:

$$
\begin{aligned}
\dot{z}_1 &= \lambda_1 z_1 + p_1 \longrightarrow z_1(t) = [z_1(0) - \bar{z}_1]e^{\lambda_1 t} + \bar{z}_1 \\
\dot{z}_2 &= \lambda_2 z_2 + p_2 \longrightarrow z_2(t) = [z_2(0) - \bar{z}_2]e^{\lambda_2 t} + \bar{z}_2 \\
&\vdots \\
\dot{z}_n &= \lambda_n z_3 + p_3 \longrightarrow z_n(t) = [z_n(0) - \bar{z}_n]e^{\lambda_n t} + \bar{z}_n
\end{aligned}
$$

Response of a linear system - multidimensional case (easy)

Similar (stability) conclusions follow directly for n independent systems:

$$
\begin{aligned}
\dot{z}_1 &= \lambda_1 z_1 + p_1 \longrightarrow z_1(t) = [z_1(0) - \bar{z}_1]e^{\lambda_1 t} + \bar{z}_1 \\
\dot{z}_2 &= \lambda_2 z_2 + p_2 \longrightarrow z_2(t) = [z_2(0) - \bar{z}_2]e^{\lambda_2 t} + \bar{z}_2 \\
&\vdots \\
\dot{z}_n &= \lambda_n z_3 + p_3 \longrightarrow z_n(t) = [z_n(0) - \bar{z}_n]e^{\lambda_n t} + \bar{z}_n
\end{aligned}
$$

Using matrix notation, the system can be equivalently expressed as:

$$
\underbrace{\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \vdots \\ \dot{z}_n \end{bmatrix}}_{\dot{z}} = \underbrace{\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}}_{z} + \underbrace{\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}}_{p}
$$

Response of a linear system - multidimensional case (easy)

Similar (stability) conclusions follow directly for n independent systems:

$$
\begin{aligned}
\dot{z}_1 &= \lambda_1 z_1 + p_1 \longrightarrow z_1(t) = [z_1(0) - \bar{z}_1]e^{\lambda_1 t} + \bar{z}_1 \\
\dot{z}_2 &= \lambda_2 z_2 + p_2 \longrightarrow z_2(t) = [z_2(0) - \bar{z}_2]e^{\lambda_2 t} + \bar{z}_2 \\
&\vdots \\
\dot{z}_n &= \lambda_n z_3 + p_3 \longrightarrow z_n(t) = [z_n(0) - \bar{z}_n]e^{\lambda_n t} + \bar{z}_n
\end{aligned}
$$

Using matrix notation, the system can be equivalently expressed as:

$$
\underbrace{\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \vdots \\ \dot{z}_n \end{bmatrix}}_{\dot{z}} = \underbrace{\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}}_{z} + \underbrace{\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}}_{p} \longleftrightarrow \dot{z} = \Lambda z + p; \quad \text{with} \quad \Lambda = \mathrm{diag}(\lambda_i) \in \mathbb{R}^{n \times n}
$$

Response of a linear system - multidimensional case

However, your closed linearized systems $\dot{x} = A(x - \bar{x})$ will look a bit different$\rightarrow$ A matrix is not diagonal!$\rightarrow$ hard to tell if it is stable by just looking at it.

$$\underbrace{\begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \\ \vdots \\ \dot{\tilde{x}}_n \end{bmatrix}}_{\dot{\tilde{x}}} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{bmatrix}}_{\tilde{x}} \longleftrightarrow \dot{\tilde{x}} = A\tilde{x}$$

Response of a linear system - multidimensional case

However, your closed linearized systems $\dot{x} = A(x - \bar{x})$ will look a bit different$\rightarrow$ A matrix is not diagonal!$\rightarrow$ hard to tell if it is stable by just looking at it.

$$\underbrace{\begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \\ \vdots \\ \dot{\tilde{x}}_n \end{bmatrix}}_{\dot{\tilde{x}}} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{bmatrix}}_{\tilde{x}} \longleftrightarrow \dot{\tilde{x}} = A\tilde{x}$$

What is the system response $x(t)$, for a given initial value $x(0)$?

Response of a linear system - multidimensional case

However, your closed linearized systems $\dot{x} = A(x - \bar{x})$ will look a bit different$\rightarrow$ A matrix is not diagonal!$\rightarrow$ hard to tell if it is stable by just looking at it.

$$\underbrace{\begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \\ \vdots \\ \dot{\tilde{x}}_n \end{bmatrix}}_{\dot{\tilde{x}}} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{bmatrix}}_{\tilde{x}} \longleftrightarrow \dot{\tilde{x}} = A\tilde{x}$$

What is the system response $x(t)$, for a given initial value $x(0)$?

- we propose $\tilde{x}(t) = e^{St} \cdot c$ with $c \in \mathbb{R}^n$ and $S \in \mathbb{R}^{n \times n}$ and replace it in $\dot{\tilde{x}} = A\tilde{x}$, where we get $Se^{St} \cdot c = A \cdot e^{St} \cdot c \implies S = A$.
- evaluating our (updated) solution $x(t) = e^{At} \cdot c + \bar{x}$ at $t = 0$ gives $x(0) = c + \bar{x}$. Thus, $x(t) = e^{At}[x(0) - \bar{x}] + \bar{x}$

Outline

Recap from previous lectures

Linearization of nonlinear systems

Bloc 3 Challenge$\times 1/2$

Response of a linear system

Modal transformation & eigenvalues

Bloc 3 (full) Challenge

Modal transformation: diagonal form

Question: How can we then know if our linearized system $\dot{\tilde{x}} = A\tilde{x}$ is stable or not when our A-matrix is nondiagonal?

Modal transformation: diagonal form

Question: How can we then know if our linearized system $\dot{\tilde{x}} = A\tilde{x}$ is stable or not when our A-matrix is nondiagonal?

Answer: We transform it into an equivalent easier one in diagonal form.

Modal transformation: diagonal form

Question: How can we then know if our linearized system $\dot{\tilde{x}} = A\tilde{x}$ is stable or not when our A-matrix is nondiagonal?
Answer: We transform it into an equivalent easier one in diagonal form.

Laying down our strategy:

Modal transformation: diagonal form

Question: How can we then know if our linearized system $\dot{\tilde{x}} = A\tilde{x}$ is stable or not when our A-matrix is nondiagonal?
Answer: We transform it into an equivalent easier one in diagonal form.

Laying down our strategy:

- Take the linear/linearized system $\dot{\tilde{x}} = A\tilde{x}$ as starting point.

Modal transformation: diagonal form

Question: How can we then know if our linearized system $\dot{\tilde{x}} = A\tilde{x}$ is stable or not when our A-matrix is nondiagonal?
Answer: We transform it into an equivalent easier one in diagonal form.

Laying down our strategy:

- Take the linear/linearized system $\dot{\tilde{x}} = A\tilde{x}$ as starting point.
- Consider the variable change $\tilde{z} := Q^{-1}\tilde{x}$, with $\tilde{z} \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n\times n}$ with $\det Q \neq 0$.

Modal transformation: diagonal form

Question: How can we then know if our linearized system $\dot{\tilde{x}} = A\tilde{x}$ is stable or not when our A-matrix is nondiagonal?
Answer: We transform it into an equivalent easier one in diagonal form.

Laying down our strategy:

- Take the linear/linearized system $\dot{\tilde{x}} = A\tilde{x}$ as starting point.
- Consider the variable change $\tilde{z} := Q^{-1}\tilde{x}$, with $\tilde{z} \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n\times n}$ with $\det Q \neq 0$.
- Replacing $\tilde{x} = Q\tilde{z}$ in our linear system gives:

Modal transformation: diagonal form

Question: How can we then know if our linearized system $\dot{\tilde{x}} = A\tilde{x}$ is stable or not when our A-matrix is nondiagonal?
Answer: We transform it into an equivalent easier one in diagonal form.

Laying down our strategy:

- Take the linear/linearized system $\dot{\tilde{x}} = A\tilde{x}$ as starting point.
- Consider the variable change $\tilde{z} := Q^{-1}\tilde{x}$, with $\tilde{z} \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n\times n}$ with $\det Q \neq 0$.
- Replacing $\tilde{x} = Q\tilde{z}$ in our linear system gives:

$$\dot{\tilde{x}} = A\tilde{x} \longrightarrow$$

Modal transformation: diagonal form

Question: How can we then know if our linearized system $\dot{\tilde{x}} = A\tilde{x}$ is stable or not when our A-matrix is nondiagonal?
Answer: We transform it into an equivalent easier one in diagonal form.

Laying down our strategy:

- Take the linear/linearized system $\dot{\tilde{x}} = A\tilde{x}$ as starting point.
- Consider the variable change $\tilde{z} := Q^{-1}\tilde{x}$, with $\tilde{z} \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n\times n}$ with $\det Q \neq 0$.
- Replacing $\tilde{x} = Q\tilde{z}$ in our linear system gives:

$$\dot{\tilde{x}} = A\tilde{x} \longrightarrow Q\dot{\tilde{z}} = AQ\tilde{z}$$

Modal transformation: diagonal form

Question: How can we then know if our linearized system $\dot{\tilde{x}} = A\tilde{x}$ is stable or not when our A-matrix is nondiagonal?
Answer: We transform it into an equivalent easier one in diagonal form.

Laying down our strategy:

- Take the linear/linearized system $\dot{\tilde{x}} = A\tilde{x}$ as starting point.
- Consider the variable change $\tilde{z} := Q^{-1}\tilde{x}$, with $\tilde{z} \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n\times n}$ with $\det Q \neq 0$.
- Replacing $\tilde{x} = Q\tilde{z}$ in our linear system gives:

$$\dot{\tilde{x}} = A\tilde{x} \longrightarrow Q\dot{\tilde{z}} = AQ\tilde{z} \longrightarrow \dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{:=\Lambda}\,\tilde{z}$$

Modal transformation: diagonal form

Question: How can we then know if our linearized system $\dot{\tilde{x}} = A\tilde{x}$ is stable or not when our A-matrix is nondiagonal?
Answer: We transform it into an equivalent easier one in diagonal form.

Laying down our strategy:

- Take the linear/linearized system $\dot{\tilde{x}} = A\tilde{x}$ as starting point.
- Consider the variable change $\tilde{z} := Q^{-1}\tilde{x}$, with $\tilde{z} \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n\times n}$ with $\det Q \neq 0$.
- Replacing $\tilde{x} = Q\tilde{z}$ in our linear system gives:

$$\dot{\tilde{x}} = A\tilde{x} \longrightarrow Q\dot{\tilde{z}} = AQ\tilde{z} \longrightarrow \dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{:=\Lambda}\,\tilde{z}$$

- **Strategy:** Let's choose Q such that $\Lambda := Q^{-1}AQ$ becomes diagonal

$$\rightarrow \Lambda = \mathrm{diag}(\lambda_i) \in \mathbb{R}^{n\times n}$$

Modal transformation: diagonal form

- Then $\dot{\tilde{z}} = \Lambda \tilde{z}$ would look like n-independent linear systems; i.e.:

$$\underbrace{\begin{bmatrix} \dot{\tilde{z}}_1 \\ \dot{\tilde{z}}_2 \\ \vdots \\ \dot{\tilde{z}}_n \end{bmatrix}}_{\dot{\tilde{z}}} = \underbrace{\begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \vdots & 0 \\ \vdots & \dots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_n \end{bmatrix}}_{\tilde{z}}$$

- ...which we know is stable as long as (the real part) of **all** λ's is negative.

Modal transformation: diagonal form

- Then $\dot{\tilde{z}} = \Lambda\tilde{z}$ would look like n-independent linear systems; i.e.:

$$\underbrace{\begin{bmatrix} \dot{\tilde{z}}_1 \\ \dot{\tilde{z}}_2 \\ \vdots \\ \dot{\tilde{z}}_n \end{bmatrix}}_{\dot{\tilde{z}}} = \underbrace{\begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \vdots & 0 \\ \vdots & \dots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_n \end{bmatrix}}_{\tilde{z}}$$

- ...which we know is stable as long as (the real part) of **all** λ's is negative.
- If trajectories of $\tilde{z}(t)$ are stable, and since $\tilde{z} := Q^{-1}\tilde{x} \leftrightarrow \tilde{x} = Q\tilde{z}$, then the trajectories of $\tilde{x}(t)$ are also stable.

Modal transformation: diagonal form

- Then $\dot{\tilde{z}} = \Lambda \tilde{z}$ would look like n-independent linear systems; i.e.:

$$\underbrace{\begin{bmatrix} \dot{\tilde{z}}_1 \\ \dot{\tilde{z}}_2 \\ \vdots \\ \dot{\tilde{z}}_n \end{bmatrix}}_{\dot{\tilde{z}}} = \underbrace{\begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \vdots & 0 \\ \vdots & \dots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_n \end{bmatrix}}_{\tilde{z}}$$

- ...which we know is stable as long as (the real part) of **all** λ's is negative.
- If trajectories of $\tilde{z}(t)$ are stable, and since $\tilde{z} := Q^{-1}\tilde{x} \leftrightarrow \tilde{x} = Q\tilde{z}$, then the trajectories of $\tilde{x}(t)$ are also stable.
- **ok, but how? :** we achieve this by selecting Q as a collection of all the (right) eigenvectors of the A-matrix

Modal transformation: diagonal form

- Then $\dot{\tilde{z}} = \Lambda\tilde{z}$ would look like n-independent linear systems; i.e.:

$$\underbrace{\begin{bmatrix} \dot{\tilde{z}}_1 \\ \dot{\tilde{z}}_2 \\ \vdots \\ \dot{\tilde{z}}_n \end{bmatrix}}_{\dot{\tilde{z}}} = \underbrace{\begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \vdots & 0 \\ \vdots & \dots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}}_{\Lambda} \underbrace{\begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_n \end{bmatrix}}_{\tilde{z}}$$

- ...which we know is stable as long as (the real part) of **all** λ's is negative.
- If trajectories of $\tilde{z}(t)$ are stable, and since $\tilde{z} := Q^{-1}\tilde{x} \leftrightarrow \tilde{x} = Q\tilde{z}$, then the trajectories of $\tilde{x}(t)$ are also stable.
- **ok, but how? :** we achieve this by selecting Q as a collection of all the (right) eigenvectors of the A-matrix
- ...leading to $\Lambda = Q^{-1}AQ = \mathrm{diag}(\lambda_i)$ being a diagonal matrix containing all eigenvalues of the A-matrix.

Eigenvalues and Eigenvectors

Eigenvalues and Eigenvectors

- A real or complex number λ is called an *eigenvalue* of a square matrix $A \in \mathbb{R}^{n\times n}$ if there exists a nonzero vector v such that $Av = \lambda v$.

Eigenvalues and Eigenvectors

- A real or complex number λ is called an *eigenvalue* of a square matrix $A \in \mathbb{R}^{n\times n}$ if there exists a nonzero vector v such that $Av = \lambda v$.

$$Av = \lambda v \longleftrightarrow \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{22} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{v} = \lambda \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{v}$$

Eigenvalues and Eigenvectors

- A real or complex number λ is called an *eigenvalue* of a square matrix $A \in \mathbb{R}^{n\times n}$ if there exists a nonzero vector v such that $Av = \lambda v$.

$$Av = \lambda v \longleftrightarrow \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{22} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{v} = \lambda \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{v}$$

- To find the eigenvalue of A, we write $Av = \lambda v = \lambda \mathbb{I} v$ (with $\mathbb{I} \in \mathbb{R}^{n\times n}$ the identity matrix) as

$$(A - \lambda \mathbb{I})v = 0_n \tag{1}$$

Eigenvalues and Eigenvectors

- A real or complex number λ is called an *eigenvalue* of a square matrix $A \in \mathbb{R}^{n\times n}$ if there exists a nonzero vector v such that $Av = \lambda v$.

$$Av = \lambda v \longleftrightarrow \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{22} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{v} = \lambda \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{v}$$

- To find the eigenvalue of A, we write $Av = \lambda v = \lambda \mathbb{I} v$ (with $\mathbb{I} \in \mathbb{R}^{n\times n}$ the identity matrix) as

$$(A - \lambda \mathbb{I})v = 0_n \tag{1}$$

- if the matrix $(A - \lambda \mathbb{I})$ is *nonsingular* then the only solution of (1) is $v = 0_n$.

Eigenvalues and Eigenvectors

- A real or complex number λ is called an *eigenvalue* of a square matrix $A \in \mathbb{R}^{n \times n}$ if there exists a nonzero vector v such that $Av = \lambda v$.

$$Av = \lambda v \longleftrightarrow \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{22} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{v} = \lambda \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}}_{v}$$

- To find the eigenvalue of A, we write $Av = \lambda v = \lambda \mathbb{I} v$ (with $\mathbb{I} \in \mathbb{R}^{n \times n}$ the identity matrix) as

$$(A - \lambda \mathbb{I})v = 0_n \tag{1}$$

- if the matrix $(A - \lambda \mathbb{I})$ is *nonsingular* then the only solution of (1) is $v = 0_n$.
- to have a (more interesting) non zero solution, $(A - \lambda \mathbb{I})$ must be *singular*:

$$\Delta(\lambda) := \det(A - \lambda \mathbb{I}) = 0$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1$$

$$Av_2 = \lambda_2 v_2$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1$$

$$Av_2 = \lambda_2 v_2$$

$$\vdots$$

$$Av_n = \lambda_n v_n$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots ; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$Av_2 = \lambda_2 v_2$$

$$\vdots$$

$$Av_n = \lambda_n v_n$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix}$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad [\quad;\quad;\quad;\quad]$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix}$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad [Av_1; \quad ; \quad ; \quad]$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix}$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad [Av_1; Av_2; \quad ; \quad]$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix}$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$[Av_1; Av_2; \cdots; Av_n]$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix}$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix}$$

$$[Av_1; Av_2; \cdots; Av_n] = \begin{bmatrix} \quad ; \quad ; \quad ; \quad \end{bmatrix}$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix}$$

$$[Av_1; Av_2; \cdots; Av_n] = \begin{bmatrix} Q \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; & \quad & ; & \quad ; & \quad \end{bmatrix}$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix}$$

$$[Av_1; Av_2; \cdots; Av_n] = \begin{bmatrix} Q \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; Q \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}; \quad ; \quad \end{bmatrix}$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix}$$

$$[Av_1; Av_2; \cdots; Av_n] = \left[Q \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; Q \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}; \cdots; Q \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix} \right]$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix}$$

$$[Av_1; Av_2; \cdots; Av_n] = \left[Q \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; Q \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}; \cdots; Q \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix} \right]$$

$$A \underbrace{[v_1; v_2; \cdots; v_n]}_{Q}$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix}$$

$$[Av_1; Av_2; \cdots; Av_n] = \left[Q \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; Q \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}; \cdots; Q \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix} \right]$$

$$A \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} = Q \underbrace{\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \vdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}}_{\Lambda}$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q}\begin{bmatrix}\lambda_1\\0\\\vdots\\0\end{bmatrix}$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q}\begin{bmatrix}0\\\lambda_2\\\vdots\\0\end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q}\begin{bmatrix}0\\0\\\vdots\\\lambda_n\end{bmatrix}$$

$$[Av_1; Av_2; \cdots; Av_n] = \left[Q\begin{bmatrix}\lambda_1\\0\\\vdots\\0\end{bmatrix}; Q\begin{bmatrix}0\\\lambda_2\\\vdots\\0\end{bmatrix}; \cdots; Q\begin{bmatrix}0\\0\\\vdots\\\lambda_n\end{bmatrix}\right]$$

$$A\underbrace{[v_1; v_2; \cdots; v_n]}_{Q} = Q\underbrace{\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\0 & \lambda_2 & \vdots & 0\\\vdots & \cdots & \ddots & \vdots\\0 & 0 & \cdots & \lambda_n\end{bmatrix}}_{\Lambda}$$

$$AQ = Q\Lambda$$

How does $Av = \lambda v$ help diagonalize our system $\dot{\tilde{z}} = \underbrace{Q^{-1}AQ}_{\Lambda}\,\tilde{z}$?

$$Av_1 = \lambda_1 v_1 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$Av_2 = \lambda_2 v_2 = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

$$Av_n = \lambda_n v_n = \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix}$$

$$[Av_1; Av_2; \cdots; Av_n] = \left[Q \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; Q \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}; \cdots; Q \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_n \end{bmatrix} \right]$$

$$A \underbrace{[v_1; v_2; \cdots; v_n]}_{Q} = Q \underbrace{\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \vdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}}_{\Lambda}$$

$$AQ = Q\Lambda$$

we get $\rightarrow \Lambda = Q^{-1}AQ = \mathrm{diag}(\lambda_i)$

What can the eigenvalues tell us about system performance?

What can the eigenvalues tell us about system performance?

- We know that eigenvalues allow us to conclude if system is stable or not.

What can the eigenvalues tell us about system performance?

- We know that eigenvalues allow us to conclude if system is stable or not.
- However, they also contain information about the system:

What can the eigenvalues tell us about system performance?

- We know that eigenvalues allow us to conclude if system is stable or not.
- However, they also contain information about the system:
  1. rate(s) of convergence – damping,

What can the eigenvalues tell us about system performance?

- We know that eigenvalues allow us to conclude if system is stable or not.
- However, they also contain information about the system:
  1. rate(s) of convergence – damping,
  2. oscillation(s) frequency.

What can the eigenvalues tell us about system performance?

- We know that eigenvalues allow us to conclude if system is stable or not.
- However, they also contain information about the system:
  1. rate(s) of convergence – damping,
  2. oscillation(s) frequency.
- To view this, consider the following **example**:

What can the eigenvalues tell us about system performance?

- We know that eigenvalues allow us to conclude if system is stable or not.
- However, they also contain information about the system:
  1. rate(s) of convergence – damping,
  2. oscillation(s) frequency.
- To view this, consider the following **example**:

$$\begin{bmatrix} \dot{\tilde{z}}_1 \\ \dot{\tilde{z}}_2 \\ \dot{\tilde{z}}_3 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \tilde{z}_3 \end{bmatrix}, \quad \text{with:} \quad \lambda_1 = \alpha + j\beta, \quad \lambda_2 = \lambda_1^*, \quad \lambda_3 \in \mathbb{R}$$

What can the eigenvalues tell us about system performance?

- We know that eigenvalues allow us to conclude if system is stable or not.
- However, they also contain information about the system:
  1. rate(s) of convergence – damping,
  2. oscillation(s) frequency.
- To view this, consider the following **example**:

$$\begin{bmatrix} \dot{\tilde{z}}_1 \\ \dot{\tilde{z}}_2 \\ \dot{\tilde{z}}_3 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \tilde{z}_3 \end{bmatrix}, \quad \text{with:} \quad \lambda_1 = \alpha + j\beta, \quad \lambda_2 = \lambda_1^*, \quad \lambda_3 \in \mathbb{R}$$

$$\tilde{z}_1(t) = \tilde{z}_1(0)e^{\lambda_1}t = \tilde{z}_1(0)e^{(\alpha+j\beta)t}$$

What can the eigenvalues tell us about system performance?

- We know that eigenvalues allow us to conclude if system is stable or not.
- However, they also contain information about the system:
  1. rate(s) of convergence – damping,
  2. oscillation(s) frequency.
- To view this, consider the following **example**:

$$\begin{bmatrix} \dot{\tilde{z}}_1 \\ \dot{\tilde{z}}_2 \\ \dot{\tilde{z}}_3 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \tilde{z}_3 \end{bmatrix}, \quad \text{with:} \quad \lambda_1 = \alpha + j\beta, \quad \lambda_2 = \lambda_1^*, \quad \lambda_3 \in \mathbb{R}$$

$$\tilde{z}_1(t) = \tilde{z}_1(0)e^{\lambda_1 t} = \tilde{z}_1(0)e^{(\alpha+j\beta)t}$$

$$\tilde{z}_2(t) = \tilde{z}_2(0)e^{\lambda_2 t} = \tilde{z}_2(0)e^{(\alpha-j\beta)t},$$

What can the eigenvalues tell us about system performance?

- We know that eigenvalues allow us to conclude if system is stable or not.
- However, they also contain information about the system:
  1. rate(s) of convergence – damping,
  2. oscillation(s) frequency.
- To view this, consider the following **example**:

$$\begin{bmatrix} \dot{\tilde{z}}_1 \\ \dot{\tilde{z}}_2 \\ \dot{\tilde{z}}_3 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \tilde{z}_3 \end{bmatrix}, \quad \text{with:} \quad \lambda_1 = \alpha + j\beta, \quad \lambda_2 = \lambda_1^*, \quad \lambda_3 \in \mathbb{R}$$

$$\begin{aligned} \tilde{z}_1(t) &= \tilde{z}_1(0)e^{\lambda_1}t = \tilde{z}_1(0)e^{(\alpha+j\beta)t} \\ \tilde{z}_2(t) &= \tilde{z}_2(0)e^{\lambda_2}t = \tilde{z}_2(0)e^{(\alpha-j\beta)t}, \\ \tilde{z}_3(t) &= \tilde{z}_3(0)e^{\lambda_3 t} \end{aligned}$$

What can the eigenvalues tell us about system performance?

- We know that eigenvalues allow us to conclude if system is stable or not.
- However, they also contain information about the system:
  1. rate(s) of convergence – damping,
  2. oscillation(s) frequency.
- To view this, consider the following **example**:

$$\begin{bmatrix} \dot{\tilde{z}}_1 \\ \dot{\tilde{z}}_2 \\ \dot{\tilde{z}}_3 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \tilde{z}_3 \end{bmatrix}, \quad \text{with:} \quad \lambda_1 = \alpha + j\beta, \quad \lambda_2 = \lambda_1^*, \quad \lambda_3 \in \mathbb{R}$$

$$\begin{aligned} \tilde{z}_1(t) &= \tilde{z}_1(0)e^{\lambda_1 t} = \tilde{z}_1(0)e^{(\alpha+j\beta)t} \\ \tilde{z}_2(t) &= \tilde{z}_2(0)e^{\lambda_2 t} = \tilde{z}_2(0)e^{(\alpha-j\beta)t}, \\ \tilde{z}_3(t) &= \tilde{z}_3(0)e^{\lambda_3 t} \end{aligned}$$

since $\tilde{x}(t) = Q\tilde{z}(t)$, we have that:

What can the eigenvalues tell us about system performance?

$$\underbrace{\begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \\ \tilde{x}_3(t) \end{bmatrix}}_{\tilde{x}(t)} = \underbrace{\begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} \tilde{z}_1(t) \\ \tilde{z}_2(t) \\ \tilde{z}_3(t) \end{bmatrix}}_{z(t)}$$

What can the eigenvalues tell us about system performance?

$$\underbrace{\begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \\ \tilde{x}_3(t) \end{bmatrix}}_{\tilde{x}(t)} = \underbrace{\begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} \tilde{z}_1(t) \\ \tilde{z}_2(t) \\ \tilde{z}_3(t) \end{bmatrix}}_{z(t)} = \underbrace{\begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} \tilde{z}_1(0)e^{(\alpha+j\beta)t} \\ \tilde{z}_2(0)e^{(\alpha-j\beta)t} \\ \tilde{z}_3(0)e^{\lambda_3 t} \end{bmatrix}}_{\tilde{z}(t)}$$

What can the eigenvalues tell us about system performance?

$$\underbrace{\begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \\ \tilde{x}_3(t) \end{bmatrix}}_{\tilde{x}(t)} = \underbrace{\begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} \tilde{z}_1(t) \\ \tilde{z}_2(t) \\ \tilde{z}_3(t) \end{bmatrix}}_{z(t)} = \underbrace{\begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} \tilde{z}_1(0)e^{(\alpha+j\beta)t} \\ \tilde{z}_2(0)e^{(\alpha-j\beta)t} \\ \tilde{z}_3(0)e^{\lambda_3 t} \end{bmatrix}}_{\tilde{z}(t)}$$

$$\tilde{x}_1(t) = q_{11}\tilde{z}_1(0)e^{(\alpha+j\beta)t} + q_{12}\tilde{z}_2(0)e^{(\alpha-j\beta)t} + q_{13}\tilde{z}_3 e^{\lambda_3 t}$$

What can the eigenvalues tell us about system performance?

$$\underbrace{\begin{bmatrix}\tilde{x}_1(t)\\ \tilde{x}_2(t)\\ \tilde{x}_3(t)\end{bmatrix}}_{\tilde{x}(t)} = \underbrace{\begin{bmatrix}q_{11} & q_{12} & q_{13}\\ q_{21} & q_{22} & q_{23}\\ q_{31} & q_{32} & q_{33}\end{bmatrix}}_{Q} \underbrace{\begin{bmatrix}\tilde{z}_1(t)\\ \tilde{z}_2(t)\\ \tilde{z}_3(t)\end{bmatrix}}_{z(t)} = \underbrace{\begin{bmatrix}q_{11} & q_{12} & q_{13}\\ q_{21} & q_{22} & q_{23}\\ q_{31} & q_{32} & q_{33}\end{bmatrix}}_{Q} \underbrace{\begin{bmatrix}\tilde{z}_1(0)e^{(\alpha+j\beta)t}\\ \tilde{z}_2(0)e^{(\alpha-j\beta)t}\\ \tilde{z}_3(0)e^{\lambda_3 t}\end{bmatrix}}_{\tilde{z}(t)}$$

$$\begin{aligned}\tilde{x}_1(t) &= q_{11}\tilde{z}_1(0)e^{(\alpha+j\beta)t} + q_{12}\tilde{z}_2(0)e^{(\alpha-j\beta)t} + q_{13}\tilde{z}_3 e^{\lambda_3 t}\\ &= e^{\alpha t}\left\{q_{11}\tilde{z}_1(0)e^{j\beta t} + q_{12}\tilde{z}_2(0)e^{-j\beta t}\right\} + q_{13}\tilde{z}_3 e^{\lambda_3 t}\\ &= e^{\alpha t}\left\{q_{11}\tilde{z}_1(0)\left[\cos(\beta t) + j\sin(\beta t)\right] + q_{12}\tilde{z}_2(0)\left[\cos(\beta t) - j\sin(\beta t)\right]\right\} + q_{13}\tilde{z}_3 e^{\lambda_3 t}\\ &= e^{\alpha t}\{\underbrace{\left[q_{11}\tilde{z}_1(0) + q_{12}\tilde{z}_2(0)\right]}_{c_{\mathrm{p}1}\in\mathbb{R}}\cos(\beta t) + j\underbrace{\left[q_{11}\tilde{z}_1(0) - q_{12}\tilde{z}_2(0)\right]}_{jc_{\mathrm{m}1},\ \text{with } c_{\mathrm{m}1}\in\mathbb{R}}\sin(\beta t)\} + q_{13}\tilde{z}_3 e^{\lambda_3 t}\\ &= e^{\alpha t}\{c_{\mathrm{p}1}\cos(\beta t) + c_{\mathrm{m}1}\sin(\beta t)\} + q_{13}\tilde{z}_3 e^{\lambda_3 t}\end{aligned}$$

What can the eigenvalues tell us about system performance?

In summary,

$$\tilde{x}_1(t) = e^{\alpha t}\{c_{\mathrm{p}1}\cos(\beta t) + c_{\mathrm{m}1}\sin(\beta t)\} + q_{13}\tilde{z}_3 e^{\lambda_3 t}$$

$$\tilde{x}_2(t) = e^{\alpha t}\{c_{\mathrm{p}2}\cos(\beta t) + c_{\mathrm{m}2}\sin(\beta t)\} + q_{23}\tilde{z}_3 e^{\lambda_3 t}$$

$$\tilde{x}_3(t) = e^{\alpha t}\{c_{\mathrm{p}3}\cos(\beta t) + c_{\mathrm{m}3}\sin(\beta t)\} + q_{33}\tilde{z}_3 e^{\lambda_3 t}$$

What can the eigenvalues tell us about system performance?

In summary,

$$
\begin{aligned}
\tilde{x}_1(t) =& e^{\alpha t}\{c_{\mathrm{p}1}\cos(\beta t) + c_{\mathrm{m}1}\sin(\beta t)\} + q_{13}\tilde{z}_3 e^{\lambda_3 t} \\
\tilde{x}_2(t) =& e^{\alpha t}\{c_{\mathrm{p}2}\cos(\beta t) + c_{\mathrm{m}2}\sin(\beta t)\} + q_{23}\tilde{z}_3 e^{\lambda_3 t} \\
\tilde{x}_3(t) =& e^{\alpha t}\{c_{\mathrm{p}3}\cos(\beta t) + c_{\mathrm{m}3}\sin(\beta t)\} + q_{33}\tilde{z}_3 e^{\lambda_3 t}
\end{aligned}
$$

- The **real** (and hopefully negative) part α of the complex conjugate eigenvalues λ_1, λ_2 is the rate of decay (damping) of the associated oscillations

What can the eigenvalues tell us about system performance?

In summary,

$$
\begin{aligned}
\tilde{x}_1(t) =& e^{\alpha t}\{c_{\text{p}1}\cos(\beta t) + c_{\text{m}1}\sin(\beta t)\} + q_{13}\tilde{z}_3 e^{\lambda_3 t} \\
\tilde{x}_2(t) =& e^{\alpha t}\{c_{\text{p}2}\cos(\beta t) + c_{\text{m}2}\sin(\beta t)\} + q_{23}\tilde{z}_3 e^{\lambda_3 t} \\
\tilde{x}_3(t) =& e^{\alpha t}\{c_{\text{p}3}\cos(\beta t) + c_{\text{m}3}\sin(\beta t)\} + q_{33}\tilde{z}_3 e^{\lambda_3 t}
\end{aligned}
$$

- The **real** (and hopefully negative) part α of the complex conjugate eigenvalues λ_1, λ_2 is the rate of decay (damping) of the associated oscillations
- ...these oscillations oscillate at a frequency equal to the **imaginary** part β of the complex conjugate eigenvalues λ_1, λ_2.

What can the eigenvalues tell us about system performance?

In summary,
$$
\begin{aligned}
\tilde{x}_1(t) =& e^{\alpha t}\{c_{\mathrm{p}1}\cos(\beta t) + c_{\mathrm{m}1}\sin(\beta t)\} + q_{13}\tilde{z}_3 e^{\lambda_3 t}\\
\tilde{x}_2(t) =& e^{\alpha t}\{c_{\mathrm{p}2}\cos(\beta t) + c_{\mathrm{m}2}\sin(\beta t)\} + q_{23}\tilde{z}_3 e^{\lambda_3 t}\\
\tilde{x}_3(t) =& e^{\alpha t}\{c_{\mathrm{p}3}\cos(\beta t) + c_{\mathrm{m}3}\sin(\beta t)\} + q_{33}\tilde{z}_3 e^{\lambda_3 t}
\end{aligned}
$$

- The **real** (and hopefully negative) part α of the complex conjugate eigenvalues λ_1, λ_2 is the rate of decay (damping) of the associated oscillations
- ...these oscillations oscillate at a frequency equal to the **imaginary** part β of the complex conjugate eigenvalues λ_1, λ_2.
- there is a second (and hopefully negative) rate of convergence $\lambda_3 \in \mathbb{R}$ which is not associated to any additional oscillations.

What can the eigenvalues tell us about system performance?

In summary,
$$\begin{aligned}\tilde{x}_1(t) =& e^{\alpha t}\{c_{\text{p}1}\cos(\beta t) + c_{\text{m}1}\sin(\beta t)\} + q_{13}\tilde{z}_3 e^{\lambda_3 t}\\ \tilde{x}_2(t) =& e^{\alpha t}\{c_{\text{p}2}\cos(\beta t) + c_{\text{m}2}\sin(\beta t)\} + q_{23}\tilde{z}_3 e^{\lambda_3 t}\\ \tilde{x}_3(t) =& e^{\alpha t}\{c_{\text{p}3}\cos(\beta t) + c_{\text{m}3}\sin(\beta t)\} + q_{33}\tilde{z}_3 e^{\lambda_3 t}\end{aligned}$$

- The **real** (and hopefully negative) part α of the complex conjugate eigenvalues λ_1, λ_2 is the rate of decay (damping) of the associated oscillations
- ...these oscillations oscillate at a frequency equal to the **imaginary** part β of the complex conjugate eigenvalues λ_1, λ_2.
- there is a second (and hopefully negative) rate of convergence $\lambda_3 \in \mathbb{R}$ which is not associated to any additional oscillations.
- the slowest (smallest) rate of convergence will then dominate the system dynamics.

Outline

Recap from previous lectures

Linearization of nonlinear systems

Bloc 3 Challenge $\times 1/2$

Response of a linear system

Modal transformation & eigenvalues

Bloc 3 (full) Challenge

Bloc 3 Challenge:

1. Show linearization procedure of your nonlinear state-space equations for your *open* system $\rightarrow$ when no control is being applied.
2. **Validate** your linearized model in Simulink by comparing its results against the nonlinear model under *direct equilibrium control.*

Bloc 3 Challenge:

1. Show linearization procedure of your nonlinear state-space equations for your *open* system $\rightarrow$ when no control is being applied.
2. **Validate** your linearized model in Simulink by comparing its results against the nonlinear model under *direct equilibrium control.*
3. **Briefly** extend your model to include the (*already linear*) PI control; obtaining an $A_{\mathrm{PI}} \in \mathbb{R}^{3\times 3}$ and $B_{\mathrm{PI}} \in \mathbb{R}^{3\times 1}$ using a state x_1^{ref} as the new input.

Bloc 3 Challenge:

1. Show linearization procedure of your nonlinear state-space equations for your *open* system $\rightarrow$ when no control is being applied.
2. **Validate** your linearized model in Simulink by comparing its results against the nonlinear model under *direct equilibrium control.*
3. **Briefly** extend your model to include the (*already linear*) PI control; obtaining an $A_{\mathrm{PI}} \in \mathbb{R}^{3\times 3}$ and $B_{\mathrm{PI}} \in \mathbb{R}^{3\times 1}$ using a state x_1^{ref} as the new input.

   if you have some time, you may wanna validate this model too, to be extra sure…otherwise, thread carefully!

Bloc 3 Challenge:

1. Show linearization procedure of your nonlinear state-space equations for your *open* system $\rightarrow$ when no control is being applied.
2. **Validate** your linearized model in Simulink by comparing its results against the nonlinear model under *direct equilibrium control.*
3. **Briefly** extend your model to include the (*already linear*) PI control; obtaining an $A_{\text{PI}} \in \mathbb{R}^{3\times 3}$ and $B_{\text{PI}} \in \mathbb{R}^{3\times 1}$ using a state x_1^{ref} as the new input.

   if you have some time, you may wanna validate this model too, to be extra sure...otherwise, thread carefully!
4. Investigate (small-signal) stability of your system in closed-loop by plotting eigenvalues of the A-matrix for:

Bloc 3 Challenge:

1. Show linearization procedure of your nonlinear state-space equations for your *open* system $\rightarrow$ when no control is being applied.
2. **Validate** your linearized model in Simulink by comparing its results against the nonlinear model under *direct equilibrium control.*
3. **Briefly** extend your model to include the (*already linear*) PI control; obtaining an $A_{\text{PI}} \in \mathbb{R}^{3\times 3}$ and $B_{\text{PI}} \in \mathbb{R}^{3\times 1}$ using a state x_1^{ref} as the new input.

   if you have some time, you may wanna validate this model too, to be extra sure...otherwise, thread carefully!
4. Investigate (small-signal) stability of your system in closed-loop by plotting eigenvalues of the A-matrix for:

   4.1 at least five different values of the same control parameter (e.g.: K_P),

Bloc 3 Challenge:

1. Show linearization procedure of your nonlinear state-space equations for your *open* system $\rightarrow$ when no control is being applied.
2. **Validate** your linearized model in Simulink by comparing its results against the nonlinear model under *direct equilibrium control.*
3. **Briefly** extend your model to include the (*already linear*) PI control; obtaining an $A_{\mathrm{PI}} \in \mathbb{R}^{3\times 3}$ and $B_{\mathrm{PI}} \in \mathbb{R}^{3\times 1}$ using a state x_1^{ref} as the new input.

   if you have some time, you may wanna validate this model too, to be extra sure...otherwise, thread carefully!
4. Investigate (small-signal) stability of your system in closed-loop by plotting eigenvalues of the A-matrix for:

   4.1 at least five different values of the same control parameter (e.g.: K_P),

   4.2 **or alternatively,** program a `for` loop to see how eigenvalues change for many (e.g: 10-100) different values of the same (control) parameter (parametric sweep) (**optional**).

Bloc 3 Challenge:

1. Show linearization procedure of your nonlinear state-space equations for your *open* system $\rightarrow$ when no control is being applied.
2. **Validate** your linearized model in Simulink by comparing its results against the nonlinear model under *direct equilibrium control.*
3. **Briefly** extend your model to include the (*already linear*) PI control; obtaining an $A_{\text{PI}} \in \mathbb{R}^{3\times3}$ and $B_{\text{PI}} \in \mathbb{R}^{3\times1}$ using a state x_1^{ref} as the new input.

   if you have some time, you may wanna validate this model too, to be extra sure...otherwise, thread carefully!
4. Investigate (small-signal) stability of your system in closed-loop by plotting eigenvalues of the A-matrix for:

   4.1 at least five different values of the same control parameter (e.g.: K_P),

   4.2 **or alternatively,** program a `for` loop to see how eigenvalues change for many (e.g: 10-100) different values of the same (control) parameter (parametric sweep) (**optional**).

   ...to show the stability limits; i.e., for what approx. range of a parameter (e.g.: K_P) the system is stable and when it is unstable.

Bloc 3 Challenge:

1. Show linearization procedure of your nonlinear state-space equations for your *open* system $\rightarrow$ when no control is being applied.
2. **Validate** your linearized model in Simulink by comparing its results against the nonlinear model under *direct equilibrium control.*
3. **Briefly** extend your model to include the (*already linear*) PI control; obtaining an $A_{\mathrm{PI}} \in \mathbb{R}^{3\times 3}$ and $B_{\mathrm{PI}} \in \mathbb{R}^{3\times 1}$ using a state x_1^{ref} as the new input.

   if you have some time, you may wanna validate this model too, to be extra sure...otherwise, thread carefully!
4. Investigate (small-signal) stability of your system in closed-loop by plotting eigenvalues of the A-matrix for:
   - **4.1** at least five different values of the same control parameter (e.g.: K_P),
   - **4.2** **or alternatively,** program a `for` loop to see how eigenvalues change for many (e.g: 10-100) different values of the same (control) parameter (parametric sweep) (**optional**).

   ...to show the stability limits; i.e., for what approx. range of a parameter (e.g.: K_P) the system is stable and when it is unstable.
5. Include example of how α and β appear in the simulation plots.